Control and Cybernetics

vol. **27** (1998) No. 3

Sub-gradient algorithms for computation of extreme eigenvalues of a real symmetric matrix

by

Adnan Yassine

Université Henri Poincaré (Nancy I), Institut Elie Cartan
UMR 9973, B.P. 239, 54506 Vandoeuvre-Les-Nancy, France
e-mail: `yassine@loria.fr` or `yassine@iecn.u-nancy.fr`

Abstract: The computation of eigenvalues of a matrix is still of importance from both theoretical and practical points of view. This is a significant problem for numerous industrial and scientific situations, notably in dynamics of structures (e.g. Géradin, 1984), physics (e.g. Rappaz, 1979), chemistry (e.g. Davidson, 1983), economy (e.g. Morishima, 1971; Neumann, 1946), mathematics (e.g. Golub, 1989; Chatelin, 1983, 1984, 1988). The study of eigenvalue problems remains a delicate task, which generally presents numerical difficulties in relation to its sensivity to roundoff errors that may lead to numerical unstabilities, particularly if the eigenvalues are not well separated. In this paper, new subgradient-algorithms for computation of extreme eigenvalues of a symmetric real matrix are presented. Those algorithms are based on stability of Lagrangian duality for non-convex optimization and on duality in the difference of convex functions. Some experimental results which prove the robustness and efficiency of our algorithms are provided.

Keywords: non-convex optimization, difference of convex functions, sub-gradient algorithms, Lagrangian duality, eigenvalue problems

1. Introduction

In recent years, active research has been conducted regarding the following class of the non-convex and non-differentiable optimization problem:

$$(PNC) : \qquad inf\,\{g(x) - h(x) : x \in X\}$$

where g and h are convex, $X = \mathbf{R}^n$.

The problem (PNC) is called the DC optimization problem and its particular structure makes significant developments in both qualitative and quantitative studies possibles (e.g. Hiriart Urruty, 1989; Hiriart Urruty and Lemaréchal, 1990; Yassine 1995, 1997)

As regards convex approaches to non-differentiable non-convex optimization (as opposed to combinatorial approaches to global optimization), we present here main results concerning DC optimization and algorithms for DC optimization problems (DCA). The DC duality was first introduced by Toland (1979) in the context of variational calculus in mechanics, and generalized by Pham Dinh (1984, 1986, 1988) for convex maximization programming.

Owing to their relative simplicity of implementation, DCA's permit to solve large-scale real world DC optimization problems. Due to their local character, they cannot guarantee the globality of computed solutions to general DC optimizations problems. In general, DCA converges to a local solution, but we observe quite often its convergence to a global one. A DC objective function has infinitely many decompositions which may exert strong influence on the quality (robustness, stability, rate of convergence and globality of solutions sought) of DCA.

The determination of eigenvalues of a matrix is still of importance from both theoretical and practical points of view. This is a significant problem for numerous industrial and scientific applications, notably in dynamics of structures (e.g. Géradin, 1984), physics (e.g. Rappaz, 1979), chemistry (e.g. Davidson, 1983), economy (e.g. Morishima, 1971; Neumann, 1946), mathematics (e.g. Golub, 1989; Chatelin, 1983, 1984, 1988). The study of eigenvalue problems remains a delicate task, which generally presents numerical difficulties in relation to its sensivity to roundoff errors that may lead to numerical unstabilities, particularly if the eigenvalues are not well separated. Up to the present time, no direct finite method for the computation of a general square matrix is available: only iterative procedures, such as LR, QR algorithms, Jacobi, Rayleigh quotient, Inverse iteration, Power method, projected Newton methods are used in the literature.

The purpose of this paper is to present new algorithms of sub-gradients, based on the stability of Lagrangian duality in non-convex optimization and on the duality in DC (Difference of two Convex functions) optimization, for the computation of extreme eigenvalues of a symmetric real matrix. In Section 2 the duality in DC optimization is presented in relation to sub-gradient algorithms. The stability of Lagrangian duality in non-convex optimization is examined in Section 3. In Sections 4 and 5 the computational procedures for the determination of extreme eigenvalues of a symmetric real matrix are presented. Section 6 presents comparative numerical experiments. Some final remarks and conclusions are given in Section 7.

