



**HAL**  
open science

# A feasible direction interior point algorithm for general nonlinear semidefinite programming

Jean Rodolphe Roche, José Herskovits, Elmer Bazán

## ► To cite this version:

Jean Rodolphe Roche, José Herskovits, Elmer Bazán. A feasible direction interior point algorithm for general nonlinear semidefinite programming. 2015. hal-01272994

**HAL Id: hal-01272994**

**<https://hal.science/hal-01272994>**

Preprint submitted on 11 Feb 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Feasible Direction Interior Point Algorithm for General Nonlinear Semidefinite Programming

Jean Rodolphe Roche · José Herskovits · Elmer Bazán

**Abstract** This paper deals with nonlinear smooth optimization problems with equality and inequality constraints, as well as semidefinite constraints on symmetric matrix-valued functions. We present first a new semidefinite programming algorithm that takes advantage of the structure of the matrix constraints. This is relevant in applications where these matrices have a favorable structure, as in the case when finite element models are employed. FDIPA\_GSDP is then obtained by integration of this new method with the well known Feasible Direction Interior Point Algorithm for nonlinear smooth optimization, FDIPA. FDIPA\_GSDP makes iteration in the primal and dual variables to solve the first order optimality conditions. Given an initial interior point, FDIPA\_GSDP generates a descent interior sequence, converging to a local solution of the problem. At each iteration a feasible descent direction is defined. A line search along this direction looks for a new interior point with a lower objective. Global convergence to stationary points is proved. Some structural optimization test problems were solved very efficiently, without need of parameters tuning.

**Keywords** structural optimization, nonlinear optimization, semidefinite programming, feasible directions, interior-point methods.

---

Jean R. Roche  
I.E.C.L., Université de Lorraine, CNRS, INRIA, B.P. 70239,  
54506 Vandoeuvre lès Nancy, France.  
E-mail: jean-rodolphe.roche@univ-lorraine.fr

José Herskovits, Elmer Bazán  
Mechanical Engineering Program - COPPE - Federal University of Rio de Janeiro, PO Box 68503,21941-972, Rio de Janeiro, Brazil.  
E-mail: jose@optimize.ufrj.br, elmer@optimize.ufrj.br

## 1 Introduction

We deal with the nonlinear optimization problem:

$$\begin{aligned} \min_x & f(x); x \in \mathbb{R}^n \\ \text{s.t.} & g(x) \leq 0, \\ & h(x) = 0, \\ & \mathcal{A}(x) \preceq 0. \end{aligned} \quad (1)$$

where  $f(x) \in \mathbb{R}$ ,  $g(x) \in \mathbb{R}^m$ ,  $h(x) \in \mathbb{R}^p$  and  $\mathcal{A}(x) \in \mathbb{S}^{q \times q}$  are smooth functions, not necessarily linear or convex. We call  $\mathbb{S}^q$  the set of real symmetric matrices of size  $q \times q$ .  $\mathcal{A}(x) \preceq 0$  means that  $\mathcal{A}(x)$  is negative semidefinite.

Problem (1) is referred as a General Nonlinear Semidefinite Program, the constraints  $g(x) \leq 0$  and  $h(x) = 0$ , as vector inequality and equality constraints respectively and  $\mathcal{A}(x) \preceq 0$  is a matrix constraint.

We call  $\Omega = \{x \in \mathbb{R}^n; \mathcal{A}(x) \preceq 0, g(x) \leq 0\}$  the set of feasible solutions with respect to inequalities and  $\text{int}(\Omega)$ , its interior. For a real number  $a$ , we denote  $\Omega_a = \{x \in \Omega \text{ such that } f(x) \leq a\}$ .

In a first stage we propose a new algorithm to solve the single SDP problem:

$$\begin{aligned} \min_x & f(x); x \in \mathbb{R}^n \\ \text{s.t.} & \mathcal{A}(x) \preceq 0. \end{aligned} \quad (2)$$

This algorithm follows similar ideas to those presented in [Aroztegui,..], but using a formulation that preserves the structure of the matrix constraints. In particular, two linear systems with the same matrix have to be solved at each iteration of the algorithm. We also present a formulation that takes advantage of the structure of the matrix constraints to solve more efficiently the internal linear systems of the algorithm.

This fact is relevant in several applications, in particular when partial differential equations are solved with

numerical methods, like Finite Elements or Finite Differences. In the case of structural optimization, computer solvers employ particular techniques that take advantage of the particular structure of stiffness and mass matrices.

In a second stage we propose a new algorithm to solve the generalized problem (1), inspired by the feasible direction interior point method for standard nonlinear programming described in [xxFDIPA].

Even if inequality vector constraints can be treated as diagonal matrix constraints, our approach deals naturally with them and also with equality constraints.

Given an initial feasible point, the present algorithm produces a decreasing sequence of feasible points that converges to a local solution of the problem. For this, at each iteration a feasible descent direction is first computed. Then, a new interior point with lower objective is obtained by means of a line search.

In [Aroztegui] several references about linear and nonlinear semidefinite programming are cited, and also some applications in structural optimization. The range of structural optimization problems that can be solved by using the formulation of the problem (1) is quite large. Local constraints can be considered, as bounds on local stresses or nodal displacements, and also constraints on the fundamental frequencies or on the structural compliance. In free material optimization the elasticity matrices are constrained to be positive definite [xx].

This paper is organized as follows. Section 2 is devoted to the presentation of some required notation and definitions and also auxiliary results needed for our study. In section 3 we introduce an SDP algorithm which preserves the structure of the matrix constraints. Then in section 4 we demonstrate the global convergence of this algorithm. Some implementation issues are described in section 5, in particular how to take advantage of the structure of the matrix constraint. In section 6 we extend the domain of application of the algorithm presented in section 3, we consider the case where we have, SDP constraints, vector inequalities and equalities constraints. Some models employed in structural optimization are described and solved with the present techniques. Following we discuss the numerical results. Finally, our conclusions about the present technique, their numerical implementation and our numerical results are described.

## 2 Basic concepts

In this section we focus Problem (2) and describe some basic concepts and theoretical results related to the present method.

### 2.1 Notation

Let  $\mathbb{R}^{q \times n}$  denote the space of  $q \times n$  real matrices. The sets of symmetric positive semidefinite and positive definite matrices of size  $q \times q$  are denoted  $\mathbb{S}_+^q$  and  $\mathbb{S}_{++}^q$ , respectively. Negative semidefinite and definite matrices are defined in a similar way.

The symbol  $\preceq$  refers to a partial order on the negative semidefinite matrices, that is,  $A \preceq B$  means that  $A - B$  is negative semidefinite, [5]. Similarly, the symbol  $\prec, \succ$  and  $\succcurlyeq$  refers to a partial order on the negative definite, positive semidefinite and positive definite matrices, respectively.

Let us define the two following maps:

$$vec(A) = [a_{11} \dots a_{q1}, a_{12}, \dots, a_{q2}, \dots, a_{1q} \dots a_{qq}]^\top,$$

$vec : \mathbb{R}^{q \times q} \rightarrow \mathbb{R}^{q^2}$ , and  $mat : \mathbb{R}^{q^2} \rightarrow \mathbb{R}^{q \times q}$ , the inverse of  $vec$ , see [3]. Here,  $a_{ij}$  is the  $(i, j)$ th entry of a matrix  $A \in \mathbb{R}^{q \times q}$ . Then, the inner product

$$\langle A, B \rangle = tr(A^\top B) = vec(A)^\top vec(B), \text{ for } A, B \in \mathbb{R}^{q \times q}.$$

The Kronecker product of two matrices  $A \in \mathbb{R}^{p \times q}$   $B \in \mathbb{R}^{r \times s}$  is denoted by  $A \otimes B$  and is defined as:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1q}B \\ a_{21}B & \dots & a_{2q}B \\ \vdots & \dots & \vdots \\ a_{p1}B & \dots & a_{pq}B \end{bmatrix}$$

**Lemma 1** *The followings properties are true, [3]:*

$$i) \forall A \in \mathbb{R}^{p \times q}, B \in \mathbb{R}^{r \times s} \text{ and } C \in \mathbb{R}^{s \times q},$$

$$(A \otimes B)vec(C) = vec(BCA^\top)$$

$$ii) \forall A \in \mathbb{R}^{p \times q} \text{ and } B \in \mathbb{R}^{r \times s}$$

$$(A \otimes B)^\top = (A^\top \otimes B^\top)$$

$$iii) \forall A \in \mathbb{R}^{p \times q}, B \in \mathbb{R}^{r \times s} \text{ and } C \in \mathbb{R}^{q \times k}, D \in \mathbb{R}^{s \times l},$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$$

$$iv) \forall A \in \mathbb{R}^{p \times p} \text{ and } B \in \mathbb{R}^{r \times r},$$

$$(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1})$$

for  $A$  and  $B$  invertible. □

The partial derivative of  $\mathcal{A}(x)$  with respect to  $x_k$ ,  $k = 1, \dots, n$  is denoted by  $\frac{\partial \mathcal{A}}{\partial x_k}(x)$ , with components  $\frac{\partial a_{ij}(x)}{\partial x_k}$ ,  $i, j = 1, \dots, q$ .

