



**HAL**  
open science

# Stability of Lagrangian Duality for Nonconvex Quadratic Programming. Solution Methods and Applications to Computer Vision

Pham-Dinh Tao, Thai-Quynh Phong, Radu Horaud, Long Quan

► **To cite this version:**

Pham-Dinh Tao, Thai-Quynh Phong, Radu Horaud, Long Quan. Stability of Lagrangian Duality for Nonconvex Quadratic Programming. Solution Methods and Applications to Computer Vision. ESAIM: Mathematical Modelling and Numerical Analysis, 1997, 31 (1), pp.57–90. inria-00590073

**HAL Id: inria-00590073**

**<https://inria.hal.science/inria-00590073>**

Submitted on 16 Jan 2014

**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.



**STABILITY OF LAGRANGIAN DUALITY FOR NONCONVEX QUADRATIC PROGRAMMING. SOLUTION METHODS AND APPLICATIONS IN COMPUTER VISION (\*) (\*\*)**

by PHAM DINH TAO <sup>(1)</sup>, THAI QUYNH PHONG <sup>(1)</sup>, Radu HORAUD <sup>(2)</sup> and LONG QUAN <sup>(2)</sup>

*Abstract.* — *The problem of minimizing a quadratic form over a ball centered at the origin is considered. The stability of Lagrangian duality is established and complete characterizations of a global optimal solution are given. On the basis of this theoretical study, two principal solution methods are presented. An important application of nonconvex quadratic programming is the computation of the step to a new iterate in the Trust Region (TR) approach methods which are known to be efficient for nonlinear optimization problems. Also, we discuss the mathematical models of some important problems encountered in Computer Vision. Most of them can be formulated as a minimization of a sum of squares of nonlinear functions. A practical TR-based algorithm is proposed for nonlinear least squares problem which seems to be well suited for our applications.*

*Key words :* Nonlinear optimization, quadratic programming, stability, duality, safeguarding, dichotomy, trust region, nonlinear least squares, computer vision.

*Résumé.* — *L'étude de la stabilité de la dualité Lagrangienne relative au problème de minimisation d'une forme quadratique non convexe sur une boule euclidienne est présentée. Elle permet d'établir les caractérisations complètes des solutions optimales globales du problème. Pour la résolution duquel nous proposons deux algorithmes globaux de type primal-dual basés sur ces résultats théoriques. Une des applications importantes de ces algorithmes concerne le calcul d'un pas de déplacement dans les méthodes de région de confiance qui sont reconnues très robustes et performantes pour les problèmes d'optimisation non linéaire. Nous discutons aussi des modélisations mathématiques des problèmes importants rencontrés en Vision par Ordinateur. La plupart peuvent être formulés comme un problème de moindres carrés non linéaires. Finalement une méthode pratique de région de confiance est proposée pour ces problèmes qui semble très bien adaptée à nos applications.*

**1. INTRODUCTION**

In this paper, we are concerned with the global minimization problem

$$(QP) \quad \min \left\{ q(x) := \frac{1}{2} x^T A x + b^T x : \|x\| \leq r \right\}$$

(\*) Manuscript received June 2, 1996.

(\*\*) This work has been partially supported by the Esprit programme through the Basic Research Project « SECOND » (Esprit BRA No. 6769).

<sup>(1)</sup> LMI-INSA Rouen, CNRS URA 1378, 76131 Mt St Aignan Cedex, France

<sup>(2)</sup> LIFIA-CNRS-INRIA, 46, avenue Félix Viallet, 38031 Grenoble France.

where  $A$  is an  $(n \times n)$ -symmetric matrix and  $b, x \in \mathbb{R}^n$ . This problem is very important both from a theoretical and a practical viewpoint. In particular, problem of the form  $(QP)$  plays a fundamental role in Trust Region (TR) methods (cf. Gay [8], Moré [15], [16], Moré & Sorensen [17], Pham Dinh Tao *et al.* [1], [26]). Despite its nonconvex structure, a necessary and sufficient condition for an optimal solution of problem  $(QP)$  has been established. The most successful algorithm for solving  $(QP)$  which is based on the optimality condition was firstly developed by Moré & Sorensen [17] in connection with TR methods.

The purpose of this paper is to provide a complete discussion on theoretical and computational aspects of problem  $(QP)$ . Especially we will establish, like a case of convex optimization, the stability in Lagrangian duality, namely there is no gap between the optimal value of the primal and dual problems. Also we will derive a complete characterization of the optimal solutions. Conceptually, problem  $(QP)$  can be easily solved provided the eigensystem of  $A$  is available. But in practice the computation of the latter, especially for large scale problems, is usually time consuming. To avoid this, several inexpensive approximate methods were proposed (e.g. the safeguarding algorithm by Moré & Sorensen and the dichotomy algorithm by Pham Dinh Tao for solving  $(QP)$ ). The idea underlying these methods is to locate a dual solution by a bisection-type technique. In fact, these methods are of primal-dual type in the sense that they provide both primal and dual solutions.

As it may be seen, the solution of problem  $(QP)$  is the most costly part in the TR model approach for unconstrained optimization. TR methods differ by the way we obtain a TR step which is an approximate solution of  $(QP)$ . When the objective function is a sum of squares of nonlinear functions we have a nonlinear least squares problem. For this special case we propose an efficient practical algorithm which is based on the TR approach. Nonlinear least squares problems have many applications in practice. Particularly, most important problems which are encountered in Computer Vision (e.g., recovering the parameters of a camera, the motion and reconstruction of an object) usually lead to solving the system of nonlinear equations

$$\phi_i(x) = 0, i = 1, \dots, m \quad x \in \mathbb{R}^n$$

which in turn can be formulated as the unconstrained minimization problem

$$\min \left\{ f(x) = \sum_{i=1}^m \phi_i^2(x) : x \in \mathbb{R}^n \right\}.$$

It should be noted that for such problems the Newton method for solving a nonlinear system can not be applied because in general  $m \gg n$  and practically there is no exact solution due to perturbed data. Our preliminary experiments show that the practical TR-based algorithm is well suited for our applications.

The paper is organized as follows. Section 2 deals with the stability of Lagrangian duality for problem  $(QP)$ . Global optimality criteria for this problem and a complete characterization of its solution set are given in Section 3. Section 4 deals with numerical methods for  $(QP)$ . The TR model approach for unconstrained minimization is presented in Section 5 where a practical TR-based algorithm for nonlinear least squares is detailed. As applications, we discuss the mathematical models of some important problems in Computer Vision in Section 6. Finally, some preliminary numerical results are reported in Section 7.

## 2. STABILITY OF LAGRANGIAN DUALITY FOR $(QP)$

Consider the problem  $(QP)$  of minimizing a quadratic form inside a sphere. We will show that, like a case of convex programming, there is no gap between problem  $(QP)$  and its dual. Towards this end, we rewrite  $(QP)$  as

$$(P) \quad \alpha = \min \left\{ q(x) : \frac{1}{2} \|x\|^2 \leq \frac{r^2}{2} \right\}$$

so the Lagrangian associated to  $(QP)$  is

$$L(x, \lambda) = \begin{cases} q(x) + \frac{\lambda}{2} (\|x\|^2 - r^2) & \text{if } \lambda \geq 0, \\ -\infty & \text{otherwise.} \end{cases} \quad (1)$$

The dual problem then reads

$$(D) \quad \beta = \sup \{ g(\lambda) : \lambda \geq 0 \},$$

where  $g(\lambda)$  is defined via the following optimization problem

$$(P_\lambda) \quad g(\lambda) = \inf \{ L(x, \lambda) : x \in \mathbb{R}^n \}, \quad \lambda \geq 0.$$

It is easily seen that  $g(\lambda)$  is a concave function so  $(D)$  is a convex program.

It should be noted that the analytical expression of the feasible region under the form  $\{x : \|x\|^2/2 \leq r^2/2\}$  is essential, because otherwise, the Lagrangian should be defined as

$$L(x, \lambda) = \begin{cases} q(x) + \frac{\lambda}{2} (\|x\| - r) & \text{if } \lambda \geq 0, \\ -\infty & \text{otherwise.} \end{cases}$$

For any  $x \in \mathcal{N}(A - \lambda_1 I)$  one has

$$L(x, \lambda) = \frac{\lambda_1}{2} \|x\|^2 + b^T x + \frac{\lambda_1}{2} (\|x\| - r)$$

so if  $\lambda_1 < 0$  then for all  $\lambda \geq 0$

$$L(x, \lambda) \rightarrow -\infty \quad \text{as} \quad \|x\| \rightarrow +\infty.$$

Therefore  $\beta = -\infty$  and the duality gap would equal  $+\infty$ .

In what follows, we denote by  $\mathcal{P}$ ,  $\mathcal{P}_\lambda$ ,  $\mathcal{D}$  the solution sets of  $(P)$ ,  $(P_\lambda)$  and  $(D)$  respectively. We first give some characteristics of the function  $g(\lambda)$ .

PROPOSITION 1: (i)  $\text{dom } g = \{\lambda \in \mathbb{R} : \lambda \geq \max\{0, -\lambda_1\}\}$  and  $\lambda = -\lambda_1$  if  $b \in \mathcal{N}(A - \lambda_1 I)^\perp$ . (ii)  $g(\lambda) + \lambda r^2/2 = 1/2 b^T x - 1/2 b^T (A + \lambda I)^+ b$  for all  $x \in \mathcal{P}_\lambda$ ,  $\lambda \in \text{dom } g$  where  $A^+$  denotes the pseudo-inverse of  $A$ .

*Proof:* Recall that for the concave function  $g$  (Rockafellar [19])  $\text{dom } g = \{\lambda \geq 0 : g(\lambda) > -\infty\}$ .

(i) If  $\lambda < -\lambda_1$  then for  $x \in \mathcal{N}(A - \lambda_1 I)$  we have

$$x^T (A + \lambda I) x + b^T x = (\lambda + \lambda_1) \|x\|^2 + b^T x \rightarrow -\infty$$

whenever  $\|x\| \rightarrow \infty$ . Therefore  $g(\lambda) = -\infty$  which means  $\lambda \notin \text{dom } g$ .

