J Glob Optim (2007) 37:609–630
DOI 10.1007/s10898-006-9069-1

ORIGINAL PAPER

Application of lower bound direct method to engineering structures

Akoa François · Hachemi Abdelkader · Le Thi Hoai An · Mouhtamid Said · Pham Dinh Tao

Received: 25 May 2006 / Accepted: 1 July 2006 / Published online: 9 August 2006

Abstract Direct methods provide elegant and efficient approaches for the prediction of the long-term behaviour of engineering structures under arbitrary complex loading independent of the number of loading cycles. The lower bound direct method leads to a constrained non-linear convex problem in conjunction with finite element methods, which necessitates a very large number of optimization variables and a large amount of computer memory. To solve this large-scale optimization problem, we first reformulate it in a simpler equivalent convex program with easily exploitable sparsity structure. The interior point with DC regularization algorithm (IPDCA) using quasi definite matrix techniques is then used for its solution. The numerical results obtained by this algorithm will be compared with those obtained by general standard code Lancelot. They show the robustness, the efficiency of IPDCA and in particular its great superiority with respect to Lancelot.

Akoa François · Pham Dinh Tao (✉)
LMI, National Institute for Applied Sciences - Rouen
BP 08, Place Emile Blondel, 76131 Mont Saint Aignan Cedex, France
e-mail: pham@insa-roven.fr

Hachemi Abdelkader · Mouhtamid Said
Institut für Allgemeine Mechanik, RWTH Aachen,
Templergraben 64, 52056 Aachen, Germany

Hachemi Abdelkader
e-mail: hachemi@iam.rwth-aachen.de

Mouhtamid Said
e-mail: mouhtani@iam.rwth-aachen.de

Le Thi Hoai An
Laboratory of Theorical and Applied computer science (LITA)
UFR MIM, Metz University
Ile de Saulcy, 57045 Metz, France
e-mail: lethi@science.univ-metz.fr

Keywords Engineering structures · Direct method · Finite element methods · DC programming · DC regularization algorithm · Interior point methods · DC regularization techniques

1 Introduction

There are many industries producing or operating safety-critical structures under heavy loading conditions. Many of these structures or structural components behave ductily and undergo plastic deformations under severe loading or under some normal operating conditions. Their lifetime may be determined by fatigue failure or incrementally increasing deformations due to plasticity. A better understanding of the behaviour of such structures under complex loading conditions may considerably improve their design. In fact, the direct industrial need for end-users of a validated design and assessment method is to improve structural design and the process in run/repair/replace/change operation decision-making. There is a strong industrial need to extend both industrial activities to complex structures (i.e. realistic geometry, complex loading, advanced material modelling, ...). A realistic description of the corresponding material response may require rather sophisticated constitutive models. In combination with well-known numerical tools such as the finite element method, it is in principle possible to study the behaviour of structures by performing a series of incremental elastic-plastic analyses. However, for complex loading histories, the required numerical expense of this kind of procedure may be very high. Furthermore, an accumulation of errors cannot be excluded in principle.

Whereas, direct methods, namely limit and shakedown analysis [1, 2], provides simply and rapidly information on limit states without recurring on the evolution on deformation and material properties as functions of the loading history. The lower bound direct methods lead to a problem of nonlinear mathematical programming in conjunction with finite element methods and need basically [3]:

- the solution of the purely elastic reference problem,
- the use of an optimization procedure, to determine the safety factor against failure and to construct a time-independent self-equilibrated residual stresses field.

The considered problem is a non-linear convex optimization problem with constraints, which necessitates for the engineering applications a very large number of optimization variables and a large amount of computer memory. To solve this large-scale problem with a reasonable computer time, it is necessary to use an efficient algorithm. For that, we apply the interior point with DC regularization algorithm [4, 5, 10]. The obtained numerical results are compared in the special case of limit analysis and shakedown analysis to those obtained by the standard code Lancelot [6].

2 Lower bound direct method

The lower bound direct method can be expressed as follows [1]:

The structure shakes down with respect to the given loading $P(x,t) \in \mathcal{L}$ if there exists a safety factor $\alpha > 1$ and a time-independent field of residual stresses $\sigma^r(x)$ such that its superposition with the purely elastic stresses σ^E does not exceed the yield condition for any time $t > 0$.

$$F(\alpha\sigma^E(x,t) + \sigma^r(x)) \leq 0, \quad \forall x \in V \tag{1}$$

The field of purely elastic stresses satisfies the following system of equations

$$\text{Div}\,\sigma^E = -f^\star, \quad \text{in } V, \tag{2}$$
$$n.\sigma^E = p^\star, \quad \text{on } S_p, \tag{3}$$
$$u^E = u^\star, \quad \text{on } S_u \tag{4}$$

with

$$\varepsilon^E = \frac{1}{2}(\nabla(u^E) + \nabla(u^E)^T), \tag{5}$$

$$\varepsilon^E = E^{-1} : \sigma^E \tag{6}$$

and the field of residual stresses satisfies

$$\text{Div}\,\sigma^r = 0 \quad \text{in } V, \tag{7}$$
$$n.\sigma^r = 0 \quad \text{on } S_p, \tag{8}$$

where V is the volume of the considered structure, S_p and S_u are the disjoint parts of the smooth surface S where the statical and kinematical boundary conditions are prescribed respectively ($S = S_p \cup S_u; \emptyset = S_p \cap S_u$) and n is the outward normal vector to S_p.

3 Discretization by finite element method

The purely elastic stresses σ^E are calculated by using the virtual work principle combined with the finite element discretization with test functions for the displacement fields. Then, the purely elastic stresses σ^E are in equilibrium with body forces $f^\star$ and surface tractions $p^\star$ if the following equality holds

$$\int_V \{\delta\epsilon^E(x)\}^T\{\sigma^E(x)\}\mathrm{d}V = \int_{S_p} \{\delta u^E\}^T\{p^\star\}\mathrm{d}S + \int_V \{\delta u^E\delta\}^T\{f^\star\}\mathrm{d}V \tag{9}$$

for any virtual displacement field δu^E and any virtual strain field $\delta\epsilon^E$ satisfying the compatibility condition (Eq. 5). The virtual strain field $\delta\epsilon^E(x)$ is derived by

$$\{\delta\epsilon^E(x)\} = \sum_{k=1}^{\text{NKE}} B_k(x)\delta u_k^e, \tag{10}$$

where δu_k^e is the vector of virtual displacements of the kth node of the element e and $[B]$ is the compatibility matrix depending on the coordinates. The integration of Eq. 9 has to be carried out over all Gaussian points NGE in the considered element, where the index i refers to the ith Gaussian point. The corresponding coordinate vector shall be denoted by x_i, i.e.

$$\begin{aligned}\int_V \{\delta\epsilon^E(x)\}^T\{\sigma^E(x)\}\mathrm{d}V &= \{\delta u^e\}^T \left\{\sum_{i=1}^{\text{NGE}} w_i|J|_i[B(x_i)]^T[E][B(x_i)]\right\}\{u^e\}, \\ &= \{\delta u^e\}^T[K]\{u^e\}, \\ &= \{\delta u^e\}^T\{F\},\end{aligned} \tag{11}$$

where $\{F\}$ denotes the vector of nodal forces, w_i the weighting factors, $|J|_i$ the determinant of the Jacobian matrix and $[K]$ the stiffness matrix. This integral leads for the ith Gaussian point to

$$\{\sigma^E(x_i)\} = [E][B(x_i)]\{u^e\}. \tag{12}$$

The residual stresses are self-equilibrated, which leads to

$$\int_V \{\delta\epsilon\}^T\{\sigma^r\}\mathrm{d}V = 0. \tag{13}$$

By introducing a vector form for the strain tensor ϵ, the corresponding virtual strains $\delta\epsilon$ are given in each element "e" by

$$\{\delta\epsilon^e\} = \sum_{k=1}^{\mathrm{NKE}} B_k \delta u_k^e. \tag{14}$$

The shape functions of the considered element are the same as for the determination of the purely elastic stresses. Using this relation and introducing the unknown residual stress vector σ_i^r at each Gaussian point i, the equilibrium condition (14) is integrated numerically by using the well-known Gauss–Legendre technique. The integration has to be carried out over all Gaussian points NGE with their weighting factors w_i in the considered element "e"

$$\int_V \{\delta\epsilon\}^T\{\sigma^r\}\mathrm{d}V = \sum_{i=1}^{\mathrm{NGE}} w_i |J|_i \left\{ \sum_{k=1}^{\mathrm{NKE}} B_k \delta u_k^e \right\} \sigma_i^r. \tag{15}$$

By summation of the contributions of all elements and by variation of the virtual node-displacements with regard to the boundary conditions, one finally gets the linear system of equations [7, 8]

$$\sum_{i=1}^{\mathrm{NG}} C_i \sigma_i^r = [C]\{\sigma^r\} = \{0\}, \tag{16}$$

where NG denotes the total number of Gaussian points of the reference body, $[C]$ is a constant equilibrium matrix, uniquely defined by the discretized system and the boundary conditions and $\{\sigma^r\}$ is the global residual stress vector of the discretized reference body.