We define  $\nabla\mathcal{A}(x)$  in  $\mathbb{R}^{n \times q^2}$ , as

$$\nabla\mathcal{A}(x) = \begin{bmatrix} \text{vec} \left( \frac{\partial\mathcal{A}}{\partial x_1}(x) \right)^\top \\ \vdots \\ \text{vec} \left( \frac{\partial\mathcal{A}}{\partial x_n}(x) \right)^\top \end{bmatrix}. \quad (3)$$

The derivative of  $\mathcal{A}$  in the direction  $d \in \mathbb{R}^n$  at  $x$ , denoted by  $D\mathcal{A}(x)d$ , is:

$$D\mathcal{A}(x)d = \sum_{p=1}^n d_p \frac{\partial\mathcal{A}}{\partial x_p}(x). \quad (4)$$

In consequence, we have

$$\text{vec}(D\mathcal{A}(x)d) = \nabla\mathcal{A}(x)^\top d. \quad (5)$$

The Lagrangian of problem (2) is

$$L(x, \Lambda) = f(x) + \langle \mathcal{A}(x), \Lambda \rangle; \quad L : \mathbb{R}^n \times \mathbb{R}^{q \times q} \rightarrow \mathbb{R}$$

Alternatively, the Lagrangian can be written as follows,

$$L(x, \lambda) = f(x) + \langle \text{vec}(\mathcal{A}(x)), \lambda \rangle; \quad L : \mathbb{R}^n \times \mathbb{R}^{q^2} \rightarrow \mathbb{R}$$

where  $\lambda = \text{vec}(\Lambda)$ . Then

$$\nabla_x L(x, \lambda) = \nabla f(x) + \nabla\mathcal{A}(x)\lambda, \quad (6)$$

see [15] and [2].

## 2.2 Definitions

We give now some definitions related to semidefinite programming extension of Karush-Kuhn-Tucker first order optimality conditions, [14] and [15].

Let  $\{b_1(x), \dots, b_r(x)\} \subset \mathbb{R}^{q^2}$  be an orthonormal basis of  $\ker(\mathcal{A}(x) \otimes I)$ .

**Definition 1** A point  $x$  is a regular point of problem (2) if the vectors  $\nabla\mathcal{A}(x)b_i(x) : i = 1, \dots, r$  are linearly independent.

**Definition 2** A regular point  $x$  is a stationary point of problem (2) if there exists  $\Lambda \in \mathbb{R}^{q \times q}$  such that the following conditions are verified:

$$\begin{aligned} \nabla_x L(x, \Lambda) &= 0 \\ \Lambda\mathcal{A}(x) &= 0 \\ \mathcal{A}(x) &\preceq 0 \end{aligned} \quad (7)$$

**Definition 3** If  $\Lambda \succcurlyeq 0$  we say that a stationary point of problem (2) is a Karush-Kuhn-Tucker point.

The following definitions are related with the method proposed in this paper.

**Definition 4** We call a vector  $d \in \mathbb{R}^n$  a feasible direction of  $\Omega$  at  $x \in \Omega$  if there exists  $\tau > 0$  such that  $x + td \in \Omega$  for all  $t \in [0, \tau]$ .

**Definition 5** The vector field  $d(x)$  defined on  $\Omega \subseteq \mathbb{R}^n$  is said to be a uniformly feasible direction field of  $\Omega$  if there exists  $\tau > 0$  such that,  $x + td(x) \in \Omega$  for all  $x \in \Omega$  and all  $t \in [0, \tau]$ .

When the vector field  $d(x)$  is a uniformly feasible direction field of  $\Omega$ , the segment  $[x, x + \tau d(x)]$  is included in  $\Omega$  for all  $x \in \Omega$ .

**Definition 6** A vector  $d \in \mathbb{R}^n$  is a descent direction of a real function  $f$  at  $x \in \mathbb{R}^n$  if there exist some  $\delta > 0$  such that:  $f(x + td) < f(x)$  for all  $t \in (0, \delta]$ .

## 2.3 Some auxiliary results

**Lemma 2** Let  $A \in \mathbb{S}_{++}^q$  and  $B \in \mathbb{S}^q$ . If  $AB + BA \prec 0$ , then  $B \prec 0$ .

*Proof* See [1]. □

The proof of the following lemma is similar to the previous one.

**Lemma 3** Let  $A \in \mathbb{S}_{++}^q$  and  $B \in \mathbb{S}^q$ . If  $AB + BA \preceq 0$ , then  $B \preceq 0$ . □

**Lemma 4** If  $A, B \in \mathbb{S}^q$  and  $A \succ 0$ , the matrix  $AB$  has real eigenvalues and the same inertia as  $B$ . See [7]. □

**Lemma 5** Let  $A \in \mathbb{S}_{++}^q$  and  $B \in \mathbb{S}_-^q$ . If they commute, then

$$y^\top AB y = 0 \iff B y = 0.$$

*Proof* See [1]. □

**Lemma 6** Let  $A \in \mathbb{S}_{++}^q$  and  $B \in \mathbb{S}_-^q$ . If  $A$  and  $B$  commute, then

$$(A \otimes I)^{-1}(B \otimes I) \in \mathbb{S}_-^{m^2}.$$

*Proof* Since  $A \succ 0$  and  $B \preceq 0$  and commute then there exist a regular matrix  $P \in \mathbb{R}^{q \times q}$  such that

$$\begin{aligned} A &= PD_A P^{-1}, \\ B &= PD_B P^{-1}. \end{aligned}$$

In view of Kronecker product properties, see for example lemma E.1 in [3] :

$$\begin{aligned} A \otimes I &= PD_A P^{-1}, \\ B \otimes I &= PD_B P^{-1}, \end{aligned}$$

where  $P = P \otimes P$ ,  $D_A = D_A \otimes I$  and  $D_B = D_B \otimes I$ .  $D_A$  and  $D_B$  are diagonal matrices and  $P$  is orthonormal. Therefore

$$(A \otimes I)^{-1}(B \otimes I) = PD_A D_B P^\top \in \mathbb{S}_-^{q^2}.$$

□

### 3 A SDP algorithm that preserves the structure of the matrix constraints

In a first stage we discuss the basic ideas leading to the present method. The formal algorithm is described later.

#### 3.1 Basic ideas

Let us consider the following assumptions about  $f$  and  $\mathcal{A}$ :

**Assumption 1** There exists a real number  $a$  such that  $\Omega_a$  is compact and  $\text{int}(\Omega_a) \neq \emptyset$ .

**Assumption 2** For any  $x \in \text{int}(\Omega_a)$ ,  $\mathcal{A}(x) \prec 0$ .

**Assumption 3**  $f$  and  $\mathcal{A}$  belong to  $C^1(\Omega_a)$ . In addition we assume that  $\nabla f$  and  $\frac{\partial \mathcal{A}}{\partial x_p}$  for  $p = 1, \dots, n$  are Lipschitz functions.

**Assumption 4** Any point  $x \in \Omega_a$  is a regular point of problem (2).

The present algorithm looks for a primal and dual feasible solution of the equations characterizing a critical point given in (7),

$$\begin{aligned} \nabla f(x) + \nabla \mathcal{A}(x)\lambda &= 0 \\ \text{vec}(\Lambda \mathcal{A}(x)) &= 0 \end{aligned} \quad (8)$$

where  $\lambda = \text{vec}(\Lambda)$ . We propose a Newton-like iteration tailored in such a way to have a minimizing sequence of strictly feasible points.

Remark that, since  $\mathcal{A}(x)$  is symmetric and  $\Lambda$  is also symmetric at a stationary point, the complementarity condition is equivalent to  $\text{svec}(\text{sym}(\Lambda \mathcal{A}(x))) = 0$ , where  $\text{svec}$  is defined in [Aroz]. The algorithm proposed in [Arozte] works with the symmetric part of the complementarity condition and with the symmetric parts of  $\mathcal{A}(x)$  and  $\Lambda$  as unknowns. These procedure reduces significantly the size of the nonlinear system to be solved, but destroys the structure of the involved matrices. In the present approach the nonlinear system is larger, but

the computer effort to solve it is dramatically reduced by taking advantage of the structure of the constraint matrix.

Considering now the definition of Kronecker product. We deduce,

$$\text{vec}(I \Lambda \mathcal{A}(x)) = [\mathcal{A}(x) \otimes I] \text{vec}(\Lambda)$$

and also,

$$\text{vec}(\Lambda \mathcal{A}(x) I) = [I \otimes \Lambda] \text{vec}(\mathcal{A}(x)).$$

Therefore, the Jacobian of the nonlinear system (8) is:

$$\begin{bmatrix} \nabla_{xx} L(x, \lambda) & \nabla \mathcal{A}(x) \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top & (\mathcal{A}(x) \otimes I) \end{bmatrix}. \quad (9)$$

In a similar way as in [6], instead of the Hessian of the Lagrangian  $\nabla_{xx} L(x, \lambda)$ , we can employ a positive definite matrix denoted  $B$ . This matrix  $B$  can be a quasi-Newton approximation, or even the identity matrix.

A Newton like iteration to solve (8) is given by the following linear system

$$\begin{aligned} & \begin{bmatrix} B & \nabla \mathcal{A}(x) \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top & (\mathcal{A}(x) \otimes I) \end{bmatrix} \begin{bmatrix} x_0 - x \\ \lambda_0 - \lambda \end{bmatrix} \\ &= - \begin{bmatrix} \nabla f(x) + \nabla \mathcal{A}(x)\lambda \\ \text{vec}(\Lambda \mathcal{A}(x)) \end{bmatrix} \end{aligned} \quad (10)$$

where  $(x, \Lambda) \in \text{int}(\Omega_a) \times \mathbb{S}_{++}^q$  is the current iterate and  $(x_0, \lambda_0) \in \mathbb{R}^n \times \mathbb{R}^{q^2}$  is the new estimate. Note that  $\lambda = \text{vec}(\Lambda)$  and  $\lambda_0 = \text{vec}(\Lambda_0)$ .