2. DC Optimization

2.1. Preliminaries

Let X be the Euclidean space $\mathbf{R}^n$ and Y its dual space ($Y \equiv \mathbf{R}^n$). Denote by consider the following optimization problem:

$$(P) : \alpha = inf\,\{g(x) - h(x) : x \in X\}, \quad g, h \in \Gamma_0(X).$$

Since g and h can become infinite simultaneously, we assume that $(+\infty)$ - $(+\infty)$ $= +\infty$ to avoid an indeterminacy problem.

The DC duality may be defined by using conjugate functions g^* and h^* such that

$$(D) : \beta = inf\,\{h^*(y) - g^*(y) : y \in Y\}$$

where $g^*(y) = sup\,\{< x, y > -g(x) : x \in X\}$ is the conjugate function of $g \in \Gamma_0(X)$ with values in $\Gamma_0(Y)$. Problem (D) is the dual of (P) and $\alpha = \beta$.

If α is finite then $dom(g) \subset dom(h)$ and only the values of $(g - h) \in dom(g)$ are involved in the search for global and local solutions to (P). This DC duality was first studied by Toland (1979) in a more general framework.

2.2. Duality in DC Optimization

THEOREM 2.1 *(Pham Dinh, 1986)*

Let $(\wp)$ and (Δ) be the solutions sets of problems (P) and (D), respectively. Then:

1. $\partial h(x) \subset \partial g(x) \qquad \forall x \in (\wp)$
2. $\partial g^*(y) \subset \partial h^*(y) \quad \forall y \in (\Delta)$
3. $\cup\{\partial g^*(y) : y \in (\Delta)\} \subset (\wp)$ *(an equality if h is sub-differentiable in $(\wp)$)*
4. $\cup\{\partial h(x) : x \in (\wp)\} \subset (\Delta)$ *(an equality if g^* is sub-differentiable in (Δ))*

DEFINITION 2.1 *A point x^* of X is a local minimum of $(g - h)$ if $g(x^*)$ and $h(x^*)$ are finite and if $g(x) - h(x) \geq g(x^*) - h(x^*)$ for each x in a neighbourhood U of x^*. Consequently, $dom(g) \cap U \subset dom(h)$.*

DEFINITION 2.2 *A point x^* of X is a critical point of $(g - h)$ if $\partial h(x^*) \cap \partial g(x^*) \neq \emptyset$.*

If x^* is a local minimum of $(g - h)$, then $\partial h(x^*) \subset \partial g(x^*)$. This necessary condition is also sufficient for several non-differentiable DC problems (Pham Dinh, 1984), in particular for a polyedral h (Hiriart Urruty, 1989).

The sub-gradient algorithms presented in the following enable us to obtain a point x^* such that $\partial h(x^*) \subset \partial g(x^*)$. The local minimum property of $(g - h)$ for x^* is likely.

Let $\wp_1$ (resp. Δ_1) be the set of points verifying the necessary conditions of local optimality for (P) (resp. for (D)), i.e.

$$\wp_1 = \{x \in X : \partial h(x) \subset \partial g(x)\}, \quad \Delta_1 = \{y \in Y : \partial g^*(y) \subset \partial h^*(y)\}.$$

For every point x^* in X (resp. y^* in Y), the following problems:

$$S(x^*) = inf\,\{h^*(y) - g^*(y) : y \in \partial h(x^*)\}$$

$$T(y^*) = inf\,\{g(x) - h(x) : x \in \partial g^*(y^*)\}$$

are defined. We denote by $s(x^*)$ (resp. $\tau(y^*)$) the set of solutions to $S(x^*)$ (resp. to $T(y^*)$).

THEOREM 2.2 *(Toland, 1979; Yassine, 1995, 1997)*
$x^* \in \wp_1$ *iff* $y^* \in \Delta_1$ *s.t.* $x^* \in \partial g^*(y^*)$
$y^* \in \Delta_1$ *iff* $x^* \in \wp_1$ *s.t.* $y^* \in \partial h(x^*)$.