Now let  $\lambda \geq \max\{0, -\lambda_1\}$ . Then

$$L(x, \lambda) = \frac{1}{2} x^T (A + \lambda I) x + b^T x - \frac{\lambda}{2} r^2$$

and  $(P_\lambda)$  is a concave quadratic program. Hence

$$\lambda \in \text{dom } g \Leftrightarrow \mathcal{P}_\lambda \neq \emptyset \Leftrightarrow \exists \bar{x} : (A + \lambda I) \bar{x} + b = 0,$$

i.e.  $(A + \lambda I) x = -b$  is solvable. Thus if  $\lambda = -\lambda_1$  then we must have  $b \in \mathcal{N}(A - \lambda_1 I)^\perp$ .

(ii) Consider  $\lambda \in \text{dom } g$ . The solution set  $\mathcal{P}_\lambda$  can be expressed as

$$\mathcal{P}_\lambda = -(A + \lambda I)^+ b + \mathcal{N}(A + \lambda I). \quad (2)$$

Notice that if  $\lambda > -\lambda_1$  then  $(A + \lambda I) = \{0\}$  and clearly  $b \in \mathcal{N}(A + \lambda I)^\perp$ . For any  $x \in \mathcal{P}_\lambda$  one has

$$(A + \lambda I) x = -b, \quad (3)$$

hence

$$g(\lambda) + \frac{\lambda}{2} r^2 = -\frac{1}{2} x^T (A + \lambda I) x + b^T x = -\frac{1}{2} b^T x + b^T x = \frac{1}{2} b^T x.$$

Also, it follows from (2) that

$$\frac{1}{2} b^T x = -\frac{1}{2} b^T (A + \lambda I)^+ b.$$

Denote by  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  the eigenvalues of  $A$  and by  $u_1, u_2, \dots, u_n$  their corresponding eigenvectors which constitute an orthogonal basis of  $\mathbb{R}^n$ . The solution set  $\mathcal{P}_\lambda$  and hence function  $g(\lambda)$  can be expressed via the eigensystem as follows:

LEMMA 1 : (i) If  $\lambda > -\lambda_1$  then  $(P_\lambda)$  has an unique solution  $x(\lambda)$  defined by the system

$$x^T u_i = -\frac{b^T u_i}{\lambda + \lambda_i}, \quad i = 1, \dots, n$$

and

$$\|x(\lambda)\|^2 = \sum_{i=1}^n \frac{(b^T u_i)^2}{(\lambda_i + \lambda)^2}. \quad (4)$$

(ii) If  $\lambda = -\lambda_1$  then problem (3) has a solution if and only if  $b \in \mathcal{N}(A - \lambda_1 I)^\perp$ . In this case  $\mathcal{P}_\lambda = x^+ + \mathcal{N}(A - \lambda_1 I)$  where

$$x^+ = -(A - \lambda_1 I)^+ b \quad (5)$$

is defined by

- if  $b = 0$  then  $x^+ = 0$
- if  $b \neq 0$  then the complement in  $\{1, \dots, n\}$  of  $J_1 = \{i : \lambda_i = \lambda_1\}$  is nonempty and

$$x^{+T} u_i = -\frac{b^T u_i}{\lambda + \lambda_i}, \quad i \notin J_1$$

so

$$\|x^+\|^2 = \sum_{i \notin J_1} \frac{(b^T u_i)^2}{(\lambda_i + \lambda)^2}. \quad (6)$$

Set  $\tilde{g}(\lambda) = -g(\lambda)$ . Clearly  $\tilde{g}$  is convex and  $\text{dom } \tilde{g} = \text{dom } g$ . The following results are direct consequences of the above Lemma and well-known calculus rules for subdifferential in convex analysis (Rockafellar [19], P. J. Laurent [13]).

COROLLARY 1: (i) For all  $\lambda \in \text{dom } \tilde{g}$  and  $\lambda > -\lambda_1$

$$\tilde{g}(\lambda) = \frac{1}{2} \sum_{i=1}^n \frac{(b^T u_i)^2}{\lambda_i + \lambda} + \frac{\lambda}{2} r^2.$$

In particular if  $\lambda_1 > 0$  then  $\text{dom } \tilde{g} = [0, +\infty)$  and the equation holds for all  $\lambda \in \text{dom } \tilde{g}$

(ii) if  $\lambda_1 \leq 0$  then  $-\lambda_1 \in \text{dom } \tilde{g}$  if  $b \in \mathcal{N}(A - \lambda_1 I)^\perp$  and

- If  $b = 0$  then  $\tilde{g}(\lambda) = \lambda r^2 / 2$
- If  $b \neq 0$  then  $(\{1, \dots, n\} \setminus J_1) \neq \emptyset$  and

$$\tilde{g}(-\lambda_1) = \frac{1}{2} \sum_{i \notin J_1} \frac{(b^T u_i)^2}{\lambda_i + \lambda} + \frac{\lambda}{2} r^2,$$

where  $J_1 = \{i : \lambda_i = \lambda_1\}$ .

COROLLARY 2: (i) If  $\lambda \in \text{dom } \tilde{g}$  and  $\lambda > -\lambda_1$  then  $\tilde{g}$  is differentiable at  $\lambda > 0$  and

$$\tilde{g}'(\lambda) = \frac{1}{2} (r^2 - \|x(\lambda)\|^2).$$

In particular, if  $\lambda_1 > 0$  then  $\tilde{g}$  is subdifferentiable at 0 and

$$\partial \tilde{g}(0) = \left[ -\infty, \frac{1}{2} (r^2 - \|x(0)\|^2) \right].$$

(ii) If  $\lambda_1 \leq 0$  and  $b \in \mathcal{N}(A - \lambda_1 I)^\perp$  then  $\tilde{g}$  is subdifferentiable at  $-\lambda_1$  and

$$\partial \tilde{g}(-\lambda_1) = \left( -\infty, \frac{1}{2} (r^2 - \|x^+\|^2) \right).$$

(Recall that  $x(\lambda)$  is the unique solution of  $(P_\lambda)$ ).

Let us consider

$$\beta = \sup \{g(\lambda) : \lambda \geq 0\} = -\inf \{\tilde{g}(\lambda) : \lambda \geq 0\}.$$

Since  $\tilde{g}$  is convex,  $\lambda^*$  is a solution to  $(D)$  if and only if  $0 \in \partial \tilde{g}(\lambda^*)$ . The following characterization of a dual optimal solution is immediate from Corollary 2.

PROPOSITION 2: (i)  $\mathcal{D}$  is singleton

(ii) Let  $\lambda^* \geq 0, \lambda^* > -\lambda_1$ . Then

$$\lambda^* \in \mathcal{D} \Leftrightarrow \|x(\lambda^*)\|^2 = r^2.$$

In particular if  $\lambda_1 > 0$  then

$$0 \in \mathcal{D} \Leftrightarrow \|x(0)\|^2 \leq r^2.$$

(iii) Let  $\lambda^* = -\lambda_1 \geq 0$  then

$$\lambda^* \in \mathcal{D} \Leftrightarrow b \in \mathcal{N}(A - \lambda_1 I)^\perp \quad \text{and} \quad \|x^+\|^2 \leq r^2.$$

We are now in position to establish the main theorem of this section.

THEOREM 1: (i)  $\alpha = \beta$ .

(ii)  $\lambda^* \in \mathcal{D}$  if and only if there exists  $x^* \in \mathcal{P}_{\lambda^*}$  such that

$$\|x^*\| \leq r \quad \text{and} \quad \lambda^*(\|x^*\|^2 - r^2) = 0. \quad (7)$$

*Proof:* Let  $\lambda \in \mathcal{D}$ . From the above proposition we have either  $\|x(\lambda^*)\| = r$  (in particular,  $(\|x(0)\| \leq r)$  if  $\lambda^* > -\lambda_1$  or  $\|x^+\| \leq r$  if  $\lambda^* = -\lambda_1$ ). We shall show that there is a solution  $x^*$  of  $(P_{\lambda^*})$  such that

$$\lambda^*(\|x^*\|^2 - r^2) = 0.$$

Obviously, if  $\lambda^* > -\lambda_1$  then we can take  $x^* = x(\lambda^*)$ . If  $\lambda^* = -\lambda_1$  then  $\mathcal{P}_{\lambda^*} = x^+ + \mathcal{N}(A - \lambda_1 I)$  so we can choose  $x^* \in \mathcal{P}_{\lambda^*}$  such that  $\|x^*\| = r$ . Thus, we have

$$\alpha \leq f(x^*) = f(x^*) + \frac{\lambda^*}{2} (\|x^*\|^2 - r^2) = \beta$$

which implies  $\beta = \alpha$ .

Also, we have shown if  $\lambda^* \in \mathcal{D}$  then  $x^*$ , constructed as above, satisfies (7). Conversely, if for  $\lambda^* \in \text{dom } g$  there exists a  $x^* \in \mathcal{P}_{\lambda^*}$  satisfying (7) then  $g(\lambda^*) = \beta$ , i.e.  $\lambda^* \in \mathcal{D}$ . The proof is complete.

The following optimality condition, whose proof is not trivial is straightforward from Theorem 1.

THEOREM 2 (Sorensen [22], Pham D. Tao *et al.* [26]):  $x^*$  is a solution to (QP) if and only if there exists  $\mu_* \geq 0$  such that:

- (i)  $A + \mu_* I$  is positive semi-definite,
- (ii)  $(A + \mu_* I)x^* = -b$ ,
- (iii)  $\|x^*\| \leq r$  and  $\mu_*(\|x^*\| - r) = 0$ .



Such a  $\mu_*$  is the unique dual solution ( $\mathcal{D} = \{\mu_*\}$ ).

Once  $\lambda^* \in \mathcal{D}$  is known a solution to (QP) can be derived from the equation  $(A + \lambda^*)x = -b$ . However,  $g(\lambda)$  is defined via  $(P_\lambda)$  which is itself a very difficult nonconvex optimization problem. The methods we will present in the sequel are primal-dual in the sense that they provide both primal and dual solutions to problem (QP).

### 3. CHARACTERIZATION OF THE SOLUTION SET OF (QP)

Denote by  $x(\mu)$  a solution of the equation

$$(A + \mu I)x = -b. \quad (8)$$

The following characteristics of the solution set of (QP) which are immediate from Theorem 2. They are helpful for understanding the algorithms described in the next sections.

PROPOSITION 3: 1. If  $\lambda_1 > 0$  (i.e.  $A$  is positive definite) then

(i) if  $\|A^{-1}b\| \leq r$  then  $x^* = -A^{-1}b$  is the only solution to (QP) ( $\mu_* = 0$  in Theorem 2).