Let be  $d_0 = x_0 - x$ . Then we have,

$$\begin{aligned} B d_0 + \nabla \mathcal{A}(x)\lambda_0 &= -\nabla f(x) \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top d_0 + (\mathcal{A}(x) \otimes I)\lambda_0 &= 0 \end{aligned} \quad (11)$$

If  $d_0 = 0$  equation (11) becomes,

$$\nabla f(x) + \nabla \mathcal{A}(x)\lambda_0 = 0 \quad (12)$$

and

$$(\mathcal{A}(x) \otimes I)\lambda_0 = 0 \quad (13)$$

Since  $\mathcal{A}(x) \prec 0$ , the matrix  $\mathcal{A}(x) \otimes I$  is nonsingular and we have  $\Lambda_0 = \text{mat}(\lambda_0) = 0$ . This proves that  $\Lambda_0 \mathcal{A}(x) = 0$ .

Therefore,  $\nabla f(x) = 0$  and  $x$  is a stationary point of the problem.

We shall prove, in the case when  $d_0 \neq 0$ , that  $d_0$  is a descent direction of the objective function.

However when  $x$  is on the boundary of  $\Omega$  we cannot ensure that  $d_0$  is a feasible direction. In effect, when  $x$  is on the boundary of  $\Omega$ , it follows from (11) that  $d_0$  is

tangent to  $\Omega$ . Thus, depending on the curvature of the boundary of  $\Omega$ ,  $d_0$  can point outwards of the feasible domain.

To obtain a feasible direction, as in [6], we modify the previous linear system introducing an appropriate term in the right hand side:

$$\begin{aligned} Bd + \nabla \mathcal{A}(x)\bar{\lambda} &= -\nabla f(x) \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top d + (\mathcal{A}(x) \otimes I)\bar{\lambda} &= -\rho\lambda \end{aligned} \quad (14)$$

where  $\rho$  is a positive real number and  $\lambda = \text{vec}(A)$ .

In the next section we prove that in fact  $d$  constitutes an uniformly feasible directions field in the sense of definition 5.

We introduce now the following assumption on  $A$ .

**Assumption 5** At each iteration the current values of  $A$  and  $\mathcal{A}(x)$  commute.

The pair  $(d, \bar{\lambda})$  obtained by the linear system (14) can also be computed solving

$$\begin{aligned} Bd_1 + \nabla \mathcal{A}(x)\lambda_1 &= 0 \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top d_1 + (\mathcal{A}(x) \otimes I)\lambda_1 &= -\lambda \end{aligned} \quad (15)$$

and taking,

$$d = d_0 + \rho d_1 \quad (16)$$

$$\bar{\lambda} = \lambda_0 + \rho \lambda_1. \quad (17)$$

The descent direction  $d_0$  verifies  $d_0^\top \nabla f(x) < 0$ . For a given  $\xi \in (0, 1)$ , we get an upper bound for  $\rho$  such that

$$d^\top \nabla f(x) \leq \xi d_0^\top \nabla f(x). \quad (18)$$

Consequently, the feasible direction  $d$  will be a descent direction also.

In fact, if  $d_1^\top \nabla f(x) > 0$ , we take

$$\rho \leq (\xi - 1) \frac{d_0^\top \nabla f(x)}{d_1^\top \nabla f(x)}.$$

Otherwise, we choose

$$\rho \leq \varphi \|d_0\|^2,$$

for some fixed parameter  $\varphi > 0$ .

Once we have computed a descent and feasible direction  $d$ , we can determine the next iterate performing a line search along  $d$  to get feasibility and an appropriate reduction of the objective function.

We shall prove global convergence to a stationary point, for any way of updating  $B$  and  $\Lambda$ , provided they are positive definite and  $\Lambda$  satisfies assumption 5.

### 3.2 The Algorithm for SDP

Now we write down the specific steps of the present algorithm for semidefinite programming:

**Parameters.**  $\xi \in (0, 1)$ ,  $\eta \in (0, 1)$ ,  $\varphi > 0$  and  $\nu \in (0, 1)$ .

**Initial data.**  $x \in \text{int}(\Omega_a)$ ,  $\Lambda \in \mathbb{S}_{++}^q$  commuting with  $\mathcal{A}(x)$  and  $B \in \mathbb{S}_{++}^n$ .

**Step 1.** Computation of the feasible descent search direction  $d$ .

(i) Solve the linear systems (11) and (15):

If  $d_0 = 0$ , stop.

(ii) Compute the positive scalar  $\rho$  such that

$$\rho = \min \left\{ \varphi \|d_0\|^2, (\xi - 1) \frac{d_0^\top \nabla f(x)}{d_1^\top \nabla f(x)} \right\} \quad (19)$$

if  $d_1^\top \nabla f(x) > 0$ . Otherwise:

$$\rho = \varphi \|d_0\|^2. \quad (20)$$

(iii) Compute the search direction  $d$  as

$$d = d_0 + \rho d_1. \quad (21)$$

**Step 2.** Line Search.

Find  $t$ , the first element of  $\{1, v, v^2, v^3, \dots\}$  such that

$$f(x + td) \leq f(x) + t\eta d^\top \nabla f(x) \quad (22)$$

and

$$\mathcal{A}(x + td) \prec 0. \quad (23)$$

**Step 3.** Updates.

(i) Set  $x := x + td$ .

(ii) Update  $B \in \mathbb{S}_{++}^n$ .

(iii) Define new value for  $\Lambda \in \mathbb{S}_{++}^q$  commuting with  $\mathcal{A}(x)$ .

(iv) Go to Step 1.

□

The line search is performed adapting Armijo's algorithm to constraint optimization problems. Extensions of Wolfe or Goldstein line search criteria [9] can be also employed.

The update of  $B$  and  $\Lambda$  computed in **Step 3.** must verify the following assumptions:

**Assumption 6** There exist positive numbers  $\lambda^I$  and  $\lambda^S$  such that

$$\lambda^I I \preceq \Lambda \preceq \lambda^S I$$

**Assumption 7** There exist positive numbers  $\sigma_1$  and  $\sigma_2$  such that

$$\sigma_1 I \preceq B \preceq \sigma_2 I$$

#### 4 Global convergence

This section is devoted to prove the convergence of the present algorithm for any initial point  $x^0 \in \text{int}(\Omega)$ . If assumptions 1 to 7 previously introduced are verified at each step, we show that the present algorithm generates a sequence  $\{(x^k, A_0^k)\}$  converging to  $(x^*, A_0^*)$  where  $x^*$  is a stationary point of problem (2).

First we prove that all the steps of the algorithm are well posed. In particular we show that the linear systems has an unique solution. Then it is shown that at each iteration  $d_0$  and  $d$  are descent directions of  $f$  at  $x$  and  $d(x)$  constitutes an uniformly feasible directions field in  $\Omega_a$ . Finally we state that any sequence generated by the algorithm converges to a stationary point of (2).

**Theorem 1** *Assume that  $x \in \Omega_a$  is a regular point of problem (2),  $B \in \mathbb{S}_{++}^n$ ,  $\Lambda \in \mathbb{S}_{++}^q$  and  $\Lambda$  and  $\mathcal{A}(x)$  commute. Then, the following matrix is regular,*

$$W(x, B, \Lambda) = \begin{bmatrix} B & \nabla \mathcal{A}(x) \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top & \mathcal{A}(x) \otimes I \end{bmatrix} \quad (24)$$

*Proof* It is enough to prove that, if for some  $v \in \mathbb{R}^{n+q^2}$  it is  $W(x, B, \Lambda)v = 0$ , then  $v = 0$ . Let  $v^\top = [r^\top, y^\top]$ ,  $r \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^{q^2}$ . Thanks to  $B \in \mathbb{S}_{++}^n$  and using Gaussian elimination we have:

$$r = -B^{-1} \nabla \mathcal{A}(x) y \quad (25)$$

$$M y = 0 \quad (26)$$

where  $M$  is the Schur complement:

$$M = \nabla \mathcal{A}(x)^\top B^{-1} \nabla \mathcal{A}(x) - (I \otimes \Lambda)^{-1} (\mathcal{A}(x) \otimes I).$$

Since  $M$  is symmetric, to conclude that is nonsingular, we show that it is positive definite.

Performing the product  $y^\top M y$  we have

$$y^\top M y = y^\top (\nabla \mathcal{A}(x)^\top B^{-1} \nabla \mathcal{A}(x)) y - y^\top (I \otimes \Lambda)^{-1} (\mathcal{A}(x) \otimes I) y \quad (27)$$

Since  $B^{-1}$  is positive definite,

$$y^\top (\nabla \mathcal{A}(x)^\top B^{-1} \nabla \mathcal{A}(x)) y \geq 0 \quad (28)$$

By hypothesis and lemma 6

$$-y^\top (I \otimes \Lambda)^{-1} (\mathcal{A}(x) \otimes I) y \geq 0 \quad (29)$$

concluding that  $y^\top M y \geq 0$ .