COROLLARY 2.1 *If* $x^* \in \wp_1$ *(resp.* $y^* \in \Delta_1$*), then:*
i) $s(x^*) = \partial h(x^*)$ *(resp.* $\tau(y^*) = \partial g^*(y^*)$*).*
ii) $h^*(y) - g^*(y) = g(x^*) - h(x^*) \quad \forall y \in \partial h(x^*)$,
(resp. $g(x) - h(x) = h^*(y^*) - g^*(y^*) \quad \forall x \in \partial g^*(y^*)$*).*

These results constitute the basis of DCA to be studied in Section 2.3. In general, DCA converges to a local solution. However, it would be interesting to formulate sufficient conditions for local optimality.

2.3. Sub-gradient algorithms (DCA Algorithms)

2.3.1. Complete form

In the sub-gradient algorithm, two sequences $\{x^k\}$ and $\{y^k\}$ verifying Theorems 1 and 2, are constructed schematically as follows: Starting from any element x^0 of X, the algorithm creates two sequences $\{x^k\}$ and $\{y^k\}$ defined by

$$y^k \in s(x^k), \quad x^{k+1} \in \tau(y^k)$$

THEOREM 2.3 *(Toland, 1979, Yassine, 1995, 1997) Let us assume that the sequences* $\{x^k\}$ *and* $\{y^k\}$ *are well-defined. Then we have:*

1. $g(x^{k+1}) - h(x^{k+1}) \le h^*(y^k) - g^*(y^k) \le g(x^k) - h(x^k)$.
   The equality $g(x^{k+1}) - h(x^{k+1}) = g(x^k) - h(x^k)$ *is fulfilled iff* $x^k \in \partial g^*(y^k)$ *and* $y^k \in \partial h(x^k)$*. Then,* $x^k \in \wp_1$ *and* $y^k \in \Delta_1$.
2. *If* α *is finite, then:*
   $\lim_{k\to+\infty} \{g(x^k) - h(x^k)\} = \lim_{k\to+\infty} \{h^*(y^k) - g^*(y^k)\} = \alpha^* \ge \alpha$.
3. *If* α *is finite and if the sequences* $\{x^k\}$ *and* $\{y^k\}$ *are bounded, then* $\forall x^* \in \Omega(x^k)$ *and (resp.* $\forall y^* \in \Omega(y^k)$*) there exists* $y^* \in \Omega(y^k)$ *(resp.* $x^* \in \Omega(x^k)$*) such that:*

   i) $x^* \in \wp_1$ *and* $g(x^*) - h(x^*) = \alpha^* \ge \alpha$

   ii) $y^* \in \Delta_1$ *and* $h^*(y^*) - g^*(y^*) = \alpha^* \ge \alpha$

   iii) $\lim_{k\to+\infty} \{g(x^k) + g^*(y^k)\} = g(x^*) + g^*(y^*) = < x^*, y^* >$

   iv) $\lim_{k\to+\infty} \{h(x^k) + h^*(y^k)\} = h(x^*) + h^*(y^*) = < x^*, y^* >$

where $\Omega(x^k)$ is the set of the accumulation points of $\{x^k\}$

From a pratical point of view, although this algorithm uses a DC decomposition mentioned above, Problems $(S(x^k))$ and $(T(x^k))$ remain DC optimization programmes. Calculation of y^k and x^{k+1} is therefore still a difficult task. In pratice, the sub-gradient algorithms are generally used on the simplified form presented in what follows.

2.3.2. Simple form

Starting from an arbitrary point x^0 in X, we define two sequences $\{x^k\}$ and $\{y^k\}$ by taking

$$y^k \in \partial h(x^k), \quad x^{k+1} \in \partial g^*(y^k)$$

In this case, all the assumptions of Theorem 2.3 are still satisfied. Moreover, one could expect to obtain the properties $\partial h(x^*) \subset \partial g(x^*)$ and $\partial g^*(y^*) \subset \partial h^*(y^*)$, but we only have $\partial h(x^*) \cap \partial g(x^*) \neq \emptyset$ and $\partial g^*(y^*) \cap \partial h^*(y^*) \neq \emptyset$.