Let us now assume that the load domain $\mathcal{L}$ is a convex hyperpolyhedron with NV vertices. If the loading has n independently varying components such that

$$\mathcal{L} = \left\{ P/P(x,t) = \sum_{j=1}^{n} \mu_j(t) P_j(x), \mu_j \in [\mu_j^-, \mu_j^+] \right\} \tag{17}$$

then the number of load domain vertices is $\mathrm{NV} = 2^n$. This quite specialized form is the most commonly used in the design of structures. Here, P is the vector of generalized loads (e.g. body forces, surface tractions, prescribed boundary displacements, temperature changes or combinations of them), μ_j are scalar multipliers with upper and lower bounds μ_j^+ and μ_j^-, respectively. For subsequent considerations, the corners of the polyhedron (load domain $\mathcal{L}$) are numbered by the index j, such that $j = 1, \dots, NV$.

The loads, which correspond to each corner of $\mathcal{L}$ are characterized symbolically by P_j. In view of the convexity of the yield function F in (1) and due to the above assumption on the load domain $\mathcal{L}$ [9], the discretized formulation of the lower bound method for the determination of the shakedown loading factor is given by

$$(\mathcal{M}) \begin{cases} \max \alpha, \\ tq, \\ \sum_{i=1}^{\mathrm{NG}} C_i \sigma_i^r \;=\; \{0\}, \\ F(\alpha \sigma_i^E(P_j) + \sigma_i^r) \;\leq\; 0, \\ j \in [1, NV], \quad i \in [1, NG]. \end{cases} \tag{18}$$

The yield criterion has to be fulfiled at Gaussian points $i \in [1, NG]$ and in each load corner $j \in [1, NV]$. The number of unknowns of the optimisation problem (25)–(27) is $\mathrm{N} = 1 + \mathrm{NG} \times \mathrm{NS}$ corresponding to α and $\{\sigma^r\}$. The number of constraints is $\mathrm{NV} \times \mathrm{NG} + \mathrm{NK}$, where NS is the dimension of the stress vector at each Gaussian point and NK denotes the degrees of freedom of displacements of the discretised body. This problem can not be solved efficiently by classical algorithms of optimization because for engineering problems the number of unknowns is in general very high. To overcome the time-consuming is to use a software package for solving large-scale nonlinear optimization problems.

4 Reformulation of the nonlinear convex program

Before applying IPDCA to find the solution of (18), we will introduce in this problem some transformations, which lead to a simpler equivalent convex programme with significantly less variables and constraints. Three reformulations corresponding to the limit analysis with one point, with two points and with four points for the shakedown analysis will be presented.

Problem $(\mathcal{M})$ with one point has (12NG+1) variables and (7NG+3NK) constraints. With two points it has (18NG + 1) variables and (13NG + 3NK) constraints. Finally with four points problem $(\mathcal{M})$ has 30NG + 1 variables and 28NG + 3NK constraints.

4.1 Reformulation of the model with one point

First rewrite the model with one point ($i = 1$), corresponding to proportional loading (limit analysis)

$$\begin{aligned} &\max \alpha \\ &\sum_{r=1}^{\mathrm{NG}} C_r X_r = 0, \\ &\sum_{j=1}^{2} (\sigma_{jr}^1 - \sigma_{j+1,r}^1)^2 + (\sigma_{3r}^1 - \sigma_{1r}^1)^2 + 6 \sum_{j=4}^{6} (\sigma_{jr}^1)^2 \leq 2(a_r)^2, \\ &\sigma_r^1 - \alpha \beta_r^1 - X_r = 0, \\ &r = 1, \ldots, \mathrm{NG}. \end{aligned} \tag{19}$$

The reformulation will need the matrix $T \in \mathbb{R}^{6\times 6}$ defined by

$$T := \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{6}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{6}} \end{pmatrix},$$

whose inverse is explicitly computed as

$$T^{-1} := \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{6} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{6} \end{pmatrix}$$

to make the change of variables

$$\sigma_r^1 = T v_r^1, \quad r = 1, \dots, \mathrm{NG} \tag{20}$$

The second constraint in (19) is then written in the variables $v_r^1 \in \mathbb{R}^6, r = 1, \dots, \mathrm{NG}$ as,

$$(v_{1r}^1)^2 + (v_{2r}^1)^2 + (v_{1r}^1 + v_{2r}^1)^2 + (v_{4r}^1)^2 + (v_{5r}^1)^2 + (v_{6r}^1)^2 \le 2(a_r)^2, \quad r = 1, \dots, \mathrm{NG}. \tag{21}$$

It is clear that in the case of one point, both first and third constraints can be equivalently replaced by the only one

$$\sum_{r=1}^{\mathrm{NG}} C_r(\sigma_r^1 - \alpha\beta_r^1) = 0, \quad r = 1, \dots, \mathrm{NG}.$$

That is in the new variables $v_r^1, r = 1, \dots, \mathrm{NG}$.

$$\sum_{r=1}^{NG} C_r T v_r^1 - \alpha\left(\sum_{r=1}^{NG} C_r \beta_r^1\right) = 0, \quad r = 1, \dots, \mathrm{NG}. \tag{22}$$

We have

$$C_r T v_r^1 = \sum_{j=1, j\neq 3}^{6} [C_r T]^j v_{jr}^1 + v_{3r}^1 [C_r T]^3,$$

where T^j denotes the jth column of T and v^1_{jr} is the jth component of v^1_r. Consider now the new variables $z^1_r \in \mathbb{R}^5$, $r = 1,\dots,\mathrm{NG}$

$$z^1_r = \begin{array}{l} z^1_{1r} = v^1_{1r}, \\ z^1_{2r} = v^1_{2r}, \\ z^1_{3r} = v^1_{4r}, \\ z^1_{4r} = v^1_{5r}, \\ z^1_{5r} = v^1_{6r} \end{array} \tag{23}$$

and $x^1 \in \mathbb{R}^{\mathrm{NG}}$ given by

$$x^1_r = v^1_{3r}, \quad r = 1,\dots,\mathrm{NG} \tag{24}$$

At this point, Problem (19) can be equivalently expressed in the new variables as

$$\begin{array}{c} \max \alpha \\ \sum_{r=1}^{\mathrm{NG}} D_r z^1_r + Bx^1 - \alpha w^1 = 0, \\ (z^1_{1r})^2 + (z^1_{2r})^2 + (z^1_{1r} + z^1_{2r})^2 + (z^1_{3r})^2 + (z^1_{4r})^2 + (z^1_{5r})^2 \le 2(a_r)^2, \\ r = 1,\dots,\mathrm{NG}, \end{array} \tag{25}$$

where $D_r \in \mathbb{R}^{(3NK)\times 5}, B \in \mathbb{R}^{(3NK)\times NG}$ and $w^1 \in \mathbb{R}^{3NK}$ are defined by

$$D_r := [(C_rT)^1 \ (C_rT)^2 \ (C_rT)^4 \ (C_rT)^5 \ (C_rT)^6], \tag{26}$$

$$B := [(C_1T)^3 \ (C_2T)^3 \dots (C_{NG}T)^3], \tag{27}$$

$$w^1 := \sum_{r=1}^{\mathrm{NG}} C_r \beta^1_r. \tag{28}$$

It remains to transform the convex quadratic constraints into the Euclidean ball constraints. For this we first consider the convex quadratic form

$$f(z) := \frac{1}{2}\langle Qz, z\rangle = (z_1)^2 + (z_2)^2 + (z_1 + z_2)^2 + (z_3)^2 + (z_4)^2 + (z_5)^2, \tag{29}$$

where Q is the following positive definite symmetric matrix

$$Q := \begin{pmatrix} 4 & 2 & 0 & 0 & 0 \\ 2 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