Now, suppose that  $y^\top M y = 0$ . We must prove that  $y = 0$ . From (27), (28) and (29) we have

$$y^\top \nabla \mathcal{A}(x)^\top B^{-1} \nabla \mathcal{A}(x) y = 0 \quad (30)$$

and

$$y^\top (I \otimes \Lambda)^{-1} (\mathcal{A}(x) \otimes I) y = 0. \quad (31)$$

Since  $B$  is positive definite, from equation (30) we have

$$\nabla \mathcal{A}(x) y = 0. \quad (32)$$

Due to lemma 5 and from equation (31),

$$(\mathcal{A}(x) \otimes I) y = 0. \quad (33)$$

Since  $x$  is a regular point we have  $y = 0$ .  $\square$

As a consequence of the last theorem, since  $\Omega_a$ ,  $\Lambda$  and  $B$  are bounded, we have that the values of  $d_0$ ,  $\lambda_0$ ,  $d_1$  and  $\lambda_1$  are bounded also.

If in Step 1  $d_0 = 0$ , we have  $\lambda_0 = 0$  and  $\nabla f(x) = 0$ . Then, the computed iterate is a local interior optimal solution and the algorithm stops. If not, we are going to show that  $d_0$  is a descent direction.

**Lemma 7** *The vector  $d_0$  computed by the algorithm is a descent direction and*

$$d_0^\top \nabla f(x) \leq -d_0^\top B d_0.$$

*Proof* Multiplying the first equation of (11) by  $d_0^\top$ ,

$$d_0^\top \nabla f(x) = -d_0^\top B d_0 - d_0^\top \nabla \mathcal{A}(x) \lambda_0.$$

In view of the second equation of (11),

$$-d_0^\top \nabla \mathcal{A}(x) \lambda_0 = \lambda_0^\top (\mathcal{A}(x) \otimes I) (I \otimes \Lambda)^{-1} \lambda_0.$$

Then,

$$d_0^\top \nabla f(x) = -d_0^\top B d_0 + \lambda_0^\top (\mathcal{A}(x) \otimes I) (I \otimes \Lambda)^{-1} \lambda_0.$$

Thanks to assumptions 5 and 7,  $(I \otimes \Lambda)^{-1} (\mathcal{A}(x) \otimes I) \in \mathbb{S}_-^{q^2}$  and  $B \in \mathbb{S}_{++}^n$  then,

$$d_0^\top \nabla f(x) \leq -d_0^\top B d_0.$$

As a consequence,  $d_0 \neq 0$  is a descent direction of  $f$  at  $x$ .  $\square$

**Lemma 8** *The vector  $d$  computed by the algorithm is a descent direction and satisfies*

$$d^\top \nabla f(x) \leq \xi d_0^\top \nabla f(x). \quad (34)$$

*Proof* see [1].  $\square$

**Lemma 9** *There exists  $\delta \geq 1$  and  $\varphi_0 > 0$  such that the search direction  $d$  and the parameter  $\rho$  computed by the algorithm verify:*

$$\varphi_0 \|d_0\|^2 \leq \rho \leq \varphi \|d_0\|^2 \quad (35)$$

and

$$\|d\| \leq \delta \|d_0\|. \quad (36)$$

*Proof* see [1] □

As a consequence of (35) and (36),  $\rho$  and  $\|d\|^2$  have the same order of magnitude. Thus, there exist  $\varphi_0 > 0$  and  $\varphi > 0$  such that

$$\varphi_0 \|d(x)\|^2 \leq \rho(x) \leq \varphi \|d(x)\|^2, \quad x \in \Omega_a. \quad (37)$$

**Lemma 10** *It follows from assumption (3) that there exists a positive real number  $L$  such that*

$$\mathcal{A}(y) \preceq \mathcal{A}(x) + D\mathcal{A}(x)(y - x) + L\|y - x\|^2 I \quad (38)$$

where  $x, y \in \Omega$ .

*Proof* See [1]. □

In [6], it was shown the existence of  $\tau_f > 0$  such that at any  $x \in \Omega_a$ , condition (22) is verified for any  $t \in [0, \tau_f]$ .

**Proposition 1** *For all  $x \in \Omega_a$  such that  $\|d(x)\| \geq M > 0$  there exist  $\tau > 0$  such that:*

$$\mathcal{A}(x + td(x)) \preceq 0 \quad (39)$$

for all  $t \in [0, \tau]$ .

*Proof* see [1]. □

It will be proved latter that, as a consequence of proposition 1, if  $x^* \in \Omega_a$  is an accumulation point of a sequence  $\{x^k\}_{k \in \mathbb{N}}$  generated by the algorithm, then  $\|d(x^*)\| = 0$ . Let us consider now the following result,

**Proposition 2** *Let be  $\{x^k\}_{k \in \mathbb{N}}$  a sequence given by the algorithm converging to  $x^*$  with  $\|d(x^*)\| = 0$ . There exist  $\delta > 0$  and  $\tau > 0$  such that, if  $x^k \in B(x^*, \delta) \cap \Omega_a$ , then*

$$\mathcal{A}(x^k + td(x^k)) \preceq 0 \quad (40)$$

for all  $t \in [0, \tau]$ .

*Proof* See [1]. □

Since  $\Omega_a$  is a compact, the sequence  $\{x^k\}_{k \in \mathbb{N}} \in \text{int}(\Omega_a)$  generated by the algorithm has an accumulation point  $x^* \in \Omega_a$ . The sequences  $A^k, B^k, \mathcal{A}(x^k)$  and  $\rho^k$  are bounded. Then, there exists  $\mathbb{K}_1 \subset \mathbb{N}$  such that  $\{d_0(x^k), d(x^k), \rho(x^k), A_0(x^k), \bar{A}(x^k), \mathcal{A}(x^k)\}_{k \in \mathbb{K}_1}$  converge to  $\{d_0(x^*), d(x^*), \rho(x^*), A_0(x^*), \bar{A}(x^*), \mathcal{A}(x^*)\}$ .

**Theorem 2** *If  $x^*$  is an accumulation point of the sequence  $\{x^k\}$  generated by the algorithm, then  $x^*$  is a stationary point of problem (2).*

*Proof* Let be the set  $\mathbb{K}_1 \subset \mathbb{K} \subset \mathbb{N}$  previously defined. It follows from propositions 1 and 2 that there exists  $\mathbb{K}_2 \subset \mathbb{K}_1$  such that  $t^k$  goes to  $t^* > 0$  for  $k \in \mathbb{K}_2$ .

We prove by contradiction that  $\|d(x^*)\| = 0$ . If we assume that  $\|d^*\| > \eta_d > 0$ , it follows from the line search condition (22),

$$f(x^{fol(k)}) \leq f(x^k) + \eta t^k (d^k)^\top \nabla f(x^k).$$

where  $fol(k)$  is the element that follows  $k$  in  $\mathbb{K}_2$ . Taking the limits for  $k \rightarrow \infty$ ,

$$f(x^*) \leq f(x^*) + \eta t^* (d^*)^\top \nabla f(x^*).$$

Then,

$$0 \leq (d^*)^\top \nabla f(x^*). \quad (41)$$

But, from lemma 8, when  $k \rightarrow \infty$ , we obtain

$$(d^*)^\top \nabla f(x^*) \leq \xi (d_0^*)^\top \nabla f(x^*)$$

and from lemma 7 we have, for  $k \rightarrow \infty$ ,

$$(d_0^*)^\top \nabla f(x^*) \leq -(d_0^*)^\top B^* d_0^*.$$

Since  $B^*$  is positive definite, we have  $(d^*)^\top \nabla f(x^*) < 0$ . This result is in contradiction with (41). Thus,  $d^* = 0$ .

Let  $\lambda_0^* = \text{vec}(A_0^*)$ , from (11) we have that

$$\nabla f(x^*) + \nabla \mathcal{A}(x^*) \lambda_0^* = 0 \quad (42)$$

$$(\mathcal{A}(x^*) \otimes I) \lambda_0^* = 0. \quad (43)$$

Then  $A_0^* \mathcal{A}(x^*) = 0$ . Therefore  $(x^*, \lambda_0^*)$  is a stationary point of our problem. □

## 5 Implementation details

### 5.1 Solving the linear systems

We discuss some issues about the numerical resolution of the internal linear systems in order to take advantage of the structure of  $\mathcal{A}$ . In the particular case of structural optimization we have than the stiffness and mass matrices have a particular structure that is very efficiently treated by finite element codes.

It follows from equations (11) that:

$$\lambda_0 = -(\mathcal{A}(x) \otimes I)^{-1} (I \otimes A) \nabla \mathcal{A}(x)^\top d_0 \quad (44)$$

Let  $U = (\mathcal{A}(x) \otimes I)^{-1} (I \otimes A) \nabla \mathcal{A}(x)^\top$ ,  $U \in \mathbb{R}^{q^2 \times n}$ . Then, we obtain the symmetric linear system :

$$[B - \nabla \mathcal{A}(x) U] d_0 = -\nabla f(x) \quad (45)$$



In consequence of Lemma 1:

$$U = (\mathcal{A}(x)^{-1} \otimes \Lambda) \nabla \mathcal{A}(x)^\top. \quad (46)$$

Let  $U_k$  be the  $k$ -th column of  $U$ . Thus, we have

$$\text{vec}(\Lambda \frac{\partial \mathcal{A}}{\partial x_k}(x) \mathcal{A}(x)^{-1}) = U_k. \quad (47)$$

Then  $U_k$  can be obtained by solving the following  $q$  linear systems,

$$\mathcal{A}(x) \text{mat}(U_k) = \frac{\partial \mathcal{A}}{\partial x_k}(x) \Lambda \quad (48)$$

The consequence is that the computation of  $U$  needs to solve  $q \times n$  linear systems of dimension  $q$ , all with the same symmetric negative definite matrix  $\mathcal{A}(x)$ .