(ii) otherwise  $\|x(0)\| > r$  and there exists a unique solution  $\mu_* > 0$  such that  $x(\mu_*)$  is the only solution to (QP).

2. If  $\lambda_1 = 0$  (i.e.  $A$  is only positive semi-definite) one has

(i) if  $\|x^+\| \leq r$  and  $Ax^+ = -b$  (this equality occurs if and only if  $(b \in \mathcal{N}(A)^\perp)$ ) then the set

$$x^* = x^+ + u, \quad u \in \mathcal{N}(A)$$

such that

$$\|x^*\|^2 = \|x^+\|^2 + \|u\|^2 \leq r^2$$

is the solution set to (QP) ( $\mu_* = 0$  in Theorem 2).

(ii) otherwise there exists a unique  $\mu_* > 0$  such that  $x(\mu_*)$  is the only solution to (QP).

3. If  $\lambda_1 < 0$  (i.e.  $A$  is not positive semi-definite) then

(i) if  $\|x^+\| \leq r$  and  $(A - \lambda_1 I)x^+ = -b$  (this equality occurs if and only if  $(b \in \mathcal{N}(A - \lambda_1 I)^\perp)$ ) then the set

$$x^* = x^+ + u, \quad u \in \mathcal{N}(A - \lambda_1 I)$$

such that

$$\|x^*\|^2 = \|x^+\|^2 + \|u\|^2 = r^2$$

is the solution set to (QP) ( $\mu_* = -\lambda_1$  in Theorem 2).

(ii) otherwise there exists a unique solution  $\mu_* > -\lambda_1$  such that  $x(\mu_*)$  is the only solution to (QP).

In particular when  $b = 0$  one has

COROLLARY 3: Suppose  $b = 0$  then the solution set of (QP) is

$$\{0\} \quad \text{if } \lambda_1 > 0,$$

$$\{x \in \mathcal{N}(A) : \|x\| \leq r\} \quad \text{if } \lambda_1 > 0,$$

$$\{x \in \mathcal{N}(A - \lambda_1 I) : \|x\| \leq r\} \quad \text{if } \lambda_1 > 0.$$

COROLLARY 4: Problem (QP) admits only solutions in the interior of the ball  $\{x \in \mathbb{R}^n : \|x\| \leq r\}$  if and only if  $A$  is positive definite and  $\|A^{-1}b\| < r$ .

#### 4. NUMERICAL ALGORITHMS FOR SOLVING (QP)

It can be seen from Proposition 3 that solving (QR) usually leads to finding  $\mu_* > \max\{0, -\lambda_1\}$  such that  $\|x(\mu_*)\| = r$ . Let

$$\phi(\mu) = \|x(\mu)\| \tag{9}$$

then we have to solve the nonlinear equation  $\phi(\mu) = r$ . In view of (4) we have, for all  $\mu > -\lambda_1$ ,

$$\phi(\mu) = \left( \sum_{i=1}^n \frac{(u_i^T b)^2}{(\lambda_i + \mu)^2} \right)^{1/2}$$

From this spectral presentation of  $\phi$  we see that  $\phi(\mu)$  is convex in  $(-\lambda_1, +\infty)$ . The following lemma is straightforward and shows the behavior of  $\phi$  at  $-\lambda_1$ .

LEMMA 2: If  $b \neq 0$  then  $\phi(\mu)$  is positive and strictly decreasing in  $(-\lambda_1, +\infty)$  with

$$\lim_{\mu \rightarrow \infty} \phi(\mu) = 0.$$

2. If  $b \notin \mathcal{N}(A - \lambda_1 I)^\perp$  then

$$\lim_{\mu \rightarrow (-\lambda_1)^+} \phi(\mu) = +\infty.$$

3. If  $b \in \mathcal{N}(A - \lambda_1 I)^\perp$  then

$$\lim_{\mu \rightarrow (-\lambda_1)^+} \phi(\mu) = \|x^+\|.$$

Suppose that for some  $\mu > \max\{0, -\lambda_1\}$  we have  $\phi(\mu) > r$ . Then  $\mu_* > -\lambda_1$  and the optimal solution to (QP) can be found by the Hebden algorithm which can be viewed as the Newton method for the equivalent zero finding problem

$$\psi(\mu) = \frac{1}{r} - \frac{1}{\phi(\mu)} = 0, \quad \text{for } \mu \in (-\lambda_1, +\infty). \quad (10)$$

A Hebden's step is computed by

$$\mu_+ = \mu + \frac{\phi(\mu)(r - \phi(\mu))}{\phi'(\mu)r}. \quad (11)$$

Typically, function  $\psi$  behaves very similarly to a piecewise linear function and strictly decreases on  $(-\lambda_1, +\infty)$  so Newton's method converges very rapidly to the solution. Moreover, the computation of the Cholesky factorization of  $A + \mu I$  makes it possible to implement (11) without explicit knowledge of the eigensystem of  $A$ .

**ALG. 1.** (Hebden algorithm)

Given  $\mu > \max\{0, -\lambda_1\}$  such that  $\phi(\mu) > r$ .

1. Factor  $(A + \mu I) = R^T R$ . Solve  $R^T R x = -b$  for  $p$ .
2. If  $\|x\| - r < \varepsilon$  then stop.
3. Otherwise, solve  $R^T y = x$ .
4. Compute

$$\mu_+ = \mu + \left( \frac{\|x\|}{\|y\|} \right)^2 \frac{(\|x\| - r)}{r}. \quad (12)$$

Set  $\mu = \mu_+$  and go to 1.

The Hebden algorithm is in fact a primal-dual algorithm in the sense that it provides both primal solution  $p$  and dual solution  $\mu$  to (QP). This algorithm works only if  $\mu_*$  is known to be in the interval  $(-\lambda_1, \infty)$ . Numerical difficulties arise when  $\mu_* = -\lambda_1$  and  $b \in \mathcal{N}(A - \lambda_1 I)^\perp$  because we have to deal with multiple solutions. This case was referred to as the hard case ([17]).

It should be noted that in general, an approximate solution can be easily obtained provided the eigensystem (or at least the smallest eigenvalue and a corresponding eigenvector) of the matrix  $A$  is known. Unfortunately, finding even only the smallest eigenvalue of a matrix is time consuming. We shall discuss in what follows two alternative approaches for dealing with problems ( $QP$ ) in more efficient way. The first one has been proposed by Moré & Sorensen [17] and the second one is due to Pham Dinh Tao ([23]-[25]). Both algorithms use for locating  $\mu_*$  a bisection-type process. They differ essentially by the choice of an tempting point and the way to involve the Hebden's process. The outline of these method is as follows.

Let  $[\mu_L, \mu_U]$  be an interval containing  $\mu_*$  and  $\mu \in (\mu_L, \mu_U)$ . There are 2 possible cases :

1.  $\mu > -\lambda_1$  so  $(A + \mu I)x = -b$  admits a solution  $x(\mu)$ .
  - a) If  $\|x(\mu)\| \geq r$  then the condition for applying Hebden's algorithm holds. That means  $\mu_* > \mu$  and Hebden's step will give another value more close to  $\mu_*$ .
  - b) If  $\|x(\mu)\| < r$  then  $\mu_* \leq \mu$ .

2.  $\mu \leq -\lambda_1$  then  $\mu_* \geq \mu$ .

A simplest way to update  $\mu_L, \mu_U$  is to set  $\mu_U = \mu$  if  $\mu > -\lambda_1$  and  $\|x(\mu)\| < r$  and  $\mu_L = \mu$  if  $\mu \leq -\lambda_1$ . It is easy to see that if  $\mu_U - \mu_L \rightarrow 0$  and  $\mu_* > -\lambda_1$  then 1.a) must occur after a finite number of steps.

Now consider the choice of  $\mu$ . In the case 1 we can take  $\mu = \mu_+$  a Hebden's step. But it is good only in the case 1.a) because in the case 1.b)  $\mu$  may not belong to the next interval  $[\mu_L, \mu_U]$ . Therefore, Moré & Sorensen proposed a so-called *safeguard* procedure in order to get a  $\mu$  satisfying :  $\mu > -\lambda_1$  and  $\mu \in (\mu_L, \mu_U)$ . Towards this end they used a parameter  $\mu_S \leq -\lambda_1$  and if  $\mu < \mu_S$  one sets  $\mu = \max\{0.001 \mu_U, (\mu_L, \mu_U)^{1/2}\}$  (note that  $\mu_U > \mu_L \geq 0$ ). This procedure ensures that if  $\mu_* > -\lambda_1$  then one must have 1.a) after a finite number of steps. Otherwise, one must have  $\mu > -\lambda_1$  which is very close to  $-\lambda_1$ . To check it, we can use the Cholesky decomposition  $R^T R$  of  $A + \mu I$ . If  $\mu$  is close to  $-\lambda_1$  then  $R^T R$  is nearly singular so it is possible to choose  $\hat{z}$  with  $\|\hat{z}\| = 1$  such that  $\|R\hat{z}\|$  is close to the smallest singular value of the triangular matrix  $R$  (e.g. by the LINPACK technique (cf. [17])). In particular, if  $R$  is nearly singular then  $\|R\hat{z}\|$  must be close to zero. It has been proposed to compute an approximate solution to ( $QP$ ) by taking  $x(\mu) + \xi \hat{z}$  such that  $\|x(\mu) + \xi \hat{z}\| = r$ .

Alternatively, Pham Dinh Tao proposed to use the dichotomy scheme ( $\mu = 1/2(\mu_L + \mu_U)$ ) and in case 1.a) to use Hebden's algorithm. If  $\mu_* = -\lambda_1$  then  $\mu_U - \mu_L \rightarrow 0$  so we can stop the process when  $\mu_U - \mu_L$

becomes sufficiently small and compute an approximate solution as above. This scheme is simpler and does not require the nonnegativity of  $\mu_L, \mu_U$ .

#### 4.1. Safeguarding algorithm (Moré & Sorensen)

The following algorithm has been developed in [17] (see also [15], [8]) in connection with Trust Region methods. The main ingredients of the algorithm are the following :

*Safeguard* : Given an attempting  $\mu$ . Then

1.  $\mu := \max \{ \mu, \mu_L \}, \quad \mu := \min \{ \mu, \mu_U \}$
2. If  $\mu \leq \mu_S$  then  $\mu := \max \{ 0.001 \mu_U, (\mu_L \mu_U)^{1/2} \}$ ;

*Approximate solution* : Let  $R^T R$  be the Cholesky factorization of  $A + \mu I$  ( $\mu > -\lambda_1$ ) and  $\hat{z}$  be the vector obtained by the LINPACK technique for estimating the smallest singular value of the triangular matrix  $R$ . If  $\|R\hat{z}\|$  is close to zero then  $x(\mu) + \xi \hat{z}$  with  $\xi$  satisfying  $\|x(\mu) + \xi \hat{z}\| = r$  is an approximate solution.