Using the Cholesky factorization LL^T of the matrix $\frac{1}{2}Q$ easily computed by

$$L := \begin{pmatrix} \sqrt{2} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

one transforms the convex quadratic constraints in (25) into ($\|\ .\ \|$ being the Euclidean norm of $\mathbb{R}^5$)

$$\left\| L^T z_r^1 \right\|^2 \leq 2(a_r)^2, \quad r = 1, \ldots, \text{NG}$$

Hence, within the new change of variables

$$y_r^1 := L^T z_r^1, \quad r = 1, \ldots, \text{NG}. \tag{30}$$

Problem (25) is formulated as

$$\begin{aligned} &\max \alpha \\ &\sum_{r=1}^{\text{NG}} A_r L^{-T} y_r^1 + B x^1 - \alpha w^1 = 0, \\ &\left\| y_r^1 \right\|^2 \leq 2(a_r)^2, \quad r = 1, \ldots, \text{NG}. \end{aligned} \tag{31}$$

Finally, by introducing the matrices $A_r \in \mathbb{R}^{(3NK)\times 5}$ and the variables $u_r^1 \in \mathbb{R}^5$

$$A_r := \frac{D_r L^{-T}}{a_r\sqrt{2}} \quad \text{and} \quad u_r^1 := \frac{y_r^1}{a_r\sqrt{2}}, \quad r = 1, \ldots, \text{NG}, \tag{32}$$

one gets the new reformulation of (25)

$$\begin{aligned} &\max \alpha \\ &\sum_{r=1}^{NG} A_r u_r^1 + B x^1 - \alpha w^1 = 0, \\ &\left\| u_r^1 \right\|^2 \leq 1, \quad r = 1, \ldots, \text{NG}. \end{aligned} \tag{33}$$

The unknowns are $\alpha \in \mathbb{R}, x^1 \in \mathbb{R}^{\text{NG}}$, $u_r^1 \in \mathbb{R}^5$, $r = 1, \ldots, \text{NG}$. Problem (33) has $(6\text{NG} + 1)$ variables et $(3\text{NK} + \text{NG})$ constraints.

4.2 Reformulation of the model with two points

Let us recall the model with two points ($i = 1, 2$), corresponding to one-parameter variable loading (shakedown analysis)

$$
\begin{aligned}
&\max \alpha \\
&\sum_{r=1}^{\mathrm{NG}} C_r X_r = 0, \\
&\sum_{j=1}^{2} (\sigma_{jr}^1 - \sigma_{j+1,r}^1)^2 + (\sigma_{3r}^1 - \sigma_{1r}^1)^2 + 6 \sum_{j=4}^{6} (\sigma_{jr}^1)^2 \le 2(a_r)^2, \\
&\sigma_r^1 - \alpha \beta_r^1 - X_r = 0, \\
&\sum_{j=1}^{2} (\sigma_{jr}^2 - \sigma_{j+1,r}^2)^2 + (\sigma_{3r}^2 - \sigma_{1r}^2)^2 + 6 \sum_{j=4}^{6} (\sigma_{jr}^2)^2 \le 2(a_r)^2, \\
&\sigma_r^2 - \alpha \beta_r^2 - X_r = 0, \\
&r = 1, \ldots, \mathrm{NG}.
\end{aligned}
\tag{34}
$$

It is clear that Problem (34) is equivalent to that obtained from the former by replacing the first, third and fifth constraints with

$$
\begin{aligned}
&\sum_{r=1}^{\mathrm{NG}} C_r \sigma_r^1 - \alpha \sum_{r=1}^{\mathrm{NG}} C_r \beta_r^1 = 0, \\
&\sigma_r^1 - \alpha \beta_r^1 = \sigma_r^2 - \alpha \beta_r^2, \\
&r = 1, \ldots, \mathrm{NG}.
\end{aligned}
$$

On the other hand, using the same transformations as in the model with one point, one will state the new reformulation of (34) after obtaining the resulting constraints related to

$$
\sigma_r^1 - \alpha \beta_r^1 = \sigma_r^2 - \alpha \beta_r^2, \quad r = 1, \ldots, \mathrm{NG}. \tag{35}
$$

For this one follows each step of the transformations performed in the model of one point. By the change of variables (20), the constraints become in the variables $v_r^i := T^{-1} \sigma_r^i$, $i = 1, 2$ and $r = 1, \ldots, \mathrm{NG}$

$$
v_r^1 - v_r^2 - \alpha T^{-1} (\beta_r^1 - \beta_r^2) = 0, \quad r = 1, \ldots, \mathrm{NG}. \tag{36}
$$

That can be written in the variables $z_r^i \in \mathrm{I\!R}^5$ and $x^i \in \mathrm{I\!R}^{\mathrm{NG}}, i = 1, 2$ (introduced in (23) and (24))

$$
\begin{aligned}
&z_r^1 - z_r^2 - \alpha [T^{-1}]_{\{1,2,4,5,6\}} (\beta_r^1 - \beta_r^2) = 0, \\
&x_r^1 - x_r^2 - \alpha [T^{-1}]_3 (\beta_r^1 - \beta_r^2) = 0, \\
&r = 1, \ldots, \mathrm{NG},
\end{aligned}
\tag{37}
$$

where $[T^{-1}]_{\{1,2,4,5,6\}} \in \mathbb{R}^{5\times 6}$ is the submatrix of T formed by the rows $[T^{-1}]_k$, $k = 1, 2, 4, 5, 6$

$$[T^{-1}]_{\{1,2,4,5,6\}} := \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{6} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{6} \end{pmatrix}.$$

The third change of variables (30): $y_r^i := L^T z_r^1, i = 1, 2$ and $r = 1, \ldots, \mathrm{NG}$ leads to the following equivalence of (37)

$$\begin{aligned} y_r^1 - y_r^2 - \alpha L^T [T^{-1}]_{\{1,2,4,5,6\}}(\beta_r^1 - \beta_r^2) &= 0, \\ x_r^1 - x_r^2 - \alpha [T^{-1}]_3(\beta_r^1 - \beta_r^2) &= 0, \\ r &= 1, \ldots, \mathrm{NG}. \end{aligned} \tag{38}$$

Finally the last change of variables (32): $u_r^1 := \frac{y_r^1}{a_r\sqrt{2}}$, $r = 1, \ldots, \mathrm{NG}$, transforms (38) into

$$\begin{aligned} u_r^1 - u_r^2 - \alpha \frac{1}{a_r\sqrt{2}} L^T [T^{-1}]_{\{1,2,4,5,6\}}(\beta_r^1 - \beta_r^2) &= 0, \\ x_r^1 - x_r^2 - \alpha [T^{-1}]_3(\beta_r^1 - \beta_r^2) &= 0, \\ r &= 1, \ldots, \mathrm{NG} \end{aligned} \tag{39}$$

or

$$u^1 - u^2 - \alpha\gamma = 0, \quad x^1 - x^2 - \alpha\eta = 0, \tag{40}$$

where $\gamma \in \mathbb{R}^{5\mathrm{NG}}$, and $\eta \in \mathbb{R}^{\mathrm{NG}}$ are the following two constant vectors

$$\begin{aligned} \gamma_r &:= \frac{1}{a_r\sqrt{2}} L^T [T^{-1}]_{\{1,2,4,5,6\}}(\beta_r^1 - \beta_r^2), \\ \eta_r &:= [T^{-1}]_3(\beta_r^1 - \beta_r^2), \\ r &= 1, \ldots, \mathrm{NG} \end{aligned}$$

and we deduce the new reformulation of the model with two points (34)

$$\begin{aligned} &\max \alpha \\ \sum_{r=1}^{\mathrm{NG}} A_r u_r^1 + Bx^1 - \alpha w^1 &= 0, \\ u^1 - u^2 - \alpha\gamma &= 0, \\ x^1 - x^2 - \alpha\eta &= 0 \quad (*), \\ \left\| u_r^1 \right\|^2 &\le 1, \\ \left\| u_r^2 \right\|^2 &\le 1, \\ r &= 1, \ldots, \mathrm{NG}. \end{aligned} \tag{41}$$

We actually need to solve the simpler convex program

$$
\begin{aligned}
&\max \alpha\\
\sum_{r=1}^{\mathrm{NG}} A_r u_r^1 + Bx^1 - \alpha w^1 &= 0,\\
u^1 - u^2 - \alpha\gamma &= 0,\\
\left\|u_r^1\right\|^2 &\le 1,\\
\left\|u_r^2\right\|^2 &\le 1,\\
r &= 1,\ldots,\mathrm{NG}.
\end{aligned}
\tag{42}
$$

Indeed, by virtue of $(*)$ in (41), if (u^1, u^2, x^1, α) is an optimal solution of (42) then $(u^1, u^2, x^1, x^2, \alpha)$ with $x^2 := x^1 - \alpha\eta$ solves (41). The converse is also true and both (41) and (42) have the same optimal value.