DEFINITION 2.3 *A function f is strongly convex on X if there exists a real $\rho > 0$ (called the coercivity coefficient) such that*

$$f[\lambda x + (1-\lambda)y] \leq \lambda f(x) + (1-\lambda) f(y) - \frac{\lambda(1-\lambda)}{2}\rho\|x-y\|^2$$

$$\forall \lambda \in [0,1]; \quad \forall x, y \in X.$$

THEOREM 2.4 *(Auslender, 1976) If f is strongly convex on X, then there exists a real $\rho > 0$ such that*

$$f(x') \geq f(x) + < y, x'-x > + \rho\|x'-x\|^2 \quad \forall x, x' \in X; \quad \forall y \in \partial f(x).$$

The converse is true if f is sub-differentiable.

THEOREM 2.5 *(Yassine, 1995, 1997) Let us assume that g and h are strongly convex functions and the sequences $\{x^k\}$ and $\{y^k\}$ are well-defined. Then we get the following properties:*

1. $g(x^{k+1}) - h(x^{k+1}) \leq h^*(y^k) - g^*(y^k) - \rho_h\|x^{k+1} - x^k\|^2$
   $\leq g(x^k) - h(x^k) - (\rho_h + \rho_g)\|x^{k+1} - x^k\|^2$
   where ρ_h and ρ_g are the respective coefficients of coercivity related to h and g.
2. $h^*(y^{k+1}) - g^*(y^{k+1}) \leq g(x^{k+1}) - h(x^{k+1}) - \rho_{g^*}\|y^{k+1} - y^k\|^2$
   $\leq h^*(y^k) - g^*(y^k) - (\rho_{h^*} + \rho_{g^*})\|y^{k+1} - y^k\|^2$
   where ρ_{h^*} and ρ_{g^*} are the respective coefficients related to h^* and g^*.

COROLLARY 2.2 *(convergence of the simple form)*

1. $g(x^{k+1}) - h(x^{k+1}) = h^*(y^k) - g^*(y^k) \iff y^k \in \partial h(x^{k+1})$ *and* $x^{k+1} = x^k$.
   In this case, we get $y^k \in \partial h(x^k) \cap \partial g(x^k)$

2. *$h^*(y^k) - g^*(y^k) = g(x^k) - h(x^k) \iff x^k \in \partial g^*(y^k)$ and $y^{k-1} = y^k$. Here, we get $y^k \in \partial h(x^k) \cap \partial g(x^k)$.*
3. *If α is finite and the sequences $\{x^k\}$ and $\{y^k\}$ are bounded, then $\forall x^* \in \Omega(x^k)$ (resp. $\forall y^* \in \Omega(y^k)$) there exists $y^* \in \Omega(y^k)$ (resp. $x^* \in \Omega(x^k)$) such that:*

   i) $g(x^k) - h(x^k) = h^*(y^k) - g^*(y^k) \longrightarrow [h^*(y^*) - g^*(y^*)] = \alpha^* \geq \alpha$ *as* $k \to +\infty$

   ii) $y^* \in \partial h(x^*) \cap \partial g(x^*)$ *and* $x^* \in \partial h^*(y^*) \cap \partial g^*(y^*)$.

   iii) $\lim_{k \to +\infty} \|x^{k+1} - x^k\| = 0$ *and* $\lim_{k \to +\infty} \|y^{k+1} - y^k\| = 0$

   when $\Omega(z^k)$ is the set of accumulation points of $\{z^k\}$.