*Update  $\mu_L, \mu_U, \mu_S$*  :

1. If  $\mu > -\lambda_1$  (i.e.  $A + \mu I$  is positive definite) then if  $\phi(\mu) < r$  (i.e.  $\psi(\mu) < 0$ ) then

$$\mu_U = \min \{ \mu_U, \mu \}, \quad (13)$$

$$\mu_S = \max \{ \mu_S, \mu - \|R\hat{z}\|^2 \}, \quad (14)$$

$$\mu_L = \max \{ \mu_L, \mu \}, \quad (15)$$

else

$$\mu_L = \max \{ \mu_U, \mu \}. \quad (16)$$

2. If  $\mu < -\lambda_1$  then during the Cholesky decomposition of  $A + \mu I$ , it is possible to find  $\theta > 0$  and a vector  $u \in \mathbb{R}^n$  such that

$$u^T (A + \mu I + \theta e_i e_i^T) u = 0$$

with  $u_i = 1$  and  $u_i = 0 \forall i > l$  ( $\{e_i : i = 1, \dots, n\}$  is the canonical basis of  $\mathbb{R}^n$  (cf. Gay [8])). Set

$$\mu_S := \max \left\{ \mu_S, \mu + \frac{\theta}{\|u\|^2} \right\}. \quad (17)$$

3. Let  $\mu_L := \max \{ \mu_L, \mu_S \}$

*Initialization*

$$\mu_S = \max \{-A_{ii}, i = 1, \dots, n\}, \quad (18)$$

$$\mu_L = \max \left\{ 0, \mu_S, \frac{\|b\|}{r} - \|A\|_1 \right\}, \quad (19)$$

$$\mu_U = \frac{\|b\|}{r} + \|A\|_1, \quad (20)$$

where  $\|A\|_1 = \max \left\{ \sum_{i=1}^n |A_{ij}|, j = 1, \dots, n \right\}$  is the matrix norm relative to the vector norm  $l_1$  in  $\mathbb{R}^n$ .

**ALG 2.**

• Initialization : Let  $\sigma_1, \sigma_2 \in (0, 1)$  be given

— Compute  $\mu_S, \mu_L, \mu_U$  by (18), (19), (20).

— Let  $\mu := \mu_L$ .

• Iteration :

k.1 Safeguard  $\mu$ .

k.2 If  $\mu > -\lambda_1$  then factor  $(A + \mu I) = R^T R$  and solve  $R^T R x = -b$ .

k.2.1 If

$$\|\|x\|\| - r \leq \sigma_1 r \quad \text{or} \quad (\|x\| < r \quad \text{and} \quad \mu = 0) \quad (21)$$

then stop :  $x$  is a solution of  $(QP)$ .

k.2.3 If  $\|x\| < r$  then compute  $\hat{z}, \xi$ . If

$$\|R(\xi \hat{z})\|^2 \leq \sigma_1 (2 - \sigma_1) \max \{ \sigma_2, (\|R x\|^2 + \mu r^2) \} \quad (22)$$

then stop :  $x + \xi \hat{z}$  is a solution of  $(QP)$ .

k.4 Update  $\mu_L, \mu_U, \mu_S$ .

k.5 If  $\mu \leq -\lambda_1$  or  $b = 0$  then  $\mu = \mu_S$ . Otherwise  $\mu = \mu_+$ . Return to k.1.

*Remark 1 :* If  $A$  is positive semi-definite and  $b \neq 0$ , then we can set all  $\mu_S = 0$ . Thus, we obtain a version of Moré originally proposed for solving nonlinear least squares problems (cf. [15]). It should be noted in this case  $A = J^T J$  where  $J$  is the Jacobian matrix of the nonlinear system. The QR-decomposition technique was then used instead of the Cholesky factorization. As pointed out in Moré [15], the QR decomposition is more expensive but works well in the case where  $J$  is rank deficient.

It has been shown in [17] that ALG 2 terminates after a finite number of iterations yielding an approximate solution which satisfies

$$q(x) - q^* \leq \sigma_1(2 - \sigma_1) \max\{|q^*|, \sigma_2\} \quad \text{and} \quad \|x\| \leq (1 + \sigma_1)r \quad (23)$$

where  $q^*$  is the optimal value and  $\sigma_1, \sigma_2 \in (0, 1)$ .

#### 4.2. Dichotomy algorithm (Pham Dinh Tao)

Note that if  $A$  is positive semi-definite or positive definite (which can easily verified using the Cholesky factorization) then by Proposition 3 we can check if  $\mu_* > 0$  and then apply Hebden's algorithm. The dichotomy procedure will be used essentially for handling the case where  $A$  is not positive semi-definite ([23]-[25]).

##### ALG 3.

Let  $\sigma_1, \varepsilon \in (0, 1)$  be given.

1. If  $A$  is positive definite then compute  $x = -A^{-1}b$ .
  - a) If  $\|x\| - r \leq \sigma_1 r$  then stop:  $x$  is a solution to  $(QP)$ .
  - b) Otherwise apply ALG 1 to get a solution of  $(QP)$ .
2. If  $A$  is only positive semi-definite take  $\gamma > 0$  sufficiently small. Compute  $x = -(A + \gamma I)^{-1}b$ .
  - a) If  $\|x\| - r \leq \sigma_1 r$  then stop:  $x$  is a solution to  $(QP)$ .
  - b) Otherwise apply ALG 1 to get a solution of  $(QP)$ .
3. If  $A$  is not positive semi-definite then compute  $\mu_L, \mu_U$ .

*Iteration*: (Dichotomy procedure)

3.1 If  $(\mu_U - \mu_L) \leq \varepsilon(\mu_L + \mu_U)$  then go to 3.6.

3.2 Set  $\mu = \frac{1}{2}(\mu_L + \mu_U)$ .

3.3 If  $\mu > -\lambda_1$  then factor  $(A + \mu I) = R^T R$  and solve  $R^T R x = -g$ .

3.3.1 If

$$\|\|x\| - r\| \leq \sigma_1 r \quad \text{or} \quad (\|x\| \leq r \quad \text{and} \quad \mu = 0) \quad (24)$$

then stop:  $x$  is a solution of  $(QP)$ .

3.3.2 If  $\|x\| > r$  then apply ALG 1 to get a solution of  $(QP)$ . Stop.

3.3.3 Otherwise set  $\mu_U = \mu$  and go to 3.5.

3.4 If  $\mu \leq -\lambda_1$  then set  $\mu_L = \mu$  and go to 3.5.

3.5 Return to 3.1.

3.6 Compute  $x + \xi \hat{z}$  by the LINPACK technique. Stop.

*Comment 1:* a) If  $\mu > -\lambda_1$  and  $\phi(\mu) < r$  then either  $\mu_+ \leq -\lambda_1$  or  $(\mu_+ > -\lambda_1$  and  $(\phi(\mu_+) \geq r)$ . This fact can be used for updating  $\mu_L$  as follows:

1. If  $\mu > -\lambda_1$  then set  $\mu_L = \max\{\mu_L, \mu_+\}$  and  $\mu_U = \mu$ .
2. If  $\mu \leq -\lambda_1$  then

$$\mu_L = \max \left\{ \mu_L, \mu + \frac{\theta}{\|u\|^2} \right\} \quad (25)$$

where  $\theta, u$  are calculated as above.

b) In step k.1 when  $\mu_U - \mu_L$  is small enough  $\mu_U (> -\lambda_1)$  is closed to  $-\lambda_1$  and an approximate solution satisfying criterion (22) can be obtained for an appropriately chosen  $\varepsilon$ . Thus (23) remains valid for ALG 3. The choice of  $\varepsilon$  is very important. A smaller value of  $\varepsilon$  requires more iterations. But  $\varepsilon$  not sufficiently small may give unsatisfactory solutions. It suggests then to perform some more dichotomy steps to obtain a solution verifying criterion (22).

#### 4.3. Discussion

1. Our recent numerical experiments for large-scale problems with dimension up to 500 ([25]) shown the superiority of ALG 3 with regard to ALG 2, especially in the hard case. Also, it was observed that the Cholesky factorization is rather expensive ( $O(n^3/6)$ ) and the convergence of the method becomes slow as the problem dimension is sufficiently large. An alternative is to use the following decomposition of  $A$  by Lanczos' algorithm (cf. [9]):

$$A = QSQ^T$$

where  $S$  is a tridiagonal matrix and  $Q$  is an orthogonal matrix. By replacing  $y = Q^T x$ , problem (QP) can be rewritten as

$$\min \left\{ \frac{1}{2} y^T S y + \bar{b}^T y : \|y\| \leq r \right\} \quad (26)$$



where  $\bar{b} = Q^T b$ . Obviously, if  $y^*$  is a solution of (26) then  $x^* = Qy^*$  is a solution of (QP). Instead of the Cholesky factorization we shall decompose  $S + \mu I = LDL^T$  where  $D = \text{diag}(d_1, \dots, d_n)$  and

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ e_1 & 1 & \dots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & e_{n-1} & 1 \end{pmatrix}$$

being a lower triangular matrix. If all  $d_i > 0$  then  $S + \mu I$  is positive definite and solving  $(S + \mu I)y = -\bar{b}$  is reduced to solving systems:

$$Lw = -\bar{b}, \quad Dz = w, \quad L^T y = z.$$

A Hebden's step (see eq. (12)) is then defined by

$$\mu_+ = \mu + \frac{\|y\|^2 (\|y\| - r)}{\langle z, w \rangle r}.$$

This approach requires only  $O(5n)$  operations for the decomposition of  $S + \mu I$  and for solving  $(S + \mu I)y = -\bar{b}$ , so it can reduce considerably the execution time. Nevertheless, due to the error in decomposition by the Lanczos method one may obtain  $Q$  which is not orthogonal, hence,  $x^* = Qy^*$  may not be a good solution. However in some applications (e.g. in TR methods), we need only an approximate solution to problem (QP). This may justify a further research on the reliability of this approach for large scale practical problems.