Problem (42) has $(11\mathrm{NG} + 1)$ variables and $(7\mathrm{NG} + 3\mathrm{NK})$ constraints.

4.3 Reformulation of the model with four points

Problem $(\mathcal{M})$ with four points ($i = 1, \ldots, 4$), corresponding to two-parameters variable loading (shakedown analysis), takes the form

$$
\begin{aligned}
&\max \alpha\\
&\sum_{r=1}^{\mathrm{NG}} C_r X_r = 0,\\
&\sum_{j=1}^{2}(\sigma_{jr}^1 - \sigma_{j+1,r}^1)^2 + (\sigma_{3r}^1 - \sigma_{1r}^1)^2 + 6\sum_{j=4}^{6}(\sigma_{jr}^1)^2 \le 2(a_r)^2,\\
&\sigma_r^1 - \alpha\beta_r^1 - X_r = 0, \qquad (*)\\
&\sum_{j=1}^{2}(\sigma_{jr}^2 - \sigma_{j+1,r}^2)^2 + (\sigma_{3r}^2 - \sigma_{1r}^2)^2 + 6\sum_{j=4}^{6}(\sigma_{jr}^2)^2 \le 2(a_r)^2,\\
&\sigma_r^2 - \alpha\beta_r^2 - X_r = 0, \qquad (**)\\
&\sum_{j=1}^{2}(\sigma_{jr}^3 - \sigma_{j+1,r}^3)^2 + (\sigma_{3r}^3 - \sigma_{1r}^3)^2 + 6\sum_{j=4}^{6}(\sigma_{jr}^3)^2 \le 2(a_r)^2,\\
&\sigma_r^3 - \alpha\beta_r^3 - X_r = 0, \qquad (***)\\
&\sum_{j=1}^{2}(\sigma_{jr}^4 - \sigma_{j+1,r}^4)^2 + (\sigma_{3r}^4 - \sigma_{1r}^4)^2 + 6\sum_{j=4}^{6}(\sigma_{jr}^4)^2 \le 2(a_r)^2,\\
&\sigma_r^4 - \alpha\beta_r^4 - X_r = 0, \qquad (****)\\
&r = 1,\ldots,\mathrm{NG}.
\end{aligned}
\tag{43}
$$

As in the case of two points, by replacing the vectors $X_r, r = 1, \ldots, \mathrm{NG}$ by $\sigma_r^1 - \alpha\beta_r^1$ in the first constraint, one observes that the constraints $(*) - (****)$ in (43) can be gathered in the following

$$
\begin{aligned}
\sum_{r=1}^{\mathrm{NG}} A_r u_r^1 + Bx^1 - \alpha w^1 &= 0,\\
u_r^1 - u_r^2 - \alpha \frac{1}{a_r\sqrt{2}} L^T [T^{-1}]_{\{1,2,4,5,6\}} (\beta_r^1 - \beta_r^2) &= 0,\\
x_r^1 - x_r^2 - \alpha [T^{-1}]_3 (\beta_r^1 - \beta_r^2) &= 0,\\
u_r^2 - u_r^3 - \alpha \frac{1}{a_r\sqrt{2}} L^T [T^{-1}]_{\{1,2,4,5,6\}} (\beta_r^2 - \beta_r^3) &= 0,\\
x_r^2 - x_r^3 - \alpha [T^{-1}]_3 (\beta_r^2 - \beta_r^3) &= 0,\\
u_r^3 - u_r^4 - \alpha \frac{1}{a_r\sqrt{2}} L^T [T^{-1}]_{\{1,2,4,5,6\}} (\beta_r^3 - \beta_r^4) &= 0,\\
x_r^3 - x_r^4 - \alpha [T^{-1}]_3 (\beta_r^3 - \beta_r^4) &= 0,\\
r &= 1, \ldots, \mathrm{NG}.
\end{aligned} \tag{44}
$$

On the other hand the four convex quadratic constraints are transformed into

$$
\begin{aligned}
\left\| u_r^1 \right\|^2 &\le 1,\\
\left\| u_r^2 \right\|^2 &\le 1,\\
\left\| u_r^3 \right\|^2 &\le 1,\\
\left\| u_r^3 \right\|^2 &\le 1,\\
\left\| u_r^4 \right\|^2 &\le 1,\\
r &= 1, \ldots, \mathrm{NG}.
\end{aligned} \tag{45}
$$

Now let $\xi, \zeta \in \mathbb{R}^{5\mathrm{NG}}$ and $\lambda, \mu, \nu \in \mathbb{R}^{\mathrm{NG}}$ be the constant vectors defined by

$$
\xi_r \ : = \frac{1}{a_r\sqrt{2}} L^T [T^{-1}]_{\{1,2,4,5,6\}} (\beta_r^2 - \beta_r^3), \tag{46}
$$

$$
\begin{aligned}
\zeta_r \ &: = \frac{1}{a_r\sqrt{2}} L^T [T^{-1}]_{\{1,2,4,5,6\}} (\beta_r^3 - \beta_r^4),\\
\lambda_r \ &: = [T^{-1}]_3 (\beta_r^1 - \beta_r^2),\\
\mu_r \ &: = [T^{-1}]_3 (\beta_r^2 - \beta_r^3),\\
\nu_r \ &: = [T^{-1}]_3 (\beta_r^3 - \beta_r^4),\\
r &= 1, \ldots, \mathrm{NG}.
\end{aligned} \tag{47}
$$

Then the model with four points (43) takes the equivalent form

$$
\begin{aligned}
&\max \alpha \\
\sum_{r=1}^{NG} A_r u_r^1 + Bx^1 - \alpha w^1 &= 0, \\
u^1 - u^2 - \alpha\gamma &= 0, \\
u^2 - u^3 - \alpha\xi &= 0, \\
u^3 - u^4 - \alpha\zeta &= 0, \\
x^1 - x^2 - \alpha\lambda &= 0 \quad (*) \\
x^2 - x^3 - \alpha\mu &= 0 \quad (**) \\
x^3 - x^4 - \alpha\nu &= 0 \quad (***) \\
\|u_r^i\|^2 &\le 1, \quad i = 1, \dots, 4, \\
r &= 1, \dots, NG
\end{aligned} \tag{48}
$$

As above in the model with two points, it suffices to solve the simpler convex program

$$
\begin{aligned}
&\max \alpha \\
\sum_{r=1}^{NG} A_r u_r^1 + Bx^1 - \alpha w^1 &= 0, \\
u^1 - u^2 - \alpha\gamma &= 0, \\
u^2 - u^3 - \alpha\xi &= 0, \\
u^3 - u^4 - \alpha\zeta &= 0, \\
\|u_r^i\|^2 &\le 1, \quad i = 1, \dots, 4, \\
r &= 1, \dots, NG
\end{aligned} \tag{49}
$$

because of the following properties:

(1) if $(u^1, u^2, u^3, u^4, x^1, \alpha)$ is an optimal solution to (49) then $(u^1, u^2, u^3, u^4, x^1, x^2, x^3, x^4, \alpha)$, where (x^2, x^3, x^4) is computed by $(*)$, $(**)$ and $(***)$ of (48)

$$
\begin{aligned}
x^2 &:= x^1 - \alpha\lambda, \\
x^3 &:= x^2 - \alpha\mu, \\
x^4 &:= x^3 - \alpha\nu,
\end{aligned} \tag{50}
$$

solves (48) and both (48) and (49) have the sample optimal value.