Proof. This result follows immediately from Theorems 2.4 and 2.5. ■

3. Stability of the Lagrangian duality in non-convex optimization

3.1. Problem statement

Let X be the Euclidean space $\mathbf{R}^n$ and Y be its dual space (i.e. $Y \equiv \mathbf{R}^n$). In this section, we consider the stability of the Lagrangian duality in non-convex optimization problems of the form:

$$(P) : max\,\{f(x) : \phi(x) \leq 1\}$$

where $f \in \Gamma_0(\mathbf{R}^n)$, and it is positively homogeneous, non identically zero, and ϕ is any of the norms on X.

The problem (P) (called the primary problem) may be formulated as follows:

$$(P) : min\,\{-f(x) : \frac{1}{2}\phi^2(x) \leq \frac{1}{2}\}$$

The Lagrangian function related to problem (P) is defined by

$$L(x,\lambda) = \begin{cases} -f(x) + \dfrac{\lambda}{2}(\phi^2(x) - 1) & \text{if } \lambda \geq 0 \\ -\infty & \text{otherwise.} \end{cases}$$

We define

$$(P_\lambda) : g(\lambda)$$

$$= inf\,\{L(x,\lambda) : x \in \mathbf{R}^n\} = inf\,\{-f(x) + \frac{\lambda}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}.$$

The dual problem (D) related to (P) may be written down as follows:

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

1. It provides significant additional information to characterize primary and dual solutions. Consequently, we are able to obtain the primary solution from the dual solution and vice-versa.
2. The study permits to use problem (P_λ) whose solution leads to that of (P).

3.2. Study of problem (P_λ)

We denote by $(\wp_\lambda)$ (resp. $\mathbf{P}$ and $\mathbf{D}$) the set of solutions of (P_λ) (resp. (P) and (D)).

PROPOSITION 3.1 *1.* $dom(g) =]0, +\infty[$

2. $(\wp_\lambda) \subset \{x \in \mathbf{R}^n : g(\lambda) = \dfrac{-f(x)}{2} - \dfrac{\lambda}{2} = -\dfrac{\lambda}{2}(\phi^2(x) + 1)\}$.

Proof.

1. If f is positively homogeneous and non identically zero, then
$$g(0) = inf\,\{-f(x) : x \in X\} = -\infty \tag{1}$$
Hence $dom(g) \subset]0, +\infty[$.
If f is finite, then $\forall x \in X$, $\exists b > 0$ such that $f(x) \leq b\phi(x)$. We have
$$L(x, \lambda) = -f(x) + \frac{\lambda}{2}\{\phi(x)^2 - 1\} \geq -b.\phi(x) + \frac{\lambda}{2}\{\phi(x)^2 - 1\}$$
and
$$\lim_{\phi(x)\to+\infty} L(x, \lambda) = \lim_{\phi(x)\to+\infty} \frac{\lambda}{2}\phi(x)^2 = +\infty$$
Consequently, $dom(g) =]0, +\infty[$.
2. $x \in (P_\lambda) \Rightarrow 0 \in \partial_x L(x, \lambda) \Rightarrow 0 \in -\partial f(x) + \lambda\phi(x)\partial\phi(x) \Rightarrow \partial f(x) \subset \lambda\phi(x)\partial\phi(x)$.
Then
$$\forall y \in \partial f(x), \forall z \in \partial\phi(x), \quad x^t y = \lambda\phi(x)x^t z \tag{2}$$
If f is positively homogeneous and ϕ is a norm on X, then $\forall y \in \partial f(x)$, $\forall z \in \partial\phi(x)$
$$x^t y = f(x), \qquad x^t z = \phi(x) \tag{3}$$
Combinating (2) and (3), we get
$$f(x) = \lambda\phi(x)^2 \tag{4}$$
Hence
$$g(\lambda) = -\frac{\lambda}{2} - f(x) + \frac{\lambda}{2}\phi(x)^2 = -\frac{\lambda}{2} - \frac{f(x)}{2} = -\frac{\lambda}{2}\{1 + \phi(x)^2\}$$
■

COROLLARY 3.1 *Let* $x \in (\wp_\lambda)$, *then we have*
i) $\phi(x) > 1 \Rightarrow -\lambda > g(\lambda) > -f(x)$
ii) $\phi(x) < 1 \Rightarrow -\lambda < g(\lambda) < -f(x)$
iii) $\phi(x) = 1 \Rightarrow -\lambda = g(\lambda) = -f(x)$ *and* $x \in \mathbf{P}$, $\lambda \in \mathbf{D}$.