2. Problem (QP) is closely related to another nonconvex quadratic programming problem

$$(\overline{QP}) \quad \min \left\{ \frac{1}{2} x^T A x + b^T x : \|x\| = r \right\}.$$

At first glance, the latter problem seems to be more difficult because of the equality constraint which makes the feasible set no longer convex. In fact, two problems are equivalent in the sense that, if (QP) admits an optimal solution with the norm equal to  $r$  then this solution solves  $(\overline{QP})$ . On the other hand, we can always choose a  $\rho$  such that  $A + \rho I$  is non positive semi-definite (i.e.  $\rho + \lambda_1 < 0$ ) so  $(\overline{QP})$  is equivalent to the problem (see Theorem 2)

$$\min \left\{ \frac{1}{2} x^T (A + \rho I) x + b^T x : \|x\| \leq r \right\}$$

which has the form  $(QP)$ . It is not difficult to derive similar results, such as the optimality condition, the stability in Lagrangian duality and complete characterizations of the solution set for problem  $(\overline{QP})$ . Also based upon these results, we can develop an algorithm of the dichotomy type for solving it. It should be noted that in the case of equality constraint, the dual solution  $\lambda^*$  as in Theorem 2 needs not be nonnegative and the safeguarding algorithm is not ready to be applied to problem  $(\overline{QP})$ .

##### 5. TRUST REGION MODEL APPROACH IN UNCONSTRAINED OPTIMIZATION

This section concerns the problem of finding a solution to the problem

$$(P) \quad \min \{f(x) : x \in \mathbb{R}^n\}$$

where  $f(x)$  is assumed to be twice continuously differentiable on  $\mathbb{R}^n$ . Most of solution methods are iterative and based upon

- a *model*, that is some convenient approximation to the objective function, which enables a prediction of the location of a local minimizer to be made,
- a *prototype* which describes the broad strategy of the approach, i.e., how to use the model prediction in such a way as to obtain satisfactory convergence properties.

A widely used model is quadratic, i.e.

$$q(d) = g^T d + \frac{1}{2} d^T G d \quad (27)$$

where  $g \in \mathbb{R}^n$  and  $G$  is  $n \times n$  symmetric matrix such that

$$f(x+d) \approx f(x) + q(d)$$

in some neighborhood of  $x$ . It is natural to take  $g$  as the gradient of  $f$  at  $x$  and  $G$  as the Hessian of  $f$  at  $x$  (so  $f(x) + q(d)$  is a truncated Taylor series expansion of  $f(x)$  about  $x$ ).

The Newton method for unconstrained minimization is effectively Newton's method applied for finding a zero of the gradient of the objective function. The iterate  $x^{k+1}$  is simply taken to be  $x^k + d^k$  where  $d^k = -G_k^{-1} g_k$ . The local rate of convergence of this iteration is quadratic. Unfortunately, since the objective function is generally non convex, the initial iterate must be very « close » to a local minimizer in order to ensure the convergence of the iteration. Moreover,  $x^{k+1}$  is well defined only if  $G_k$  is positive definite and even in this case  $f(x^{k+1})$  may not decrease.

The approach we present below is well known and appropriately called a trust region model approach. This method, as the Newton method, is based on the quadratic model, but is generally applicable and globally convergent while

it retains the rapid rate of convergence of the latter. The development of TR methods can be traced back to the work of Levenberg (1944) and Marquardt (1963) on nonlinear least squares problems.

### 5.1. Trust region prototype

The idea underlying the TR approach is to compute a step  $d^k$  by solving the problem

$$(P_k) \quad \min \{q_k(d) : \|d\| \leq r_k\}$$

where

$$q_k(d) = 1/2 d^T G_k d + g_k^T d$$

$G_k$  and  $g_k$  are the Hessian matrix and the gradient vector of  $f$  at  $x^k$ . The trust radius  $r_k$  is determined adaptively according to a certain measure of agreement between  $q_k(d^k)$  and  $f(x^k + d^k)$ . The measure is generally defined by the following quantity, called « quality coefficient »

$$r_k = \frac{\Delta f_k}{\Delta q_k} = \frac{f(x^k) - f(x^k + d^k)}{q_k(0) - q_k(d^k)} \quad (28)$$

which is the ratio between the *actual reduction* in  $f$  on the  $k$ th step and the corresponding *predicted reduction* in the quadratic model.

#### Prototype

- Let  $x^0 \in \mathbb{R}^n$  and  $r_0$  be given.
- For  $k = 0, 1, \dots$ ,
  - k.1 Compute  $f(x^k)$  and the quadratic model  $q_k$ .
  - k.2 Determine a solution  $d^k$  to problem  $(P_k)$
  - k.3 Compute  $r_k$  via (28).
  - k.4 If  $r_k \leq s_1$  then  $x^{k+1} = x^k$  and  $r_{k+1} = r_k/2$ .  
 Otherwise  $x^{k+1} = x^k + d^k$ .

If  $r_k > s_2$  then  $r_{k+1} = 2 r_k$ . Otherwise  $r_{k+1} = r_k$ .  
 Here,  $s_1, s_2$  are prechosen positive number (e.g. in our implementation  $s_1 = 0.25, s_2 = 0.75$ ).

The main source of computational effort in the above prototype is the work on the problem  $(P_k)$  which is of the form  $(QP)$ . We will see that the approximate solution obtained by the methods described in the preceding section is sufficient to guarantee the good convergence properties of TR algorithms.

## 5.2. Global convergence properties

The following result is due to Moré & Sorensen [17] (see also Sorensen [22], Moré [16], Fletcher [7]).

Assume that the step  $d^k$  satisfies

$$q_k(d^k) \leq \beta_1 \min \{q_k(d) : \|d\| \leq r_k\}, \quad \|d^k\| \leq \beta_2 r_k \quad (29)$$

where the constants  $\beta_1, \beta_2$  are small positive numbers.

**THEOREM 3:** *Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be twice continuously differentiable and bounded below on  $\mathbb{R}^n$ , and assume that  $\nabla^2 f$  is bounded on the level set*

$$\{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}.$$

Let  $\{x^k\}$  be the sequence generated by TR algorithms with  $G_k = \nabla^2 f(x^k)$  and  $d^k$  satisfying (29). Then

1.  $\lim_{k \rightarrow +\infty} \|\nabla f(x^k)\| = 0$ .
2. If  $\{x^k\}$  is bounded then there is a limit point  $x^*$  with  $\nabla^2 f(x^*)$  positive semi-definite.
3. If  $x^*$  is an isolated limit point of  $\{x^k\}$  then  $\nabla^2 f(x^*)$  is positive semi-definite.
4. If  $\nabla^2 f(x^*)$  is non-singular for some limit point  $x^*$  of  $\{x^k\}$  then
  - a)  $\nabla^2 f(x^*)$  is positive definite,
  - b)  $\lim_{k \rightarrow +\infty} x^k = x^*$  and there exists a  $\varepsilon > 0$  and  $K$  such that  $r_k > \varepsilon$ ,  $\forall k > K$ .
  - c) the convergence is superlinear.

Clearly, a solution  $d^k$  satisfying (23) will satisfy (29) with  $\beta_1 = (1 - \sigma_1)^2$ ,  $\beta_2 = 1 + \sigma_1$ . For large scale problems the cost of obtaining a step  $d^k$  may be prohibitive. An alternative is to determine  $d^k$  by solving the problem

$$\min \{q_k(d) : d \in S_k, \|d\| \leq r_k\}.$$

In Moré [16] a variation on Theorem 2 was established under the fairly mild condition

$$q_k(d^k) \leq \beta_1 \min \{ q_k(d) : d = \tau g_k, \|d\| \leq r_k \}, \quad \|d^k\| \leq \beta_2 r_k. \quad (30)$$

In particular, it has been shown in Schultz *et al.* [20] that if  $x^k$  converges to  $x^*$  then  $\nabla^2 f(x^*)$  is positive semi-definite if  $d^k$  satisfies (30) and

$$q_k(d^k) \leq \beta_1 \min \{ q_k(d) : d = \tau u_1, \|d\| \leq r_k \} \quad (31)$$

where  $u_1$  is an eigenvalue associated with the smallest eigenvalue of  $\nabla^2 f(x^k)$ .

### 5.3. Practical algorithm for nonlinear least squares problem

A special case of unconstrained optimization is that the objective function  $f(x)$  is a sum of squares of nonlinear functions

$$f(x) = \frac{1}{2} \sum_{j=1}^m \Phi_j^2(x) = \frac{1}{2} \|\Phi(x)\|^2 \quad (32)$$

where  $\Phi(x) = (\Phi_1(x), \dots, \Phi_m(x))^T$  and  $\|\Phi(x)\|$  is termed the *residual* at  $x$ . Such problems arise as well from attempt of solving the system

$$\Phi_j(x) = 0, \quad j = 1, \dots, m$$

(e.g. the problems in Computer Vision we are going to discuss in the next section). It should be noted that when  $m \gg n$  it is usually impossible to obtain an exact solution.

Assume that  $\Phi_i$ ,  $i = 1, \dots, m$  are twice continuously differentiable on  $\mathbb{R}^n$ . It is worth noticing that the gradient and the Hessian matrix of  $f(x)$  have a special structure. Let  $J(x) = (\nabla \Phi_1(x), \dots, \nabla \Phi_m(x))^T$  denote the  $m \times n$  Jacobian matrix of  $\Phi(x)$ . Then

$$\nabla f(x) = J(x)^T \Phi(x), \quad (33)$$

$$\nabla^2 f(x) = J(x)^T J(x) + Q(x), \quad (34)$$

where  $Q(x) = \sum_{j=1}^m \Phi_j(x) \nabla^2 \Phi_j(x)$  contains second-order information. An important case in a large number of applications is the *small residual* problem where the residual  $\|\Phi(x^*)\|$  at the solution is « small », so the first-order term  $J(x)^T J(x)$  of (34) will *dominate* the second-order term  $Q(x)$ . Thus the Hessian of  $f$  can be approximated by  $G = J(x)^T J(x)$  which is generally

positive semi-definite. Therefore, a solution to Problem  $(P_k)$  (a TR step) can be computed simply as described in case 2. (ALG 3). Thus we obtain the following practical TR-based algorithm (cf. Pham D. Tao *et al.* [1], [21], [26]).