(2) The converse is also true
The unknowns in (49) are $\alpha \in \mathbb{R}$, $x^1 \in \mathbb{R}^{NG}$, $u^i = (u_r^i) \in \mathbb{R}^{5NG}$ with $u_r^i \in \mathbb{R}^5$ for $i = 1, \dots, 4$ and $r = 1, \dots, NG$, and $\alpha \in \mathbb{R}$.This convex program has $(21NG + 1)$ variables and $(19NG + 3NK)$ constraints.

Remark 1 The new reformulations of the models with one and four points reduce considerably the number of variables and constraints. Furthermore the transformtion of convex quadratic constraints into Euclidean ball constraints allows better exploting sparsity of the KKT systems in our method IPDCA introduced in [10].

5 Interior-point with DC regularization algorithm algorithm for large-scale problems

The IPDCA is based on interior points approaches and DCA [11–17]. Its aim is to find a KKT point of the following nonlinear programming problem

$$(\mathcal{P}_{\mathcal{E}\mathcal{I}}) \begin{cases} \min f(x), \\ Ax - b \quad = 0, \\ c(x) \qquad \geq 0, \\ x \in \mathbb{R}^n, \end{cases}$$

where f and $c\colon \mathbb{R}^n \to \mathbb{R}^{m_I}$ are two twice continuous differentiable functions and c is supposed to be *concave.* $A \in \mathbb{R}^{m_E \times n}$ a *surjective* matrix, and $b \in \mathbb{R}^{m_E}$ a vector.

The first step in the interior point approach considered is to add slack variables w to each of the inequality constraints in $(\mathcal{P}_{\mathcal{E}\mathcal{I}})$ and to add a linear constraints in order to handle free variable x. The problem $(\mathcal{P}_{\mathcal{E}\mathcal{I}})$ is then reformulated to,

$$(\mathcal{P}) \begin{cases} \min f(x), \\ Ax - b \;=\; 0, \\ c(x) - w \;=\; 0, \\ x - y + z \;=\; 0, \\ w \geq 0, \quad y \geq 0, \quad z \geq 0. \end{cases}$$

The second step is to consider the problem with the barrier objective function

$$(\mathcal{P}_\mu) \begin{cases} \min \bar{f}_\mu(x, w, y, z), \\ Ax - b \;=\; 0, \\ c(x) - w \;=\; 0, \\ x - y + z \;=\; 0, \\ w > 0, \quad y > 0, \quad z > 0 \end{cases}$$

with

$$\bar{f}_\mu(x, w, y, z) = f(x) - \mu \sum_{i=1}^{m_I} \log(w_i) - \mu \sum_{j=1}^{n} \log(y_j) - \mu \sum_{j=1}^{n} \log(z_j).$$

The IPDCA requires a DC decomposition of $f = g - h$, where g and h are two convex functions.

By linearizing the concave component of the objective function, IPDCA solves approximately the problem,

$$(\mathcal{DC}_k) \begin{cases} \min \bar{g}_k(x), \\ Ax - b \;=\; 0, \\ c(x) - w \;=\; 0, \\ x - y + z \;=\; 0, \\ w \geq 0, \quad y \geq 0, \quad z \geq 0, \end{cases}$$

where $\bar{g}_k(x) = g(x) - \nabla^T h(x_k)x$.

The problem is then transformed to a sequence of problems with the logarithmic barrier function:

$$(\mathcal{DC}_\mu) \begin{cases} \min g_\mu(w, x, y, z) - \nabla^T h(x_k)x, \\ Ax - b \;=\; 0, \\ c(x) - w \;=\; 0, \\ x - y + z \;=\; 0, \end{cases}$$

with

$$\begin{aligned} g_\mu(w,x,y,z) &= g(x) - \mu\sum_{i=1}^{m_I}\log(w_i) - \mu\sum_{j=1}^{n}\log(y_j) - \mu\sum_{j=1}^{n}\log(z_j),\\ h_\mu(w,x,y,z) &= h(x). \end{aligned}$$

5.1 Step computation

The Lagrangian of $(\mathcal{DC}_\mu)$ is

$$L_\mu = g_\mu(w,x,y,z) - \nabla^T h(x_k)x - \lambda_E^T(Ax-b) - \lambda_I^T(c(x)-w) - s^T(x-y+z). \quad (51)$$

Expressing the first order necessary conditions for $(\mathcal{DC}_\mu)$ yields

$$(KKT_\mu)\begin{cases} \nabla_x L_\mu &= \nabla_x g_\mu(w,x,y,z) - \nabla h(x_k) - A^T\lambda_E - B(x)^T\lambda_I - s = 0,\\ \nabla_w L_\mu &= -\mu W^{-1}e + \lambda_I = 0,\\ \nabla_y L_\mu &= -\mu Y^{-1}e + s = 0,\\ \nabla_z L_\mu &= -\mu Z^{-1}e - s = 0,\\ \nabla_{\lambda_E} L_\mu &= -Ax + b = 0,\\ \nabla_{\lambda_I} L_\mu &= -c(x) + w = 0,\\ \nabla_s L_\mu &= -x + y - z = 0, \end{cases}$$

with $B(x) = \nabla c(x)$, $W = \text{diag}(w_i : i = 1,2,\ldots,m_I).$, $Z = \text{diag}(z_i : i = 1,2,\ldots,n)$, $Y = \text{diag}(y_i : i = 1,2,\ldots,n)$.

Let us set $v = \mu Z^{-1}e$ and since $\nabla_x g_\mu(w,x,y,z) = \nabla g(x)$, after premultiplying the second and the third equation of (KKT_μ) by W and Y, respectively.

Let us set

$$F_\mu(w,x,y,z;\lambda_E,\lambda_I,v,s) = \begin{pmatrix} \nabla g(x) - \nabla h(x_k) - A^T\lambda_E - B(x)^T\lambda_I - s\\ -\mu e + W\lambda_I\\ -\mu e + Ys\\ -\mu e + Zv\\ -Ax+b\\ -c(x)+w\\ -v-s\\ -x+y-z \end{pmatrix}.$$

We should solve the nonlinear system of equations,

$$F_\mu(w,x,y,z;\lambda_E,\lambda_I,v,s) = 0. \quad (52)$$

Note that at the point $\Pi_k = (x_k,w_k,y_k,z_k;\lambda_{E,k},\lambda_{I,k},s_k,v_k,)$ this system is the KKT conditions of the problem $(\mathcal{P})$ at the point Π_k, because $\nabla g(x_k) - \nabla h(x_k) = \nabla f(x_k)$.

Using the Newton method to solve the nonlinear system (52), At the k-th iteration and for μ fixed, the linear system to solve is,

$$J(\Pi_k)\triangle\Pi = -F_\mu(\Pi_k), \quad (53)$$

where $J(\Pi_k)$ is the Jacobian matrix of $F_\mu(\Pi_k)$ and $\triangle\Pi = (\triangle x, \triangle w, \triangle y, \triangle z; \triangle\lambda_E, \triangle\lambda_I, \triangle s, \triangle v)$. With

$$J(\Pi_k) = \begin{pmatrix} G_k & 0 & 0 & 0 & -A^T & -B_k^T & -I_n & 0 \\ 0 & \Lambda_{I,k} & 0 & 0 & 0 & W_k & 0 & 0 \\ 0 & 0 & S_k & 0 & 0 & 0 & Y_k & 0 \\ 0 & 0 & 0 & V_k & 0 & 0 & 0 & Z_k \\ -A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -B_k & I_{m_I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -I_n & -I_n \\ -I_n & 0 & I_n & -I_n & 0 & 0 & 0 & 0 \end{pmatrix} \quad (54)$$

with $G_k = G(x_k)$, the Hessian of the Lagrangian at $x_k, B_k = B(x_k), S_k = \mathrm{diag}(s_k^i : i = 1, 2, \ldots, n), \Lambda_{I,k} = \mathrm{diag}(\lambda_{I,k}^i : i = 1, 2, \ldots, n), V_k = \mathrm{diag}(v_k^i : i = 1, 2, \ldots, n)$ and I_n, I_{m_I}, I_{m_E} the identity matrices.