The factorization to solve the symmetric system (45) requires  $\frac{1}{6}n^3$  floating point operations (flops).

In the case when  $\mathcal{A}(x)$  is a band matrix with bandwidth  $b$ , the Cholesky factorization needs about  $\frac{1}{2}(b(b+3)q)$  flops, see [11]. Note that in structural optimization applications in general we have  $b \ll q$ . Since the forward and backward substitution for each system requires about  $(2b+1)q$  flops, to compute  $U$  we need about  $q^2n(2b+1) + \frac{1}{2}(b(b+3)q)$  flops. The calculus of  $\lambda_0$  in (44) requires about  $2nq^2$  flops.

In another approach, we can solve the Schur complement problem:

$$\begin{aligned} (\nabla \mathcal{A}(x)^\top B^{-1} \nabla \mathcal{A}(x) - (\mathcal{A}(x) \otimes \Lambda^{-1})) \lambda_0 = \\ - \nabla \mathcal{A}(x)^\top B^{-1} \nabla f(x). \end{aligned} \quad (49)$$

In theorem (1) it was proven that the matrix of the system (49) is symmetric and positive definite. Then some preconditioned conjugate gradient method can be applied to solve this system. The resolution of this  $q^2$  linear system using a conjugate gradient method requires about  $2kq^4$  where  $k$  is the numbers of iterations.

Solving the dual system each iteration requires the product of the coefficient matrix by a vector. Employing limited memory quasi-Newton formulation, this product can be done without storing the coefficient matrix, [?].

## 5.2 Updating the dual matrix

Another crucial matter is the updating of the matrix  $\Lambda$ . To have global convergence of the presented algorithm  $\Lambda$  must satisfy assumptions 5 and 6. This is the case if  $\Lambda = \mu I$ ,  $\mu > 0$ .

$$(50)$$

However, to keep good local convergence properties of Newton like methods, we should take  $\Lambda^{k+1} = \Lambda_0^k$ . Unfortunately  $\Lambda_0^k$  is not symmetric and the assumptions 5 and 6 would not be always verified. We propose the following updating rule for  $\Lambda$ ,

- i) Compute  $\lambda_{0min}$ , the minimum eigenvalue of  $\text{sym}(\Lambda_0)$ .
- ii) If  $\lambda_{0min} \geq \lambda^I$  then, set  $\Lambda = \text{sym}(\Lambda_0)$ .
- iii) Else, set  $\Lambda = \text{sym}(\Lambda_0) + (\lambda^I - \lambda_{0min}) I$ .  $\square$

With this rule  $\Lambda$  meets assumption 5, but assumption 6 is true only at the limit. To ensure global convergence with this rule, we restart periodically with  $\Lambda = \mu I$ .  $\Lambda$  is also restarted if the search direction is not descent or feasible, [1].

In (49) the calculus of  $\Lambda^{-1}$  is required. A Choleski decomposition of  $\Lambda$  can be obtained in a procedure similar to Levenberg-Marquardt technique, [10].

## 5.3 Quasi-Newton formulation

The quasi-Newton matrix  $B$  must verify assumption 7. We employ the Broyden-Fletcher-Goldfarb-Shanno (BFGS) updating rule with Powell's correction to ensure positive definiteness of  $B$  [12], with  $B = I$  as initial value. For linear SDP problems, we have that  $\nabla_{xx}L(x, \lambda) \equiv 0$ . We take  $B = 10^{-6}I$ .

## 6 A general algorithm for nonlinear and SDP

In this section we extend the domain of application of the previous algorithm to problem (1), including inequality and equality vector constraints.

Inspired on techniques proposed in [6], we look for solutions of first order optimality conditions:

$$\begin{aligned} \nabla f(x) + \nabla g(x)\gamma + \nabla h(x)\mu + \nabla \mathcal{A}(x)\lambda &= 0 \\ \mathcal{G}(x)\gamma &= 0 \\ h(x) &= 0 \\ \Lambda \mathcal{A}(x) &= 0 \\ g(x) &\leq 0 \\ \mathcal{A}(x) &\preceq 0 \\ \gamma &\geq 0 \\ \Lambda &\preceq 0 \end{aligned} \quad (51)$$

where  $\lambda = \text{vec}(\Lambda)$ ,  $\Lambda \in \mathbb{S}_+^q$ ,  $\gamma \in \mathbb{R}_+^m$ . We denote  $\nabla g(x)$  and  $\nabla h(x)$  the transposed of the Jacobian matrices of  $g$  and  $h$  respectively and  $\mathcal{G}_{ij}(x) = \delta_{ij}g_i(x)$ ;  $i, j = 1, 2, \dots, m$ .

A Newton like iteration to solve the present problem is defined as follows:

$$\begin{bmatrix} B & \nabla g(x) & \nabla h(x) & \nabla \mathcal{A}(x) \\ \Gamma \nabla g(x)^\top & \mathcal{G}(x) & 0 & 0 \\ \nabla h(x)^\top & 0 & 0 & 0 \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top & 0 & 0 & \mathcal{A}(x) \otimes I \end{bmatrix} \begin{bmatrix} x_0 - x \\ \gamma_0 - \gamma \\ \mu_0 - \mu \\ \lambda_0 - \lambda \end{bmatrix} \quad (52)$$

$$= - \begin{bmatrix} \nabla f(x) + \nabla \mathcal{A}(x) \lambda \\ \Gamma \mathcal{G}(x) \\ h(x) \\ \text{vec}(\Lambda \mathcal{A}(x)) \end{bmatrix}$$

where  $(x, \gamma, \mu, \Lambda) \in \text{int}(\Omega_a) \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{S}_{++}^q$  is the current point and  $(x_0, \gamma_0, \mu_0, \lambda_0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^{q^2}$  are the new estimates.

In a similar way as in [6], we introduce the following potential function for the line search:

$$\varphi_c(x) = f(x) + \sum_{i=1}^p c_i |h_i(x)| \quad (53)$$

where  $c_i$  are positive constants.

Solving the following linear systems we obtain  $d_0$ , a descent direction of  $\varphi_c$ , as well as a restoring direction  $d_1$  and estimations of the dual variables,

$$\begin{bmatrix} B & \nabla g(x) & \nabla h(x) & \nabla \mathcal{A}(x) \\ \Gamma \nabla g(x)^\top & \mathcal{G}(x) & 0 & 0 \\ \nabla h(x)^\top & 0 & 0 & 0 \\ (I \otimes \Lambda) \nabla \mathcal{A}(x)^\top & 0 & 0 & \mathcal{A}(x) \otimes I \end{bmatrix} \begin{bmatrix} d_0 & d_1 \\ \gamma_0 & \gamma_1 \\ \mu_0 & \mu_1 \\ \lambda_0 & \lambda_1 \end{bmatrix} \quad (54)$$

$$= \begin{bmatrix} -\nabla f(x) & 0 \\ 0 & -\gamma \\ -h(x) & -\omega \\ 0 & -\lambda \end{bmatrix}$$

Let  $\Delta = \{x \in \Omega \text{ such that } h(x) \leq 0\}$  and  $\text{int}\{\Delta\}$  the interior of  $\Delta$  which is assumed to be not void. The proposed algorithm for Problem (2) is described now:

### The algorithm for general SDP

**Parameters.**  $\xi \in (0, 1)$ ,  $\eta \in (0, 1)$ ,  $\varphi > 0$  and  $\nu \in (0, 1)$ .

**Initial data.**  $x \in \text{int}(\Delta)$ ,  $\Lambda \in \mathbb{S}_{++}^q$  commuting with  $\mathcal{A}(x)$ ,  $\gamma > 0$ ,  $\omega = (1, \dots, 1) \in \mathbb{R}^p$ ,  $B \in \mathbb{S}_{++}^n$  and  $c \in \mathbb{R}_{++}^p$ .

**Step 1.** Computation of the search direction  $d$ .

(i) Solve the first linear system in (54).

If  $d_0 = 0$ , stop.

(ii) Solve the second linear system in (54).

(iii) If  $c_i < -1.2\mu_{0i}$ , then set  $c_i = -2\mu_{0i}$ ,  $i = 1, \dots, p$ .

(iv) Compute the parameter  $\rho$  such that

$$\rho = \min \left\{ \varphi \|d_0\|^2, (\xi - 1) \frac{d_0^\top \nabla \varphi_c(x)}{d_1^\top \nabla \varphi_c(x)} \right\} \quad (55)$$

if  $d_1^\top \nabla \varphi_c(x) > 0$ . Otherwise:

$$\rho = \varphi \|d_0\|^2. \quad (56)$$

(v) Compute the search direction  $d$  as

$$d = d_0 + \rho d_1. \quad (57)$$

**Step 2.** Line Search.

Find  $t$ , the first element of  $\{1, v, v^2, v^3, \dots\}$  such that

$$\varphi_c(x + td) \leq \varphi_c(x) + t\eta d^\top \nabla \varphi_c(x) \quad (58)$$

$$g_i(x + td) < 0, \quad i = 1, \dots, q \quad (59)$$

$$h(x + td) < 0 \text{ and}$$

$$\mathcal{A}(x + td) \prec 0. \quad (60)$$

**Step 3.** Updates.

(i) Take the new point  $x := x + td$ .