**ALG 4**

- *Initialization* : Let  $\varepsilon_g, \varepsilon_f, \sigma, \gamma > 0$  be small positive numbers.  
Choose the initial iterate  $x^0$ , the initial trust radius  $r_0 > 0$ . Set  $k := 0$ .
- *Iteration* :  $k := 0, 1, \dots$ 
  - k.1 Compute  $f_k = f(x^k)$ ,  $g_k = J(x^k)^T \Phi(x^k)$  and  $G_k = J(x^k)^T J(x^k)$  (cf., (33), (34)).
  - k.2 If  $\|g_k\| \leq \varepsilon_g$  or  $r_k \leq \varepsilon_r$  or  $f_k \leq \varepsilon_f$  then stop:  $x^k$  is a solution.
  - k.3 Let  $d$  be a solution of the system

$$(G_k + \gamma I) d = -g_k.$$

If  $\|d\| \leq r_k$  then  $d^k = d$ . Otherwise, using the Hebden algorithm to find a  $\mu > 0$  so that the solution of  $(G_k + \mu I) d = -g_k$  satisfies  $\|d\| - r_k \leq \sigma r_k$  and set  $d^k = d$ .

- k.4 Compute  $r_k$  using eq. (28).
- k.5 If  $r_k \geq 0.25$  then

$$x^{k+1} = x^k + d^k;$$

if  $r_k \geq 0.75$  then

$$r_{k+1} = 2 r_k$$

otherwise

$$r_{k+1} = r_k.$$

Set  $k = k + 1$  and return to k.1

- k.6 If  $r_k < 0.25$  then

$$r_{k+1} = r_k / 2$$

and return to k.3.

*Comment 2* :  $\gamma$  should be chosen sufficiently small to get a good approximate solution  $d^k$  to  $(P_k)$  which may improve the convergence of TR algorithms (e.g. in our implementation  $\gamma = 10^{-7}$ ). If  $\|d\| < r$  (Step k.3) then  $d^k$  can be also computed using the LINPACK technique.

*Remark 2* : In Step k.6,  $r_k$  is reduced but  $x^k$  is not changed, so  $(P_{k+1})$  differs from  $(P_k)$  only by the constraint  $\|x\| \leq r_{k+1}$ . In this case  $d^k$  (and its associated dual solution) can be used to compute a solution to  $(P_{k+1})$ . More precisely, if  $d^k \leq r_{k+1}$  then  $d^{k+1} = d^k$ . Otherwise,  $d^{k+1} > r_{k+1}$  and the Levenberg-Marquardt algorithm can be applied.

Levenberg [12] and Marquardt [14] were the first to propose a reasonable algorithm for the nonlinear least squares problem. In their method the step  $d^k$  is a solution of :

$$(J(x^k)^T J(x^k) + \lambda_k I) d^k = -J(x^k)^T \Phi(x^k)$$

for some  $\lambda_k \leq 0$ . Marquardt proved that  $d^k$  defined in this way is in fact a solution to the constrained subproblem :

$$\min \left\{ \frac{1}{2} \|J(x^k) d + \Phi(x^k)\|^2, \text{ s.t. } \|d\| \leq r_k \right\}$$

for some  $r_k = \|d^k\|$  related to  $\lambda_k$ . He proposed then to control  $d^k$  indirectly by changing  $\lambda_k$ . It is easy to see that this problem can be rewritten under the form  $(P_k)$  with  $G_k = J(x^k)^T J(x^k)$ . But, contrary to TR model approach, here  $\lambda_k$  is used as the controlling parameter instead of  $d^k$ . As noted in Moré [15], most implementations of the Levenberg-Marquardt algorithm are either not robust, or do not have a solid theoretical justification. Specifically, it does not seem to be a reasonably automatic choice of  $\lambda_0$  and a reasonable value of  $r_0$  is quite often more less than  $\|d^0\|$ . Moreover when  $x^k + d^k$  leads to an increase in  $f$  it is not clear how this information can be used to estimate  $\lambda_{k+1} > \lambda_k$ . In his work, Moré proposed a very elegant and robust algorithm for nonlinear least squares. Our practical TR-based algorithm differs from Moré's algorithm only by the way to compute the step  $d^k$ . The implementation of the algorithm seems to be simpler than that described in Moré [15].

In practice, due to presence of noise, the function value  $f(x^*)$  at a minimizer may be not « small » as desired and the second term  $Q(x)$  in (17) can not be neglected. Thus, when the residual at the solution are very « large », the exact Hessian (*cf.* eq. (34)) of  $f$  should be used which is no longer positive semi-definite. In this case, a TR step in Step k.3 could be computed by one of two algorithms described in the preceding section.

## 6. PROBLEM MODELS

## 6.1. Camera model

The camera is modeled such that a point in space  $(X, Y, Z)^T$ , its projection  $(U, V)$  and the projection center should be aligned, this condition can be expressed as follows :

$$U = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}},$$

$$V = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}.$$

If homogeneous coordinates, that allow us to manipulate indifferently the points at infinity, are used both for points in space and points in image, the projection can then be nicely modeled as a linear mapping between two projective spaces of different dimensions. The object space  $R^3$  may be considered as embedded in  $\mathcal{P}^3$  by the natural mapping  $(X, Y, Z)^T \mapsto (X, Y, Z, 1)^T = \lambda(x, y, z, t)^T$  and image space  $R^2$  embedded in  $\mathcal{P}^2$  by  $(U, V)^T \mapsto (U, V, 1)^T = \lambda(u, v, w)^T$ ,  $\lambda$  is any nonzero constant. This linear projection from  $\mathcal{P}^3$  upon  $\mathcal{P}^2$  can be represented by a  $3 \times 4$  matrix  $M_{3 \times 4}$  of rank 3 whose kernel is the projection center. The relation between the points  $P_i$  in  $\mathcal{P}^3$  and  $p_i$  in  $\mathcal{P}^2$  can be written as

$$\lambda_i p_i = M_{3 \times 4} P_i,$$

where  $p_i$  and  $P_i$  are in homogeneous coordinates. i.e.

$$\lambda_i \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ t_i \end{pmatrix}.$$

This can be rewritten in ratio form hiding the scaling factor  $\lambda_i$ ,

$$\begin{aligned} u_i : v_i : w_i &= (m_{11}x_i + m_{12}y_i + m_{13}z_i + m_{14}t_i) : \\ &(m_{21}x_i + m_{22}y_i + m_{23}z_i + m_{24}t_i) : \\ &(m_{31}x_i + m_{32}y_i + m_{33}z_i + m_{34}t_i). \end{aligned} \quad (35)$$

For each point, as  $u_i$ ,  $v_i$  and  $w_i$  can not all be zero, two independent equations can always be derived from (35).



The *calibration* process consists of the determination of all the parameters  $m_{ij}$  of the projection matrix  $M_{3 \times 4}$ . When the camera is calibrated, we know how to project any space point and how to relate a space line going through the projection center for any image point. When we are talking about *uncalibrated cameras*, it is meant that  $M_{3 \times 4}$  is totally unknown.

A few more words about calibration,  $M_{3 \times 4}$  matrix has 11 independent parameters as it is defined up to a scaling factor. These 11 parameters all have physical interpretation. To see this, we can uniquely decompose (by QR decomposition) the projection matrix into the following form

$$M_{3 \times 4} = \underbrace{\begin{pmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_A \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \underbrace{\begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix}}_D.$$

$A$  describes an affine transformation of the coordinate system in image plane, whose five entries are called the intrinsic parameters of the camera.  $(u_0, v_0)$  are the pixel coordinates of the principal point.  $k_u$  and  $k_v$  are respectively the focal length of the camera in terms of horizontal and vertical pixel size.  $s$  is introduced to rectify the non-perpendicularity of the  $u$  and  $v$  axes. Even more interesting, the knowledge of the intrinsic parameters is equivalent to that of the absolute conic in  $\mathcal{P}^3$ ,  $x^2 + y^2 + z^2 = 0 = t$ . The projection of the absolute conic is also a conic whose matrix is given by  $A^{-1}(A^{-1})^T$ . We will not detail more since we will be developing our work with uncalibrated cameras within a projective geometry framework instead of Euclidean geometry framework in which most of computer vision work was developed.

$D$  describes a transformation of three-dimensional orthonormal coordinate system. It is a rigid displacement, it counts for the six extrinsic parameters of the camera, three for rotation (matrix  $R$ ) and three for translation (vector  $\mathbf{t}$ ). This totalizes the 11 parameters of  $M_{3 \times 4}$ , as a consequence,  $M_{3 \times 4}$  describes the most general projective camera. For instance, the commonly used *CCD* cameras can be modeled by 10 out of 11 parameters with  $s = 0$ .

The various vision problems can be classified into two classes according to whether the intrinsic parameter matrix  $A$  is provided or not.

## 6.2. Problems in computer vision

Generally, in the following, for each case, two problems are given, the first one corresponds to the case where cameras are calibrated and the second one

to the case where cameras are not calibrated. When cameras are not calibrated, for each image point, we use homogeneous coordinates  $(u_p, v_p, w_i)$  to emphasize that the image plane can be defined up to a projective transformation.

### 6.2.1. Camera calibration from 2D/3D correspondences

PROBLEM 1: Given  $(U_p, V_i) \leftrightarrow (X_p, Y_p, Z_i)$ , find  $M_{3 \times 4}$ .

This is called camera calibration in computer vision.

This problem can be solved by linear least squares [6], so we will not further deal with this problem. Once  $M_{3 \times 4}$  is estimated,  $A$ ,  $R$  and  $t$  can be easily extracted as we have mentioned above.

### 6.2.2. Two images

PROBLEM 2: Given  $m_i = (U_p, V_i)^T \leftrightarrow m'_i = (U'_p, V'_i)^T$ ,  $A$  and  $A'$ , find  $E$  such that  $m_i^T E m'_i = 0$ , with  $\text{rank}(E) = 2$  and  $\sigma_1 = \sigma_2$ ,  $\sigma_i$  are singular values of  $E$  [10].

$E$  is called the essential matrix. The problem is generally called motion estimation, since once  $E$  is recovered, the rigid displacement between two images can be obtained by decomposing  $E$ .

PROBLEM 3: Given  $m_i = (u_p, v_p, w_i)^T \leftrightarrow m'_i = (u'_p, v'_p, w'_i)^T$ , find  $F$  such that  $m_i^T F m'_i = 0$ , with only  $\text{rank}(F) = 2$ .