For sick of clarity in our presentation we will leave for the moment the indexation with k.

Let us set,

$$\begin{aligned} \sigma &= \nabla f(x) - A^T\lambda_E - B(x)^T\lambda_I - s, \\ \gamma_1 &= \mu e - W\lambda_I, \\ \gamma_2 &= \mu e - Ys, \\ \gamma_3 &= \mu e - Zv, \\ \rho_1 &= Ax - b, \\ \rho_2 &= c(x) - w, \\ \beta_1 &= v + s, \\ \beta_2 &= x - y + z, \end{aligned}$$

then the linear system can be written,

$$\begin{pmatrix} G(x) & 0 & 0 & 0 & -A^T & -B(x)^T & -I_n & 0 \\ 0 & \Lambda_I & 0 & 0 & 0 & W & 0 & 0 \\ 0 & 0 & S & 0 & 0 & 0 & Y & 0 \\ 0 & 0 & 0 & V & 0 & 0 & 0 & Z \\ -A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -B(x) & I_{m_I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -I_n & -I_n \\ -I_n & 0 & I_n & -I_n & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \triangle x \\ \triangle w \\ \triangle y \\ \triangle z \\ \triangle\lambda_E \\ \triangle\lambda_I \\ \triangle s \\ \triangle v \end{pmatrix} = \begin{pmatrix} -\sigma \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \rho_1 \\ \rho_2 \\ \beta_1 \\ \beta_2 \end{pmatrix}.$$

By a successive alimination process we have from the seventh equation

$$\triangle v = -\triangle s - \beta_1 \quad (55)$$

and by replacing $\triangle v$ in the fourth equation we get,

$$\triangle z = V^{-1}Z\triangle s + \beta_1 + V^{-1}Z\beta_1 + V^{-1}\gamma_3. \quad (56)$$

The last equation of the system gives,

$$\triangle y = \triangle x + \triangle z + \beta_2. \quad (57)$$

So

$$\triangle y = \triangle x + V^{-1}Z\triangle s + \beta_1 + V^{-1}Z\beta_1 + V^{-1}\gamma_3 + \beta_2. \quad (58)$$

If we set $E = (YS^{-1} + ZV^{-1})^{-1}$ we have,

$$\triangle s = -E\triangle x + E[V^{-1}(Z\beta_1 + \gamma_3) + \beta_1 + S^{-1}\gamma_2]. \tag{59}$$

The sixth equation of the system gives,

$$\triangle w = -D\triangle\lambda_I + \Lambda_I^{-1}\gamma_1, \tag{60}$$

where $D = \Lambda_I^{-1}W$. We get after premultiplying the equations by -1 the reduced linear system,

$$\begin{pmatrix} -(G(x)+E) & A^T & B(x)^T \\ A & 0 & 0 \\ B(x) & 0 & D \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda_E \\ \Delta\lambda_I \end{pmatrix} = \begin{pmatrix} \Sigma \\ -\rho_1 \\ -\rho_2 + \Lambda_I^{-1}\gamma_1 \end{pmatrix}. \tag{61}$$

with $\Sigma = \sigma - E[V^{-1}(Z\beta_1 + \gamma_3) + \beta_1 + S^{-1}\gamma_2]$.

Note that the matrix in the linear system (61) is not quasidefinite because the 2×2 matrix block is zero. To make it quasidefinite we replace this 2×2 matrix block by the perturbation $\delta^2 I_{m_E}$ to get the perturbed linear system,

$$\begin{pmatrix} -(G(x)+E) & A^T & B(x)^T \\ A & \delta^2 I_{m_E} & 0 \\ B(x) & 0 & D \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda_E \\ \Delta\lambda_I \end{pmatrix} = \begin{pmatrix} \Sigma \\ -\bar{\rho}_1 \\ -\rho_2 + \Lambda_I^{-1}\gamma_1 \end{pmatrix} \tag{62}$$

with $\bar{\rho}_1 = Ax - b + \delta^2(\lambda_E - \lambda_{E,k})$.

We note that the right hand side of this equation is the KKT conditions of the *dual regularized problem*,

$$(\mathcal{DR}_k) \begin{cases} \max f_\mu(x) + \lambda_E^T b - \frac{\delta}{2}(\lambda_E - \lambda_{E,k})^T(\lambda_E - \lambda_{E,k}) - \lambda_I^T c(x) + [\lambda_I^T B(x)]^T x \\ \quad +\nabla f(x)^T x + s^T z, \\ \nabla f(x) - A^T\lambda_E - B(x)^T\lambda_I - s = 0, \\ \lambda_E, \lambda_I > 0, s > 0, \end{cases}$$

with

$$f_\mu(x) = f(x) - \mu\sum_{i=1}^{m_I}\log(\lambda_{I,i}) - \mu\sum_{j=1}^{n}\log(s_j).$$

The new quadratic term present in the objective penalyzes directions that move the current iterate far away from the iterate $\lambda_{E,k}$.

The problem $(\mathcal{DR}_k)$ is the regularized of the problem

$$(\mathcal{D}) \begin{cases} \max f_\mu(x) + \lambda_E^T b - \lambda_I^T c(x) + [\lambda_I^T B(x)]^T x + \nabla f(x)^T x + s^T z \\ \nabla f(x) - A^T\lambda_E - B(x)^T\lambda_I - s = 0, \\ \lambda_E, \lambda_I > 0, s > 0, \end{cases}$$

that is the dual of the problem $(\mathcal{P})$.

Note that by the choice $(\lambda_{E,k}, \lambda_{I,k}, s^k)$ as the current dual triplet at the iteration k, we ensure that the right hand side of the reduce system is not altered, that is,

$$\bar{\rho}_1 = \rho_1 = Ax - b,$$

while we benefit with the regularization $\delta^2 I_{m_E}$; and the linear system that we solve in IPDCA at each iteration is,

$$\begin{pmatrix} -(G(x)+E) & A^T & B(x)^T \\ A & \delta^2 I_{m_E} & 0 \\ B(x) & 0 & D \end{pmatrix} \begin{pmatrix} \triangle x \\ \triangle \lambda_E \\ \triangle \lambda_I \end{pmatrix} = \begin{pmatrix} \Sigma \\ -\rho_1 \\ -\rho_2 + \Lambda_I^{-1}\gamma_1 \end{pmatrix}. \tag{63}$$

The approach that we have adopted here is close to the primal-dual regularization approach presented in [18–20], the difference resides in the choice of the primal regularization. In our approach the primal regularization is the DC regularization.

To avoid using definite matrix perturbations, another alternative suggested by Vanderbei [21, 22] to treat the equality affine constraints consists in replacing these constraints by,

$$Ax + b - t = 0, \quad t + v = 0, \quad t \geq 0, \quad v \geq 0.$$

With this approach we add linear constraints and two variables, compared to our approach, this approach consumes more memory for a comparable result.

The matrix involved in (63) is now quasi-definite and is known to be strongly factorizable to a form of Cholesky-like factorization [23].