(ii) Define new value for  $B \in \mathbb{S}_{++}^n$ .

(iii) Define new value  $\gamma_i > 0$ ,  $i = 1, \dots, q$ .

(iv) Define new value for  $\Lambda \in \mathbb{S}_{++}^q$  commuting with  $\mathcal{A}(x)$ .

(v) Go to Step 1.

□

For  $c$  large enough, the potential function (53) is in fact an exact penalty function for the equality constraints, see [8]. But  $\varphi_c$  is not differentiable in points where the equality constraint are active. The present algorithm takes iterates at  $\text{int}(\Delta)$ , where smooth optimization techniques can be employed.

If  $x \in \Delta$  and  $c$  is such that  $c_i + \mu_{0i} > 0$ ;  $i = 1, \dots, p$  the search direction  $d$  is a descent direction for  $\varphi_c$  because:

$$d^\top \nabla \varphi_c(x) \leq \xi d_0^\top \nabla \varphi_c(x) \leq -d_0^\top B d_0. \quad (61)$$

Moreover,  $d$  is also a feasible direction for the inequality constraints and the constraints  $h_i(x) \leq 0$ ;  $i = 1, \dots, p$ . The line search of **Step 2** ensures a decrease of the potential function  $\varphi_c$ . The proof of these assertions are similar to the proof of lemmas 7 and 8.

Since we assume that  $\Omega_a$  is a compact, the global convergence of the presented algorithm for general and non linear SDP constraint can be proved using the same arguments that were used in section 4.

## 7 Applications in Structural Optimization and numerical tests

The present method is a very reliable tool for Structural Optimization. Several objective functions and constraints can be included in the optimization model. We mention weight, cost or compliance and also local constraints, like mechanical stresses or nodal displacements. Matrix constraints can be employed to include constraints on fundamental frequencies or stability.

Material optimization is a relevant application for our technique, since the elasticity matrices are required to be positive definite.

Thanks to the fact that all the iterates are feasible, the line search can be implemented so as a positive definite stiffness matrix is first obtained and the structural analysis is latter performed.

In this section, three trusses commonly studied in the literature [4] are selected to test the ability of the proposed algorithm on structural optimization problems with static and dynamics constraints. We also consider a L-shape 2D membrane.

The truss optimization problem of minimum weight subject to static and dynamics constraints can be formulated as follows:

$$\begin{aligned} \min_A W(A) &= \sum_{i=1}^{ne} \rho_i A_i L_i \\ \text{s.t. } A^- &\leq A_i \leq A^+ ; i = 1, \dots, ne \\ \sigma^- &\leq \sigma_i \leq \sigma^+ ; i = 1, \dots, ne \\ \sigma_{ib} &\leq \sigma_i ; i = 1, \dots, ne \\ u^- &\leq u_i \leq u^+ ; i = 1, \dots, ndg \\ K(A) - (\lambda^-)M(A) &\succcurlyeq 0 \end{aligned} \quad (62)$$

where  $W$  is the structural weight,  $ne$  is the number of bars and  $ndg$ , the number of reduced degrees of freedom.

To each element corresponds the material density  $\rho_i$ , the cross-sectional area  $A_i$ , the length  $L_i$  and the stress  $\sigma_i$ . The Euler buckling compressive stress limit is  $\sigma_{ib}$ . The nodal displacement along the  $i$ -th degree of freedom is denoted as  $u_i$ . Lower and upper constraints on a generic function  $\varphi_i$  are denoted as  $\varphi^-$  and  $\varphi^+$ , respectively. We call  $K$ , the reduced structural stiffness matrix and  $M$ , the reduced structural mass matrix. It can be proved that, if  $K(A) - (\lambda^-)M(A) \succcurlyeq 0$  is true, then  $\lambda^-$  is a lower bound on the generalized eigenvalues of  $K(A) - \lambda M(A)$ . In consequence, the last condition implies on a lower bound of the fundamental frequencies of the structure.

Eulers buckling compressive stress is taken as:

$$\sigma_{ib} = -\frac{C_E E A_i}{L_i^2} \quad (63)$$

where  $C_E$  represents Euler buckling coefficient.

The structural optimization problem of minimum volume for L-shape plate subject to static and dynamics constraints is formulated as follows:

$$\begin{aligned} \min_A V(h) &= \sum_{i=1}^{ne} h_i A_i \\ \text{s.t. } h_i &\geq 0 ; i = 1, \dots, ne \\ \sigma_{Mi} &\leq \sigma_M^+ ; i = 1, \dots, ne \\ u^- &\leq u_i \leq u^+ ; i = 1, \dots, ndg \\ K(h) - (\lambda^-)M(h) &\succcurlyeq 0 \end{aligned} \quad (64)$$

where  $V$  is the total volume of L-shape plate,  $h_i$  is the thickness of element  $i$  and  $\sigma_M^+$  is the upper bound on Von Mises stress  $\sigma_{Mi}$ .

If the inequality and matrix constraints are not verified by the initial points, a feasible initial point was found with the help of the auxiliary mathematical program

$$\begin{aligned} \min_{x,z} z \\ \text{s.t. } g(x) &\leq z \\ \mathcal{A}(x) - zI &\preceq 0 \end{aligned} \quad (65)$$

where  $z$  is a scalar auxiliary variable.

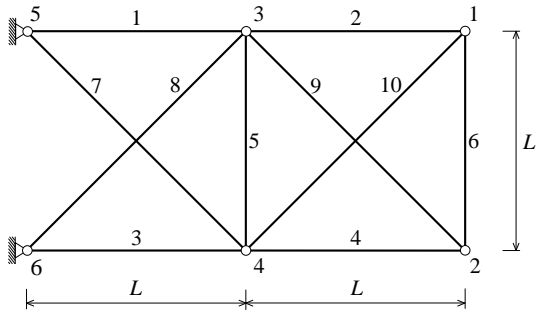
In all the examples the material density is 0.1 lb/in<sup>3</sup> (2768 Kg/m<sup>3</sup>) and the modulus of elasticity is 10000 Ksi (68.95 × 10<sup>3</sup> MPa).

All the problems were solved with the same set of parameters:  $\xi = 0.7$ ,  $\eta = 0.1$ ,  $\varphi = 1$ ,  $\nu = 0.7$ . The initial values were  $A = I$ ,  $\gamma_i = 1$ ; for  $i = 1, \dots, m$ ,  $B = I$  and  $c_i = 0$ ; for  $i = 1, \dots, p$ . Armijo's line search is employed in all the cases.

### 7.1 Ten-bar truss problem

Consider the 2D structure shown in Fig. 1 submitted to a single loading case given in Table 1. The cross-sectional area of each bar is considered as an independent variable. The initial cross-sectional of each bar is 30 in<sup>2</sup>. The stress limit is ± 25 Ksi (172.375 MPa) and displacements of nodes 1, 2, 3 and 4 are limited to ±2 in (±5.08 cm) in the  $y$  direction. Buckling constraints are not considered. The lower bounds on cross-sectional areas are 0.1 in<sup>2</sup> (0.6452 cm<sup>2</sup>) for all members. The value of  $\lambda^-$  is 23700 rad<sup>2</sup>/s<sup>2</sup> (24.5 Hz).

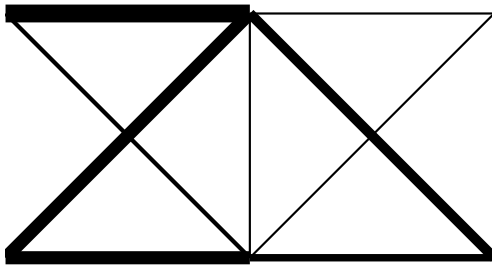
The number of iterations necessary to obtain an initial feasible point is nine.



**Fig. 1** Example 1. A ten-bar planar truss:  $L = 360$  in (914.4 cm)

**Table 1** Nodal load components Kips (KN) for example 1.

Load case	Node	$F_y$
1	2	-100 (-445)
	4	-100 (-445)



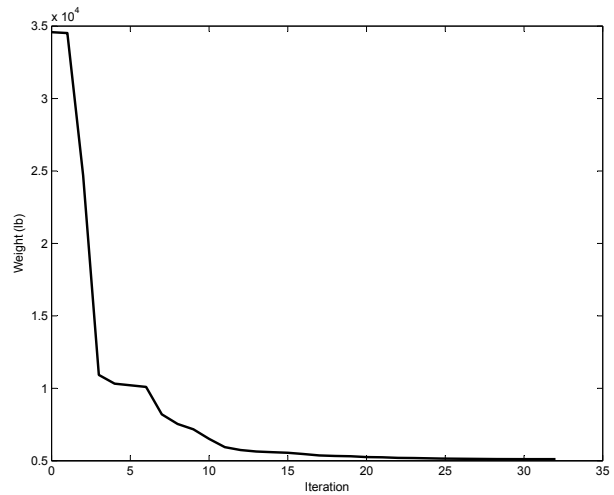
**Fig. 2** Example 1. Optimal design

**Table 2** Optimal results for example 1.

Design variables	Optimized solution
$A_1$	32.6484
$A_2$	0.1000
$A_3$	24.1002
$A_4$	13.6312
$A_5$	0.1000
$A_6$	0.2813
$A_7$	7.8205
$A_8$	23.7204
$A_9$	23.7204
$A_{10}$	23.7204
Weight (lbm)	5111.47
$\omega_1$ (Hz)	24.50

Figure 2 shows the optimum design of the structure and optimized results of the design variables are listed in Table 2.