Some call the matrix  $F$  the fundamental matrix [5]. The problem is generally considered as epipolar geometry determination for uncalibrated images, or the weak calibration of the two images.

Note the difference between Problem 2 and Problem 3, the first one needs the intrinsic parameters of the images and the second one does not. The essential matrix  $E$  is constrained with the two equal singular values, whereas the fundamental matrix  $F$  is not.

These two problems are also equivalent to the reconstruction problems from two images. From  $E$ , a Euclidean reconstruction of space points is possible from two images and a projective reconstruction of space points is obtained from  $F$ .

### 6.2.3. Multiple images

PROBLEM 4: Given  $(U_p, V_i)^{(1)} \leftrightarrow (U_p, V_i)^{(2)}, \dots, (U_p, V_i)^{(n)}$ , for  $i = 1, \dots, m$ ,  $A^1, \dots, A^n$ , find  $(X_p, Y_p, Z_i)$  and  $M^{(j)}$ .

This is the general reconstruction problem from multiple images in computer vision.

PROBLEM 5: Given  $(u_p, v_p, w_i)^{(1)} \leftrightarrow (u_p, v_p, w_i)^{(2)}, \dots, (u_p, v_p, w_i)^{(n)}$ , for  $i = 1, \dots, m$ , find  $(x_p, y_p, z_p, t_i)$  and  $M^{(j)}$ . Of course,  $M^{(j)}$  are only defined up to a space collineation.

Problem 5 is a general projective reconstruction schema from a sequence of uncalibrated images. As can be easily proved, the solution is not unique in the problem 5 if we do not fix five points of them. The knowledge of these five reference points determine the geometric nature of the solutions. When they are assigned projective canonical coordinates, the reconstruction is projective. While they are assigned to usual Euclidean coordinates, the process can be qualified as self-calibration with reference points.

### 6.3. Modelisation

The problems 1 can be nicely formulated as linear least squares problems. The problems 2 and 4 deal with motion and reconstruction with calibrated images. Many special formulations have been proposed in the computer vision literature. They are difficult to be put into a general model since it involves different parameterization of 3D rigid displacements. More recent works on vision are concentrated on motion and reconstruction from uncalibrated images [18, 3]. In this domain, the two problems 3 and 5 are two major ones. These two problems will further be modeled.

#### 6.3.1. Problem 3-Weak calibration of a pair of images

Given  $n$  points of two images  $m_i = (u_i, v_i, w_i)^T \leftrightarrow (u'_i, v'_i, w'_i)^T$ , estimate the unknown matrix  $F$ .

Different parameterizations of  $F$  and choice of objective function leads to numerous minimization formulations. For instance, the problem can be formulated as

$$\phi_1 = \min \sum_i (m_i^T F m_i)^2$$

or

$$\phi_2 = \min \sum d_i^2(m'_i, F m_i) + d_i^2(m_i, F^T m'_i)$$

where  $d_i(m_i, l_i)$  is the Euclidean distance of a point  $m_i$  to a line  $a_i x + b_i y + c_i = 0$  where  $(a_i, b_i, c_i)^T = F m_i$  and

$$d_i^2 = \frac{(a_i u_i + b_i v_i + c_i w_i)^2}{a_i^2 + b_i^2}$$

These functions should be subject to the constraint which says that the rank of  $F$  is of 2. One way to write out this constraint is to parameterize  $F$  (up to a suitable row permutation) as

$$\begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ h_1 f_{11} + h_2 f_{21} & h_1 f_{12} + h_2 f_{22} & h_1 f_{13} + h_2 f_{23} \end{pmatrix}. \quad (36)$$

### 6.3.2. Problem 5-Projective reconstruction

The problem can be formulated (cf. [18]) as minimizing

$$\phi(x_p, y_p, z_p, t_p, m_{11}^{(j)}, \dots, m_{34}^{(j)}) = \sum_{k=1}^{2 \times m \times n} f_k^2(u_{ij}, v_{ij}; x_p, y_p, z_p, t_p, m_{11}^{(j)}, \dots, m_{34}^{(j)})$$

over

$$(x_p, y_p, z_p, t_p, m_{11}^{(j)}, \dots, m_{34}^{(j)}) \quad \text{for } i = 1, \dots, m, j = 1, \dots, n;$$

where  $f_k(\cdot)$  is either

$$u_{ij} = \frac{m_{11}^{(j)} x_i + m_{12}^{(j)} y_i + m_{13}^{(j)} z_i + m_{14}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}$$

or

$$v_{ij} = \frac{m_{21}^{(j)} x_i + m_{22}^{(j)} y_i + m_{23}^{(j)} z_i + m_{24}^{(j)} t_i}{m_{31}^{(j)} x_i + m_{32}^{(j)} y_i + m_{33}^{(j)} z_i + m_{34}^{(j)} t_i}.$$

This minimization should be subject to some constraints on  $x_p, y_p, z_p, t_p$  which are defined up to a scaling factor and  $m_{ij}$ , also defined up to a scaling factor.

One possible definition of these constraints can be

$$x_i^2 + y_i^2 + z_i^2 + t_i^2 - 1 = 0 \quad \text{for } i = 1, \dots, n$$

for  $x_p, y_p, z_p, t_p$ . For  $m_{ij}$ , we can impose  $m_{34} = 1$ .

As for the five reference points, we can just put the known coordinates of the five reference points into  $f_k(\cdot)$ .  $n' = n - 5$ .

$$\phi(x_p, y_p, z_p, t_p, m_{11}^{(j)}, \dots, m_{33}^{(j)}) = \sum_{k=1}^{2 \times m \times n'} f_k^2(u_{ij}, v_{ij}; x_p, y_p, z_p, t_p, m_{11}^{(j)}, \dots, m_{34}^{(j)}) + \lambda \sum_{i=1}^m (x_i^2 + y_i^2 + z_i^2 + t_i^2 - 1)^2.$$

## 7. EXPERIMENTAL RESULTS AND DISCUSSION

The TR-based algorithm was applied early to the object pose from a single view problem. A detailed experiment for this problem has been presented in Phong *et al.* [21]. It was shown that the method is particularly well-suited for the considered problem where the error function is a sum of squares of quadratic function. In this section we give some preliminary numerical results for the weak calibration problem defined in the preceding section. It should be noted that in this case the function to be minimized is more complicated.

Given  $n$  point correspondences between two images  $m_i \leftrightarrow m'_i (i = 1, \dots, n)$ . As the usual image points cannot be at infinity, i.e.  $w_i \neq 0$ , we can always suppose that the image points have the coordinates  $(u_i, v_i, 1)$ . Recall that the weak calibration problem is to estimate the  $3 \times 3$  fundamental matrix  $F$ . As was shown 6.3.1 this problem is reduced to solving the problem

$$\min \sum d_i^2(m'_i, Fm_i) + d_i^2(m_i, F^T m'_i)$$

where  $d(p, l)$  is the Euclidean distance of a point  $p$  to a line  $ax + by + c = 0$  ( $l = (a, b, c)$ ) which is called the epipolar line of  $p$ . Since  $F$  is of rank 2, it can be parametrized by (36). Thus we are dealing with the problem

$$\min \sum_{i=1}^{2n} g_i^2(f_{11}, \dots, f_{23}, h_1, h_2)$$

where

$$g_i = \sqrt{\frac{(a_i x'_i + b_i y'_i + c_i)^2}{a_i^2 + b_i^2}}, (a_i, b_i, c_i)^T = Fm_i, i = 1, \dots, n \quad (37)$$

$$g_{i+n} = \sqrt{\frac{(a'_i x_i + b'_i y_i + c'_i)^2}{a_i^2 + b_i^2}}, (a'_i, b'_i, c'_i)^T = F^T m'_i, i = 1, \dots, n. \quad (38)$$

A starting point for nonlinear optimization methods can be obtained by a linear method as follows. First we solve a linear least-squares problem while ignoring the constraint on the rank of the matrix  $F$ . Specifically, for each pair of point correspondence between two images,  $m_i$  and  $m'_i$ , we can write the equation  $m_i^T F m'_i = 0$ , which is homogeneous and linear for each entries of  $F$ . So for each point correspondence, a linear homogeneous equation

$$(u_i u'_i, v_i u'_i, u'_i u_i, v'_i u_i, v_i v'_i, v'_i v_i, u_i v_i, 1) X = 0,$$

where  $X = (f_{11}, f_{12}, f_{13}, f_{21}, \dots, f_{33})^T$ , is obtained. For all given points, a linear system  $AX = 0$  is obtained. With at least 8 points correspondences, this linear system can be solved in the least squares sense. Practically, a least squares solution  $(f_{11}^0, \dots, f_{33}^0)$  for all given point correspondences is computed by Singular Value Decomposition (SVD). A first estimation of  $h_1^0, h_2^0$  is a solution of the linear system

$$f_{11}^0 h_1^0 + f_{21}^0 h_2^0 = f_{31}^0$$

$$f_{12}^0 h_1^0 + f_{22}^0 h_2^0 = f_{32}^0$$

$$f_{13}^0 h_1^0 + f_{23}^0 h_2^0 = f_{33}^0,$$

For simplicity, we take the first two rows of  $F$  to do reparameterization. In practice, we always looked for the two most independent rows of  $F$ .

Then the point  $(f_{11}^0, \dots, f_{23}^0, h_1^0, h_2^0)$  is served as the initial values for the non linear method (e.g. the Levenber-Marquardt method and the TR-based method).

The experiments that we perform can be summarized as follows :

— The experimental data is obtained with a Pulnix 765 camera, a lens of 18 mm and FG150 Imaging technology grab board. For a fixed camera position, we put a plane calibration pattern in front about two meters of the camera, the pattern plane is then translated three times. This is equivalent to have three transparent calibration patterns spanned in space. Then we perform the same procedure by changing the camera position, the second image of the three translated calibration patterns is obtained.

The contour points are obtained by a standard gradient based edge detector. Then follows the edge linking to obtain the least squares fitted lines. The image points are computed as the intersection points of the lines, which guarantee a good precision of the locations of the image points.