To compute the next iterate,

$$\Pi_{k+1} = \Pi_k + \Upsilon_k \triangle \Pi_k$$

with $\Upsilon_k = \operatorname{diag}\{\alpha_{x_k} I_n, \alpha_{w_k} I_{m_I}, \alpha_{y_k} I_n, \alpha_{z_k} I_n, \alpha_{\lambda_{E,k}} I_{m_E}, \alpha_{\lambda_{I,k}} I_{m_I}, \alpha_{s_k} I_n, \alpha_{v_k} I_n\}$, the steps sizes $\alpha_{x_k}, \alpha_{w_k}, \alpha_{y_k}, \alpha_{z_k}, \alpha_{\lambda_{E,k}}, \alpha_{\lambda_{I,k}}, \alpha_{s_k}, \alpha_{v_k}$, are determined in $(0, 1]$ and could be equal each other. Moreover the slack variables, w_k, y_k, z_k and dual variables $\lambda_{I,k}, s_k$, are all positive at the solution this propriety is maintained during the iteration process. We define

$$\begin{aligned}
\alpha_{w_k}^{\max} &= \gamma \max_{1\leq j\leq m_I} \left\{ -\frac{w_k^{(j)}}{\triangle w_k^{(j)}} : \triangle w_k^{(j)} < 0 \right\}, \\
\alpha_{y_k}^{\max} &= \gamma \max_{1\leq j\leq n} \left\{ -\frac{y_k^{(j)}}{\triangle y_k^{(j)}} : \triangle y_k^{(j)} < 0 \right\}, \\
\alpha_{z_k}^{\max} &= \gamma \max_{1\leq j\leq n} \left\{ -\frac{z_k^{(j)}}{\triangle z_k^{(j)}} : \triangle z_k^{(j)} < 0 \right\}, \\
\alpha_{\lambda_{I,k}}^{\max} &= \gamma \max_{1\leq j\leq m_I} \left\{ -\frac{\lambda_{I,k}^{(j)}}{\triangle \lambda_{I,k}^{(j)}} : \triangle \lambda_{I,k}^{(j)} < 0 \right\}, \\
\alpha_{s_k}^{\max} &= \gamma \max_{1\leq j\leq n} \left\{ -\frac{s_k^{(j)}}{\triangle s_k^{(j)}} : \triangle s_k^{(j)} < 0 \right\}, \\
\alpha_{v_k}^{\max} &= \gamma \max_{1\leq j\leq n} \left\{ -\frac{v_k^{(j)}}{\triangle v_k^{(j)}} : \triangle v_k^{(j)} < 0 \right\}
\end{aligned}$$

that represent the maximal steps sizes autorized to keep the positivity of the respective variables, with $\gamma \in (0, 1]$.

Once the maximal steps sizes are computed we set,

$$\Upsilon^{\max} = \min \left\{ \alpha_{w_k}^{\max}, \alpha_{y_k}^{\max}, \alpha_{z_k}^{\max}, \alpha_{\lambda_{I,k}}^{\max}, \alpha_{s_k}^{\max}, \alpha_{v_k}^{\max} \right\}$$

a backtracking line search is performed in order to decrease the L_1 merit function

$$\begin{aligned}
\mathcal{M}_{\sigma\mu}(x, w, y, z) = \bar{f}_\mu(x, w, y, z) + \sigma \parallel Ax - b \parallel + \sigma \parallel c(x) - w \parallel \\
+ \sigma \parallel x - y + z \parallel,
\end{aligned} \tag{64}$$

where σ is a positive penalty parameter. This parameter is updated at each iteration. Clearly the step size Υ_k is computed in $(0, \Upsilon^{\max}]$ and verifies the Armijo's condition,

$$\mathcal{M}_{\sigma\mu}(\Pi_k + \Upsilon_k \triangle\Pi) \leq \mathcal{M}_{\sigma\mu}(\Pi_k) + \beta\alpha_k \mathcal{M}'_{\sigma\mu}(\Pi_k; \triangle\Pi) \tag{65}$$

with β a constant in $(0, \frac{1}{2})$. If the condition (65) is not satified, then a new step size is taken in $[\beta_1\Upsilon_k, \beta_2\Upsilon_k]$, with $\beta_1, \beta_2 \in (0, 1)$ and $\beta_1 < \beta_2$. The procedure is repeated until (65) is satisfied. Then the new primal-dual iterate is,

$$\Pi_{k+1} = \Pi_k + \Upsilon_k \triangle\Pi.$$

The distance of the point Π_k to central path is mesured using the euclidian norm of the perturbed optimality conditions, that is $\| F_\mu(\Pi_k) \|$ less than a given precision ε; the parameter μ is decreased and the process is repeated until μ becomes zero.

5.2 The interior point with DC regularization algorithm algorithm for the problem $(\mathcal{P}_{\mathcal{EI}})$

We will now describe the main lines of IPDCA to solve the problem $(\mathcal{P}_{\mathcal{EI}})$.

Algorithm 1 A primal-dual interior point algorithm(IPDCA)

1. Chose Π_0, $\beta \in (0, \frac{1}{2})$, $\theta \in (0, 1)$ and σ_0 and a precision ε. Set $k = 0$.
2. Compute the step $\triangle\Pi_k$ using (52)–(63).
3. Compute the step size Υ_k using the condition (65), if there is no step size satisfying (65) then stop, else set

$$\Pi_{k+1} = \Pi_k + \Upsilon_k \triangle\Pi.$$

4. If $\| F_\mu(\Pi_k) \| \leq \varepsilon$ then stop, else update $\mu_{k+1} = \theta\mu_k$ and update σ_k set $k = k + 1$ goto step 2.

An important point in the algorithm just presented is the rule to update the penalty parameter σ_k. In the present version of IPDCA we have adopted the following rule,

if $\sigma_{k-1} \geq \| \tilde{\lambda}_k \| + \bar{\sigma}$ **then**
  $\sigma_k = \sigma_{k-1}$
else
  $\sigma_k = \max\{\zeta\sigma_{k-1}, \| \tilde{\lambda}_k \| + \bar{\sigma}\}$
end

with $\zeta > 1$, $\bar{\sigma}$ a given positive number and

$$\tilde{\lambda}_k = \lambda_{I,k} + \triangle\lambda_I + \lambda_{E,k} + \triangle\lambda_E + s_k + \triangle s.$$

Remark 2 In our implementation of IPDCA for solving the convex problem $(\mathcal{M})$ we first reformulate it as a false DC program of the form

$$(\mathcal{P}_{\mathcal{DC}}) \begin{cases} \min f(x) = g(x) - h(x), \\ Ax - b = 0, \\ c(x) \geq 0, \\ x \in \mathbb{R}^n, \end{cases}$$

where g is a strongly convex function $g(x) = f(x) + \frac{\rho}{2} \| x \|^2$ and $h(x) = \frac{\rho}{2} \| x \|^2$, with ρ being a positve number, we take $\rho = 10e^{-3}$.

6 Illustrative example

The IPDCA was coded in standard C, both codes were tested on a PC machine with 4096MB of RAM and a processor of 2200MHz with SuSE 9.1. The stopping precision for Lancelot and IPDCA is $10e^{-4}$.

To validate the presented method, we consider the behaviour of a sheet with circular hole subjected to biaxial tension. For symmetry reasons, only a quarter of the unit cell is considered where the geometry and the adopted mesh are shown in Fig. 1. Here, SOLID45 is used, where each element is defined by eight nodes having three degrees of freedom at each node. The adopted geometry and mechanical characteristics are:

$L = 50\,\text{mm}, \quad r = 10\,\text{mm}, \quad h = 2\,\text{mm}, \quad E = 2.1 \times 10^5\,\text{MPa}, \quad \sigma_Y = 2.8 \times 10^2\,\text{MPa}, \nu = 0.3$

For this example, the following loading conditions are investigated:

(a) p_x and p_y increase proportionally (model with one point);
(b) Both loads p_x and p_y vary independently (model with four points).

The problem characteristics are presented in Table 1. The results of the loading cases (a) and (b), corresponding to limit and shakedown analysis, respectively are given in Table 2 and the computing time is in seconds. The results of the loading case (b) are also presented in Fig.2

It can be observed in Table 2 that, the safety factor (loading factor α), obtained with the IPDCA method agree with the result of the standard code Lancelot. The great

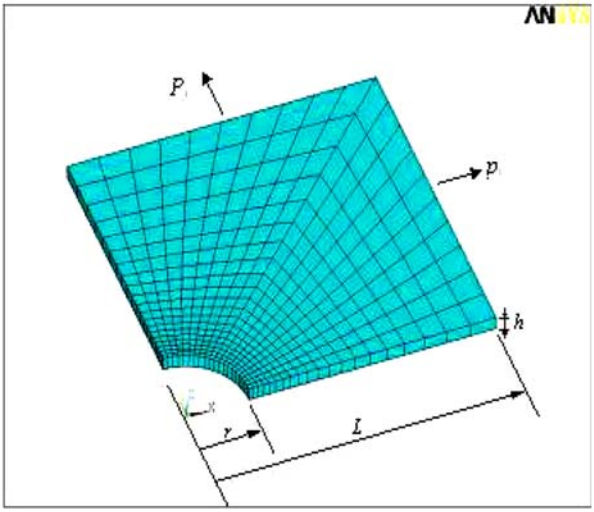

Fig. 1 Sheet with circular hole

Table 1 Test problems charateristics

Name	NG	NK	Nbr of var	Nbr of const	Nbr of charges	Nbr of points
(a)	1536	442	18433	12078	2	1
(b)	–	–	46081	44334	2	4