COROLLARY 3.2 *For every $\lambda > 0$, we get the following three properties which are mutually exclusive:*
i) $(\wp_\lambda) \subset \{x \in \mathbf{R}^n : \phi(x) > 1\}$
ii) $(\wp_\lambda) \subset \{x \in \mathbf{R}^n : \phi(x) < 1\}$
iii) $(\wp_\lambda) \subset \{x \in \mathbf{R}^n : \phi(x) = 1\}$.

This corollary follows immediately from Proposition 3.1. ■

REMARK 3.1 *If $(\wp_\lambda)$ is a singleton, then the above corollary is trivial.*

THEOREM 3.1 *1. $g(\lambda) = \dfrac{-\lambda}{2} + \dfrac{k}{\lambda}$, where k is a negative constant depending on f and ϕ.*
2. $\mathbf{D} = \{\lambda^*\} = \{\sqrt{-2k}\}$ is a singleton and we get
 i) $(\wp_\lambda) \subset \{x \in \mathbf{R}^n : \phi(x) = 1\}$
 ii) $\alpha = \beta = g(\lambda^*) = -f(x^*) = -\lambda^*$
3. $\mathbf{P} = (\wp_\lambda)$.

Proof.
1. $\partial g(\lambda) \subset co\{\dfrac{\phi^2(x)-1}{2}\}$, hence $\nabla g(\lambda) = g'(\lambda) = \dfrac{\phi^2(x)-1}{2} = -1 - \dfrac{g(\lambda)}{\lambda}$, and then $g(\lambda) = \dfrac{-\lambda}{2} + \dfrac{k}{\lambda}$, where k is a real to be determined.
2. and 3.
Let λ^* and x^* be the dual and primary solutions, respectively. Therefore: $g'(\lambda^*) = \dfrac{-1}{2} - \dfrac{k}{\lambda^{*2}} = 0$, whence $k = -\dfrac{\lambda^{*2}}{2}$ and finally $\lambda^* = \sqrt{-2k}$.
Futhermore, $g'(\lambda^*) = \dfrac{\phi^2(x^*)-1}{2} = 0 \implies \phi^2(x^*) = 1 \implies f(x^*) = \lambda^*\phi^2(x^*) = \lambda^* \implies g(\lambda^*) = -\lambda^*$, and then
$$\beta = g(\lambda^*) = -f(x^*) = \alpha = -\lambda^*,\ \phi(x^*) = 1 \text{ and } \lambda^* = \sqrt{-2k}.$$
■

3.3. Stability of Lagrangian duality when f and ϕ are two semi-norms

Let us consider the primal problem (P) when f and ϕ are two semi-norms defined on $X = \mathbf{R}^n$ such that $\mathbf{N}(\phi) \subset \mathbf{N}(f)$. We want to prove that the previous results are still consistent for this class of problems. If $\mathbf{A} = \mathbf{N}(\phi)$ and $\mathbf{B} = \mathbf{A}^\perp$, then X can be expressed as $X = \mathbf{A} \oplus \mathbf{B}$ and the following properties are established :

$$\phi(x+a) = \phi(x), \qquad f(x+a) = f(x) \quad \forall a \in \mathbf{A}$$

Consequently, (P) may be expressed as

The semi-norms f and ϕ on X are norms on $\mathbf{B}$, hence problem (Q) always possesses a solution. We denote by $\mathbf{Q}$ the set of solutions to (Q), then $\mathbf{P} = \mathbf{Q} + \mathbf{A}$ (i.e. if x is a solution to (Q), then $x^* = x + a$ is a solution to (P) $\forall a \in \mathbf{A}$). In addition, we get

$$(P_\lambda) : g(\lambda) = inf\,\{-f(x) + \frac{\lambda}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}.$$