— The linear method is applied to obtain a first estimation of  $F$  and a first solution

$$(-0.000001, 0.000006, -0.008166, 0.000002, 0.000003, \\ -0.003758, -2080.792969, 4254.842285)$$

which has the function value of 354.994446. Then the Levenberg-Marquardt method is applied which provides after 3 iterations the following estimation

$$\begin{pmatrix} -5.33842 \times 10^{-7} & 5.89883 \times 10^{-6} & -0.00816594 \\ 1.65618 \times 10^{-6} & 2.62407 \times 10^{-6} & -0.00375842 \\ 0.0081576 & -0.00110926 & 1.00013 \end{pmatrix}$$

with the function value of 13.8102, that is the average distance of 0.025018 for each point correspondence.

— We apply the TR-based algorithm with the same initial solution. Notice that the values of  $f_{ij}^0$  are very small compared with  $h_1^0, h_2^0$ . So to avoid the effect caused by rounded errors in calculations, we multiply  $f_{ij}^0$  by 1000 (recall that  $F$  is defined up to a scalar factor). A solution

$$\begin{pmatrix} -0.00053582 & 0.00589883 & -8.16678161 \\ 0.00165524 & 0.00262450 & -3.75921521 \\ 8.15996691 & -1.10862206 & 1000.06368051 \end{pmatrix}$$

is obtained after 4 iterations with the function value 12.90811386 that is the average distance of 0.023384 for each point correspondence.

— Subsets of 30, 40, 50, 276 points are randomly selected from the initial set of 276 points. For one of these sets noise with increasing amplitude is added to the location of each image point.

— The TR-based algorithm is applied for these sets of point correspondences ( $r_0 = 4$ ). A measure for the quality of a solution is given by :

1. the average distance between a point and its epipolar line.
2. the average distance between a point in the first image and its epipolar line.
3. the average distance between a point in the second image and its epipolar line.

over the 267 given point correspondences. The results are presented in Tables 1, 2.

— We also run the TR-based algorithm for a set of 100 point correspondences with different initial points and with suitable choices of initial radius.

First 6 parameters are taken as  $f_{11}^0 = \dots = f_{23}^0 = 1$  and  $h_1^0, h_2^0$  vary from -1,1 to -2000, 4000. The results are reported in Table 3.

Table 1. — Performance of the TR-based algorithm as a function of the number of points correspondences. CPU time is in second and was obtained on Sun/Sparc 2.

Number of pt. corres.	Performance		Average dist.		
	iter	time	total	1 image	2 image
10	14	0.20	0.077476	0.084993	0.069960
20	20	0.38	0.040241	0.043904	0.036578
30	16	0.60	0.027947	0.029940	0.025953
50	6	0.22	0.028519	0.031192	0.025847
100	14	1.42	0.027550	0.029516	0.025584
150	6	0.57	0.023839	0.025530	0.022149
200	6	1.52	0.023578	0.025259	0.021898
250	5	1.68	0.023404	0.025050	0.021758
276	4	1.48	0.023384	0.025020	0.021748

The following evaluations could be made based upon the obtained experimental results :

1. Both Levenberg-Marquardt and TR-based algorithms are well suited for the weak calibration problem, but the latter can give a more accurate solution. Moreover, the TR-based algorithm is more flexible. In fact, the parameters  $r_0, \gamma$  in ALG 4 can be appropriately chosen in order to improve the performance of the method.
2. Theoretically, non linear optimization methods converge well if the initial point is close to a global solution (e.g. a solution obtained by the linear method for the weak calibration problem). But the TR-based algorithm can provide very accurate solutions independently of the initialization as shown in Table 3. For an initial point far from the desired solution a large initial radius seems to be a good choice.
3. The method provides an accurate solution with a large number of point correspondences. Only the price to be paid is that the convergence is slower. For the considered example, an ideal choice should be 50 point correspondences.

The obtained results should be considered preliminary. However, we believe that the TR-based algorithm may be used whenever robustness and accuracy are needed. Also we believe that the method could be beneficially used to solve for other problems in computer vision, in particular the problems presented in the Section 6 of the paper.



Table 2. — Performance of the TR-based algorithm for 100 point correspondences in the presence of noise. The first column shows the maximum noise amplitude around the nominal image point locations.

Noise max. amplitude	Function value	Performance		Average dist		
		iter	time	total	1 image	2 image
0.00	5.061250	9	0.53	0.024083	0.025832	0.022333
0.01	5.067528	8	0.90	0.024038	0.025781	0.022295
0.05	5.057302	9	0.53	0.023909	0.025647	0.022171
0.10	5.486052	10	0.58	0.023965	0.025705	0.022225
0.50	13.028269	10	0.60	0.027815	0.030146	0.025484
1.00	35.876531	10	1.02	0.048204	0.053189	0.043220
1.50	72.739574	13	1.20	0.055637	0.059627	0.051646
2.00	98.155442	34	3.02	0.099394	0.111817	0.086971
3.00	254.267634	26	2.22	0.234603	0.254979	0.214228
4.00	575.133766	19	1.23	0.282393	0.296942	0.267844
5.00	701.835951	25	2.10	0.630864	0.684123	0.577606

Table 3. — Performance of the TR-based algorithm for 100 point correspondences with different initial points. The first 2 columns provide the value of  $h_1^0, h_2^0$  and  $r_0$  is chosen as in the third column. The algorithm converges to the same (optimal) solution as with the initialization by the linear method.

$h_1^0$	$h_2^0$	$r_0$	Performance	
			iter	time
-2 000	4 000	4	30	2.37
-1 000	1 000	10	71	5.73
-500	500	10	78	6.43
-100	100	10	87	6.87
-50	50	4	95	7.55
-10	10	200	70	4.63
-5	5	300	59	4.23
-1	1	300	59	4.32

## REFERENCES

- [1] J. R. CLERMONT, M. E. de LA LANDE, P. D. TAO and A. YASSINE, 1991, Analysis of plane and axisymmetric flows of incompressible fluids with the stream tube method : Numerical simulation by trust region algorithm, *Inter. J. for Numer. Method in Fluids*, **13**, pp. 371-399.
- [2] J. R. CLERMONT, M. E. de LA LANDE, P. D. TAO and A. YASSINE, 1992, Numerical simulation of axisymmetric converging using stream tube and a trust region optimization algorithm, *Engineering Optimization*, **19**, pp. 187-281.
- [3] O. FAUGERAS, 1992, What can be seen in three dimensions with an uncalibrated stereo rig ? In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pp. 563-578. Springer-Verlag, May.
- [4] O. D. FAUGERAS, 1992, *3D Computer Vision*, M.I.T. Press.
- [5] O. D. FAUGERAS, Q. T. LUONG and S. J. MAYBANK, 1992, Camera Self-Calibration : Theory and Experiments, In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pp. 321-334, Springer-Verlag, May.
- [6] O. D. FAUGERAS and G. TOSCANI, 1987, Camera calibration for 3D computer vision, In *Proceedings of International Workshop on Machine Vision and Machine Intelligence*, Tokyo, Japan.
- [7] R. FLETCHER, 1980, *Practical methods of Optimization*, John Wiley, New York.
- [8] D. M. GAY, 1981, Computing optimal constrained steps, *SIAM J. Sci. Stat. Comput.*, **2**, pp. 186-197.
- [9] G. H. GOLUB and C. VAN LOAN, 1989, *Matrix Computations*, North Oxford Academic, Oxford.
- [10] T. S. HUANG and O. D. FAUGERAS, 1989, Some properties of the E matrix in two-view motion estimation, *IEEE Transactions on PAMI*, **11**(12), pp. 1310-1312, December.
- [11] M. D. HEBDEN, 1973, An algorithm for minimization using exact second derivatives. Tech. Rep. TP515, Atomic energy research establishment (AERE), Harwell, England.
- [12] K. LEVENBERG, 1944, A method for the solution of certain nonlinear problems in least squares, *Quart. Appl. Math.*, **2**.
- [13] P. J. LAURENT, 1972, *Approximation et Optimisation*, Hermann, Paris.
- [14] D. W. MARQUARDT, 1963, An algorithm for least squares estimation of nonlinear parameters, *SIAM J. Appl. Math.*, **11**.
- [15] J. J. MORÉ, 1978, The Levenberg-Marquardt algorithm : Implementation and theory. In G. A. Waston, editor, *Lecture Notes in Mathematics 630*, pp. 105-116. Springer-Verlag, Berlin-Heidelberg-New York.
- [16] J. J. MORÉ, 1983, Recent developments in algorithm and software for trust region methods. In A. Bachem, M. Grötschel and B. Korte, editors, *Mathematical Programming, The state of the art*, pp. 258-287. Springer-Verlag, Berlin.

- [17] J. J. MORÉ and D. C. SORESEN, 1981, Computing a trust region step, *SIAM J. Sci. Statist. Comput.*, **4**, pp. 553-572.
- [18] R. MOHR L. QUAN and F. VEILLON, Relative 3D Reconstruction using multiples uncalibrated images, *The International Journal of Robotics Research*, (to appear).
- [19] R. T. ROCKAFELLAR, 1970, *Convex Analysis*, Princeton university Press, Princeton.
- [20] G. A. SHULTZ, R. B. SCHNABEL and R. H. BYRD, 1985, A family of trust region based algorithms for unconstrained minimization with strong global convergence properties, *SIAM J. on Numer. Anal.*, **22**, pp. 47-67.
- [21] T. Q. PHONG, R. HORAUD, P. D. TAO and A. YASSINE, Object Pose from 2-D to 3-D Point and Line Correspondences, *International Journal of Computer Visions* (to appear).
- [22] D. C. SORESEN, 1982, Newton's method with a model trust region modification, *SIAM J. Numer. Anal.*, **19**(2), pp. 409-426, avril.
- [23] PHAM DINH TAO, 1989, Méthodes numériques pour la minimisation d'une forme quadratique sur une boule euclidienne. Rapport de Recherche, Université Joseph Fourier, Grenoble.
- [24] PHAM DINH TAO and LE THI HOAI AN, 1993, Minimisation d'une forme quadratique sur une boule et une sphère euclidiennes. Stabilité de la dualité lagrangienne. Optimalité globale. Méthodes numériques. Rapport de Recherche, LMI, CNRS URA 1378, INSA-Rouen.
- [25] PHAM DINH TAO, LE THI HOAI AN and THAI QUYNH PHONG, Numerical methods for globally minimizing a quadratic form over euclidean ball and sphere (*submitted*).
- [25] PHAM DINH TAO, S. WANG and A. YASSINE, 1990, Training multi-layered neural network with a trust region based algorithm, *Math. Modell. Numer. Anal.*, **24**(4), pp. 523-553.