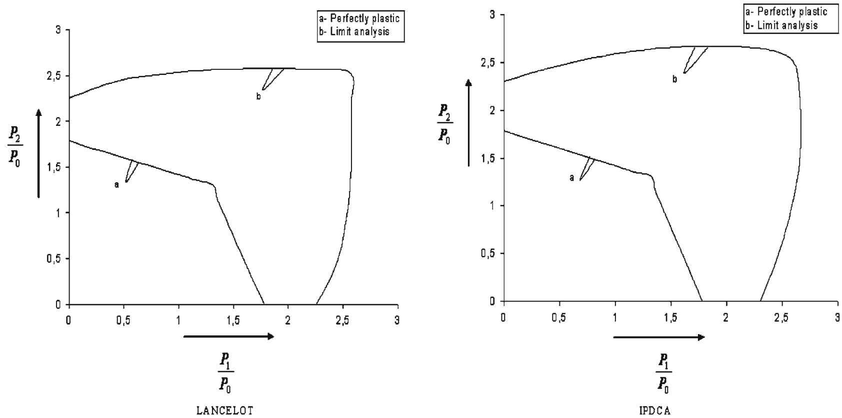

Fig. 2 Results of Lancelot and IPDCA for the loading case (b)

Table 2 Comparison between IPDCA and Lancelot

φ	Safety factor		CPU times		φ	Safety factor		CPU times	
	Lancelot	IPDCA	Lancelot	IPDCA		Lancelot	IPDCA	Lancelot	IPDCA
0°	2.258	2.302	43,200	190	0°	1.783	1.783	28,800	117
10°	2.476	2.487	65,212	137	10°	1.692	1.701	41,600	115
20°	2.691	2.740	65,120	136	20°	1.666	1.674	41,540	115
30°	2.972	3.072	64,930	128	30°	1.691	1.700	41,512	112
40°	3.360	3.461	64,850	113	40°	1.771	1.780	40,987	109
45°	3.576	3.588	64,595	82	45°	1.836	1.846	41,512	114
50°	3.360	3.462	65,245	211	50°	1.780	1.781	41,410	106
60°	2.972	3.071	65,876	234	60°	1.700	1.701	41,554	111
70°	2.691	2.737	66,005	261	70°	1.674	1.674	41,609	117
80°	2.476	2.484	66,223	365	80°	1.701	1.701	41,612	119
90°	2.258	2.300	66,798	479	90°	1.783	1.783	41,620	110

advantage of the IPDCA method becomes obvious, if we consider the computing time and the required memory spaces.

Acknowledgements The authors A. Hachemi and S. Mouhtami gratefully acknowledge the support provided by the Deutsche Forschungsgemeinschaft (DFG grant no. WE 736/22). This work was done within the scientific collaboration between the Institut für Allgemeine Mechanik, RWTH-Aachen, Germany and Equipe Modélisation, Optimisation et Recherche Opérationnelle, Insa-Rouen, France.

References

1. Melan, E.: Theorie Statisch Unbestimmter Systeme aus Ideal-Plastischem Baustoff. Sitber. Akad. Wiss. Wien, Abt. IIA **145**, 195–218 (1936)
2. Koiter, W.T.: General theorems for elastic-plastic solids. In: Sneddon, I. N., Hill, R. (eds.) Progress in Solid Mechanics, pp. 165–221. North-Holland, Amsterdam, (1960)
3. Hachemi, A., Weichert, D.: Numerical shakedown analysis of damaged structures. Comput. Methods Appl. Mech. Eng **160**, 57–70 (1998)

4. Akoa, F., Le Thi Hoai An, Pham Dinh Tao: An interior point algorithm with DC regularisation for nonconvex quadratic programming. In: Le Thi Hoai An, Pham Dinh Tao (eds.) Modelling, Computation and Optimization in Information Systems and Management Sciences, pp. 87–96. Hermes Science, London (2004)
5. Hachemi, A., Le Thi Hoai An, Mouhtamid, S., Pham Dinh Tao: Large-scale nonlinear programming and lower bound direct method in engineering applications. In: Le Thi Hoai An, Pham Dinh Tao (eds.) Modelling, Computation and Optimization in Information Systems and Management Sciences, pp. 299–310. Hermes Science, London(2004)
6. Conn, A.R., Gould, N. I. M., Toint, Ph.L.: LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A). Springer -Verlag, Berlin (1992)
7. Weichert, D., Hachemi, A., Schwabe, F.: Shakedown analysis of composites. Mech. Res. Comm. **26**, 309–318 (1999)
8. Hachemi, A., Schwabe, F., Weichert, D.: Failure investigation of fiber-reinforced composite materials by shakedown analysis. In: Weichert, D., Maier, G. (eds.) Inelastic Analysis of Structures under Variable Loads: Theory and Engineering Applications pp. 107–119. Kluwer Academic Publishers, Dordrecht (2000)
9. König, J. A., Kleiber, M.: On a new method of shakedown analysis. Bull. Acad. Polon. Sci. Sér. Sci. **26**, 165–171 (1978)
10. Akoa, F.: Approches de point intérieur et de la programmation DC en optimisation non convexe. Codes et simulations numériques industrielles. PhD Thesis, Insa - Rouen, January 2005.
11. Le Thi Hoai An: Analyse numérique des algorithmes de l'optimization D.C: Approches locale et globale, PhD Thesis, University of Rouen, December 1994.
12. Le Thi Hoai An, Pham Dinh Tao: DC programming: theory, algorithms and applications. the state of the art (28 pages), In: Proceedings (containing the refereed contributed papers) of The First International Workshop on Global Constrained Optimization and Constraint Satisfaction (Cocos' 02), Valbonne-Sophia Antipolis, France, October 2–4, 2002.
13. Le Thi Hoai An: Contribution à l'optimisation non convexe et l'optimisation globale: Théorie, Algorithmes et Applications. Habilitation à Diriger des Recherches (HDR), Université de Rouen, Juin 1997.
14. Pham Dinh Tao, Le Thi Hoai An: Convex analysis approach to DC programming: theory, algorithms and applications. Acta Mathematica Vietnamica (dedicated to Professor Hoang Tuy on the occasion of his 70th birthday) **22**(1), 289–355 (1997)
15. Pham Dinh Tao, Le Thi Hoai An: DC optimization algorithm for solving the trust region problem. SIAM J. Optim., **8**(2), 476–505 (1998)
16. Pham Dinh Tao, Le Thi Hoai An: Large scale molecular optimization from distance matrices by a DC optimization approach. SIAM J. Optim, 14(1), 77–116 (2003)
17. Le Thi Haoi An, Pham Dinh Tao: The DC (difference of convex functions) Programming and DCA revisited with DC models of real world nonconvex optimization problems. Ann. Oper Res. **133**, 23–46 (2005)
18. Altman, A., Gondzio, J.: Regularized symmetric Indefinite Systems in Interior Point Methods for Linear and Quadratic Optimization. Logilab Technical Report 1998.6.
19. Saunders, M.: Cholesky-based methods for sparse least squares: the benefits of regularization. In: Adams, L., Nazareth, L. (eds.) Linear and Nonlinear Conjugate Gradient-Related Methods pp. 92–100. SIAM Philadelphia, USA (1996)
20. Saunders, M., Tomlin, J.A.: Solving regularized linear programs using barrier methods and KKT systems. Technical Repport SOL 96-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305, USA, December 1996.
21. Vanderbei, R.J., Shanno, D.: An Interior-Point Algorithm for Nonconvex Nonlinear Programming. Statistics and Operations Research, Princeton SOR-97-21 Comput. Optim. App. 13, 231–252 (1999)
22. Vanderbei, R.J.: An Interior-Point Algorithm for Quadratic Programming. Statistics and Operations Research, Princeton SOR-94-15 (1994)
23. Vanderbei, R.J.: Symmetric quasi-definite matrices. SIAM J. Optim **5**(1), 100–113 (1995)