Based on the previous remarks, problem (P_λ) may be formulated as follows :

$$(P_\lambda) : g(\lambda) = inf\,\{-f(x) + \frac{\lambda}{2}\phi^2(x) : x \in \mathbf{B}\} - \frac{\lambda}{2}.$$

Since $g(\lambda) \leq -\frac{\lambda}{2}$, we can consider the set $\mathbf{E} = \{x \in \mathbf{B} : -f(x) + \frac{\lambda}{2}\phi^2(x) \leq 0\}$, which means that the last problem considered is equivalent to

$$(P_\lambda) : g(\lambda) = inf\,\{-f(x) + \frac{\lambda}{2}\phi^2(x) : x \in \mathbf{E}\} - \frac{\lambda}{2}.$$

Since f is bounded everywhere, $\exists b > 0$ such that $f(x) \leq b\,\phi(x)$. Hence (P_λ) can be written down in the form:

$$(P_\lambda) : g(\lambda) = inf\,\{-f(x) + \frac{\lambda}{2}\phi^2(x) : x \in \mathbf{E_1}\} - \frac{\lambda}{2}.$$

where $\mathbf{E_1} = \{x \in \mathbf{B} : \phi(x) \leq \frac{2b}{\lambda}, \quad \lambda > 0\}$. Since $\phi(x)$ is a norm on $\mathbf{B}$, we deduce that $\mathbf{E_1}$ is bounded and the previous stability results can be applied.

3.4. Resolution of problem (P)

The idea consists in solving the intermediate problem (P_λ) for a given λ^0, which leads to a value of the constant $k = \lambda^0\{g(\lambda^0) + \frac{\lambda^0}{2}\}$, since $\lambda^* = \sqrt{-2k}$ is a solution to (D). Problem (P_{λ^*}) is solved again to obtain a solution to (P). We then get the following algorithmic scheme:

1. Choose any $\lambda^0 > 0$.
2. Solve
   $(P_{\lambda^0}) : g(\lambda^0) = inf\,\{L(x, \lambda^0) : x \in \mathbf{R}^n\} = inf\,\{-f(x) + \frac{\lambda^0}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}$
3. Compute the constants $k = \lambda^0\{g(\lambda^0) + \frac{\lambda^0}{2}\}, \quad \lambda^* = \sqrt{-2k}$.
4. Solve
   $(P_{\lambda^*}) : inf\,\{L(x, \lambda^*) : x \in \mathbf{R}^n\} = inf\,\{-f(x) + \frac{\lambda^*}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}$
   The solution x^* to (P_{λ^*}) is also a solution to (P).

4. Extreme eigenvalues of a real symmetric matrix

4.1. Consider the following optimization problem

$$(P) : max \{f(x) = \sqrt{< Ax, x >} : \phi(x) = \|x\| \leq 1\} \qquad \textbf{(DC1)}$$

where A is a symmetric positive semi-definite matrix and $\|.\|$ is the Euclidean norm.

It is obvious that the optimal value of (P) equals the square root of the maximum eigenvalue of A. Using the sub-gradient algorithm previously presented in Section 3.4. (all the conditions are satisfied) and the intermediate problem (P_λ) solved by means of the sub-gradient algorithm, on the simple form already considered, we obtain the following expression:

$$x^{k+1} = \frac{1}{\lambda + \mu} [\mu x^k + \frac{Ax^k}{\sqrt{< Ax^k, x^k >}}] \qquad (*)$$

where λ and μ are positive reals.

Considering an arbitrary element x^0 such that $Ax^0 \neq 0$, the sequence $\{x^k\}$ defined by $(*)$ verifies the property $Ax^k \neq 0 \quad \forall k > 0$.