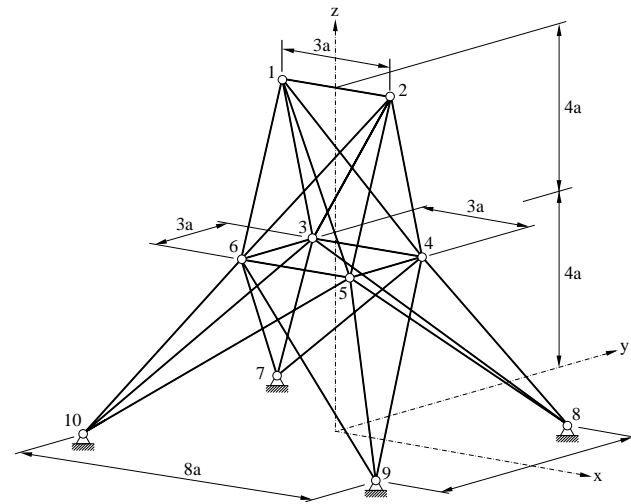
Figure 3 shows the iteration history of the structural weight of the ten-bar truss.



**Fig. 3** Convergence curve for the ten-bar truss

cm<sup>2</sup>). The stress limit is  $\pm 40$  Ksi (172.375 MPa) for each member and displacements of nodes 1 and 2 are limited to  $\pm 0.35$  in ( $\pm 0.889$  cm) in the  $x$ ,  $y$  and  $z$  directions. The lower bounds on cross-sectional areas are 0.01 in<sup>2</sup> (0.06452 cm<sup>2</sup>) for all members. The values of Euler buckling coefficient and  $\lambda^-$  are 39.274 [13] and 23700 rad<sup>2</sup>/s<sup>2</sup> (80 Hz), respectively.

The number of iterations necessary to obtain an initial feasible point is 11.



**Fig. 4** Example 2. A 25-bar space truss:  $a = 25$  in (63.5 cm)

7.2 25-bar truss problem

Let be the three-dimensional truss shown in Fig. 4 submitted to two load cases listed in Table 4. The members are divided into eight design groups according to Table 3. The initial cross-sectional of each bar is 3 in<sup>2</sup> (19.356

Fig. 5 shows the optimum design of the structure and optimized results of the design variables are listed in Table 5.

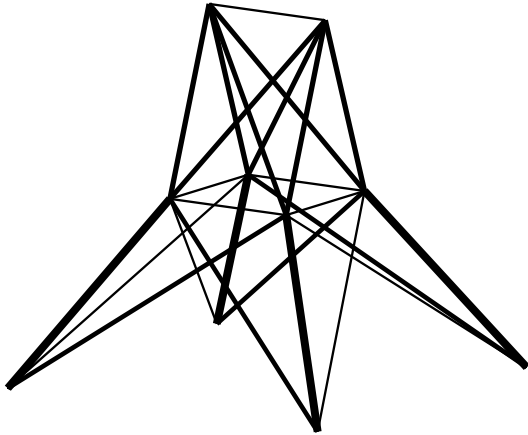
Figure 6 shows the iteration history of the structural weight of the ten-bar truss.

**Table 3** Design variable linking for example 2.

Design variables	Members	Connective of node
1	1	1-2
2	2-5	1-4, 2-3, 1-5, 2-6
3	6-9	2-4, 2-5, 1-3, 1-6
4	10,11	3-6, 4-5
5	12,13	3-4, 5-6
6	14-17	3-10, 6-7, 4-9, 5-8
7	18-21	4-7, 3-8, 5-10, 6-9
8	22-25	6-10, 3-7, 4-8, 5-9

**Table 4** Nodal load components Kips (KN) for example 2.

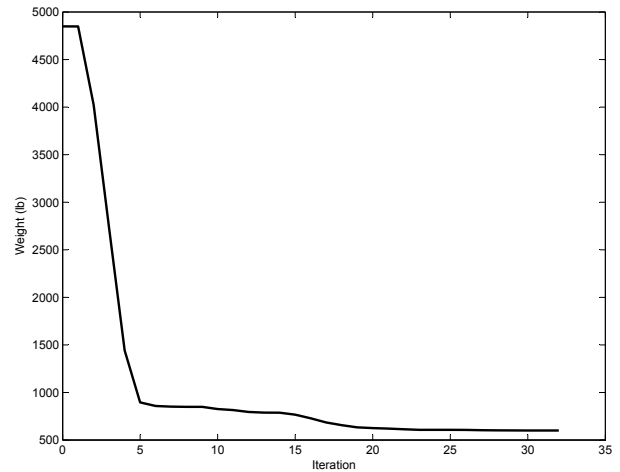
Load case	Node	$F_x$	$F_y$	$F_z$
1	1	1 (4.45)	10 (44.5)	-5 (-22.25)
	2	0	10 (44.5)	-5 (-22.25)
	3	0.5 (2.225)	0	0
	6	0.5 (2.225)	0	0
2	1	0	20 (89)	-5 (-22.25)
	2	0	-20 (-89)	-5 (-22.25)

**Fig. 5** Example 2. Optimal design**Table 5** Optimal results for example 2.

Design variables	Optimized solution
$A_1$	0.0100
$A_2$	2.1653
$A_3$	2.2947
$A_4$	0.5149
$A_5$	0.0322
$A_6$	0.6884
$A_7$	2.0322
$A_8$	3.4364
Weight (lbm)	599.927
$\omega_1$ (Hz)	80

### 7.3 72-bar truss problem

Let be the three-dimensional truss shown in Fig. 7 submitted to two load cases listed in Table 6. The members are divided into eight design groups according to Table 7. The initial cross-sectional of each bar is  $1 \text{ in}^2$  ( $6.452 \text{ cm}^2$ ). The stress limit is  $\pm 25 \text{ Ksi}$  ( $172.375 \text{ MPa}$ ) for each member and displacements of nodes 1, 2, 3 and

**Fig. 6** Convergence curve for the 25-bar truss

4 are limited to  $\pm 0.25$  in ( $\pm 0.635 \text{ cm}$ ) in the  $x$  and  $y$  directions. The lower bounds on cross-sectional areas are  $0.1 \text{ in}^2$  ( $0.6452 \text{ cm}^2$ ) for all members. The values of Euler buckling coefficient and  $\lambda^-$  are 39.274 and  $98700 \text{ rad}^2/\text{s}^2$  (50 Hz), respectively.

The number of iterations necessary to obtain an initial feasible point is 56.

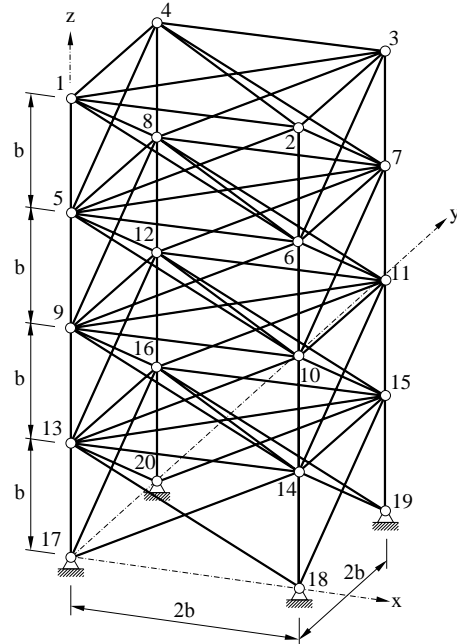
**Fig. 7** Example 2. A 72-bar space truss:  $b = 60 \text{ in}$  ( $152.4 \text{ cm}$ )

Fig. 8 shows the optimum design of the structure and optimized results of the design variables are listed in Table 8.

**Table 6** Nodal load components Kips (KN) for example 3.

Load case	Node	$F_x$	$F_y$	$F_z$
1	1	5 (22.25)	5 (22.25)	-5 (-22.25)
2	1	0	0	-5 (-22.25)
	2	0	0	-5 (-22.25)
	3	0	0	-5 (-22.25)
	4	0	0	-5 (-22.25)

**Table 7** Design variable linking for example 3.

Design variable	Members	Connective of node
1	1-4	1-5, 2-6, 3-7, 4-8
2	5-12	2-5, 1-6, 3-6, 2-7, 4-7, 3-8, 1-8, 4-5
3	13-16	1-2, 2-3, 3-4, 4-1
4	17,18	1-3, 2-4
5	19-22	5-9, 6-10, 7-11, 8-12
6	23-30	6-9, 5-10, 7-10, 6-11, 8-1, 7-12, 5-12, 8-9
7	31-34	5-6, 6-7, 7-8, 8-5
8	35,36	5-7, 6-8
9	37-40	9-13, 10-14, 11-15, 12-16
10	41-48	10-13, 9-14, 11-14, 10-15, 12-15, 11-16, 9-16, 12-13
11	49-52	9-10, 10-11, 11-12, 12-9
12	53,54	9-11, 10-12
13	55-58	13-17, 14-18, 15-19, 16-20
14	59-66	14-17, 13-18, 15-18, 14-19, 16-19, 15-20, 13-20, 16-17
15	67-70	13-14, 14-15, 15-16, 16-13
16	71,72	13-15, 14-16

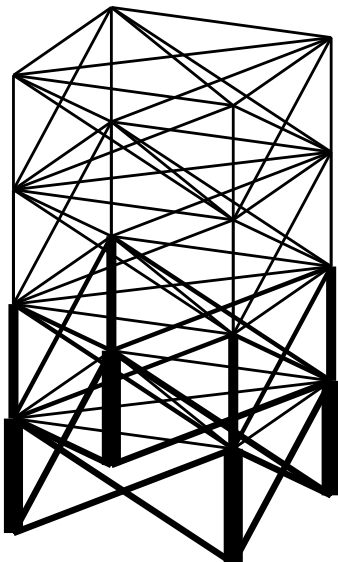
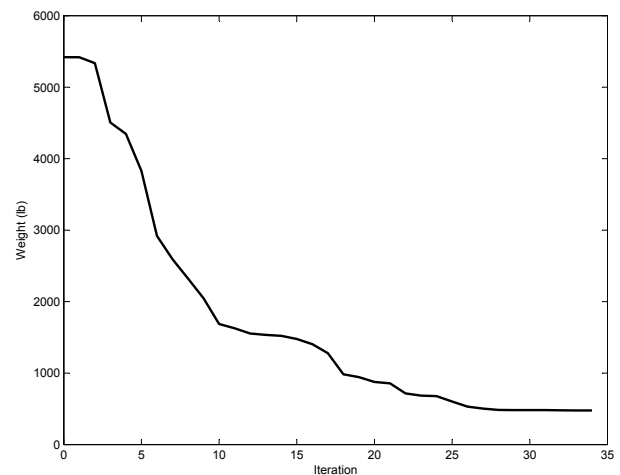
**Fig. 8** Example 3. Optimal design

Figure 9 shows the iteration history of the structural weight of the ten-bar truss.

**Table 8** Optimal results for example 3.

Design variables	Optimized solution
$A_1$	0.1990
$A_2$	0.3482
$A_3$	0.2156
$A_4$	0.3765
$A_5$	0.5526
$A_6$	0.3956
$A_7$	0.1000
$A_8$	0.1000
$A_9$	1.7779
$A_{10}$	0.7935
$A_{11}$	0.1000
$A_{12}$	0.1000
$A_{13}$	3.3604
$A_{14}$	1.1447
$A_{15}$	0.1000
$A_{16}$	0.1000
Weight (lbm)	476.93
$\omega_1$ (Hz)	50

**Fig. 9** Convergence curve for the 72-bar truss

#### 7.4 L-shape plate problem

Consider the 2D shape that is shown in Fig. 10 submitted to a total vertical of 110 Kips (489.3 KN) applied on the right side in form of punctual forces acting over the distinct nodes existing on that face and fixed in the bottom side (see Table 9). The thickness of each element is considered as an independent variable.

**Table 9** Nodal load components Kips (KN) for example 4.

Load case	Node	$F_y$
1	1	-10 (44.5)
	$\vdots$	$\vdots$
	11	-10 (44.5)

The shape is divided in 300 elements with 4 nodes quadrilateral element. The initial thickness of each el-

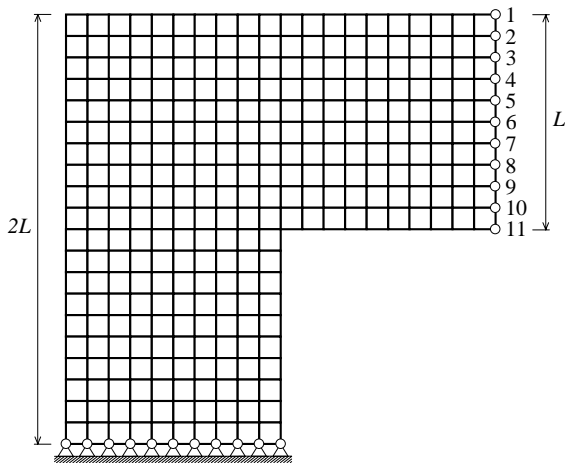


Fig. 10 Example 4. L-shape plate

element is 1 in (25.4 mm) and the initial area for each element is  $4 \text{ in}^2$  ( $101.6 \text{ mm}^2$ ). The Von Mises stress limit is 50 Ksi (345 MPa) for each element and displacement of node 11 is limited to  $\pm 1.5$  in ( $\pm 38.1$  mm) in the  $y$  direction. The value of  $\lambda^-$  is  $4 \times 10^6 \text{ rad}^2/\text{s}^2$  (318 Hz).

The number of iterations necessary to obtain an initial feasible point is 18.

Fig. 11 shows the optimum design of the L-shape plate.

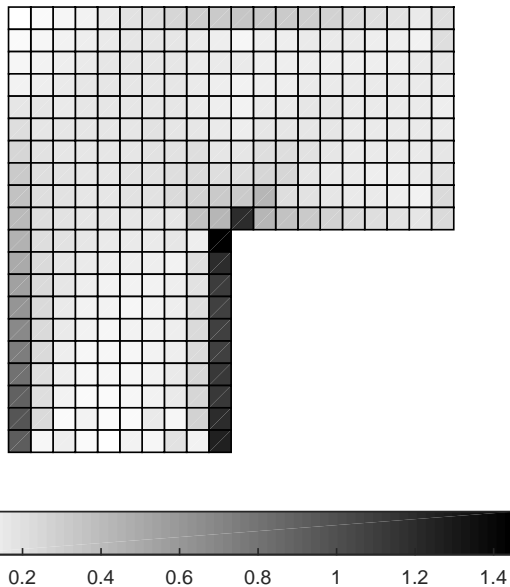


Fig. 11 Example 4. Optimal design of L-shape plate

Fig. 12 shows the iteration history volume of the L-shape plate.

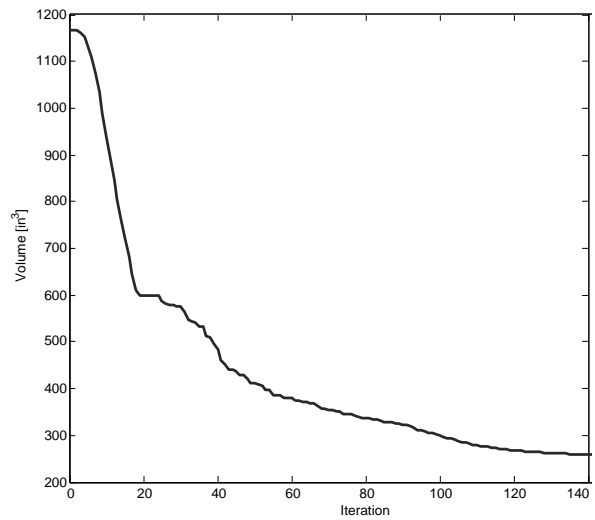


Fig. 12 Convergence curve for the L-shape plate

## 8 Conclusions

In this paper, a new approach for general nonlinear and semidefinite programming is presented for weight minimization of truss structures in terms of sizing variables with static and dynamics constraints.

The test problems were solved very efficiently with the same set of parameters.

The number of iterations in the line search was very small. The line search required one, two or three evaluations of the objective function and the constraints.

## 9 Acknowledgments

The authors thank the Brazilian Research Councils CAPES, CNPq, FAPERJ, the institutions supporting the program Ciência Sem Fronteiras of Brazil, the French Research Councils CNRS and INRIA and the Brazilian-French Network in Mathematics for the financial support.

## References

1. Miguel Aroztegui, Jos Herskovits, JeanRodolphe Roche, and Elmer Bazán. A feasible direction interior point algorithm for nonlinear semidefinite programming. *Structural and Multidisciplinary Optimization*, pages 1–17, 2014.
2. F.J. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Verlag, 2000.
3. Etienne de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*. Kluwer, 2002.
4. R. Haftka and Z. Gurdal. *Elements of Structural Optimization*. Kluwer Academic Publishers, 3 edition, 1992.
5. R. E. Hartwig. A note on the partial ordering of positive semi-definite matrices. *Linear and Multilinear Algebra*, 6:223–226, 1978.

- 
6. J. Herskovits. A feasible directions interior point technique for nonlinear optimization. *Journal of Optimization Theory and Applications*, 99:121–146, 1998.
  7. R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
  8. David G. Luenberger. *Linear and Nonlinear Programming*. Addison Wesley, 1984.
  9. David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, 2008.
  10. J.J. More. The levenberg-marquardt algorithm: implementation and theory. *Lecture Notes in Mathematics*, pages 105–116, 1978.
  11. A. Pissanetzky. *Sparse matrix technology*. 1984.
  12. M. J. D. Powell. *The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations, in Nonlinear Programming 3*. Academic Press, London, 1978.
  13. S. Rao. *Engineering Optimization. Theory and Practice*. John Wiley & Sons, Inc., 4 edition, 2009.
  14. Alexander Shapiro. First and second order analysis of nonlinear semidefinite programs. *Mathematical Programming*, 77:301–320, 1994.
  15. Alexander Shapiro. On uniqueness of lagrange multipliers in optimization problems subject to cone constraints. *SIAM Journal on Optimization*, 7:508–518, 1997.