Indeed, the relation $Ax^1 = 0 \Rightarrow (\mu I + \dfrac{A}{\sqrt{< Ax^0, x^0 >}})Ax^0 = 0 \Rightarrow Ax^0 = 0$, which contradicts our assumption. This remark allows the algorithm related to the search of eigenvalues to be written without considering points where $f(x)$ is not differentiable. This method may be used to compute the extreme eigenvalues of any symmetric matrix. Indeed, let A be a real symmetric matrix of order n, $\lambda_1 \prec \lambda_2 \prec ... \prec \lambda_n$ its eigenvalues, and $\rho = \|A\|_1 = \|A\|_\infty = max\{\sum_{j=1}^{n} |A_{ij}| : i = 1, .., n\}$, then:

Applying the above-mentioned method to the positive semi-definite matrix $A' = (A + \rho I)$, one obtains the value $(\rho + \lambda_n)$, which gives the value of λ_n.

Since the matrix $A" = (-A + \rho I)$ is still positive semi-definite, one obtains the value $(\rho - \lambda_1)$ which allows determination of λ_1.

The description of the sub-gradient algorithm may be summarized as follows:

1. Choose any $\lambda^0 > 0$
2. Solve $(P_{\lambda^0}) : g(\lambda^0) = inf\{-f(x) + \frac{\lambda^0}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}$ by the sub-gradient algorithm with regularisation (see Section 2.3.):
   Select $x^0 \in \mathbf{R}^n$ such that $\phi(x^0) = 1, k = 0$, and construct the sequence $\{x^k\}$ as follows:
   $$x^{k+1} = \frac{1}{\lambda + \mu} [\mu x^k + \frac{Ax^k}{\sqrt{< Ax^k, x^k >}}]$$
   where λ and μ are positive reals.
   This sequence converges to a limit $x : g(\lambda^0) = -f(x) + \dfrac{\lambda^0}{2}(\phi^2(x) - 1)$
3. Compute the constant $k = \lambda^0(g(\lambda^0) + \frac{\lambda^0}{2})$, $\lambda^* = \sqrt{-2k}$

4. Solve $(P_{\lambda^*}) : inf\,\{-f(x) + \frac{\lambda^*}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}$ by the sub-gradient algorithm with regularisation presented in Section 2.3. The solution x^* of (P_{λ^*}) is a solution of (P) and the maximum eigenvalue of A is $\alpha^* = [f(x^*)]^2$.

4.2. Consider the following maximization problem

$$(P) : max\,\{f(x) = \|Ax\| : \phi(x) = \|x\| \leq 1\} \qquad \textbf{(DC2)}$$

where A is a positive semi-definite symmetric matrix and $\|.\|$ the Euclidean norm.

It is obvious that the optimal value of (P) equals the maximum eigenvalue of A. The intermediate problem (P_λ) is solved by the sub-gradient algorithm with regularisation of Section 2.3. (the simple form is used), which leads to the following formula:

$$x^{k+1} = \frac{1}{\lambda + \mu}\,[\mu x^k + \frac{A^t A x^k}{\|Ax^k\|}] \qquad (**)$$

where λ and μ are positive reals.

One chooses arbitrary x^0 such that $Ax^0 \neq 0$, then the sequence $\{x^k\}$ defined by $(**)$ verifies the property $Ax^k \neq 0 \quad \forall k > 0$. Indeed, the assumption $Ax^1 = 0 \Rightarrow (\mu I + \frac{AA^t}{\|Ax^0\|})Ax^0 = 0 \Rightarrow Ax^0 = 0$, which contradicts our assumption. This remark allows to write the algorithm used for the search of eigenvalues without considering points where f is not differentiable.

According to Section 4.1., this method may be used to compute extreme eigenvalues of any symmetric matrix.

The elementary steps of the sub-gradient algorithm can be summarized as follows:

1. Select any $\lambda^0 > 0$
2. Solve $(P_{\lambda^0}) : g(\lambda^0) = inf\,\{-f(x) + \frac{\lambda^0}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}$ by the sub-gradient algorithm with regularisation presented in Section 2.3. as:
   We select any $x^0 \in \mathbf{R}^n$ such that $\phi(x^0) = 1, k = 0$,and construct the sequence $\{x^k\}$ as follows:
   $$x^{k+1} = \frac{1}{\lambda + \mu}\,[\mu x^k + \frac{A^t A x^k}{\|Ax^k\|}]$$
   where λ and μ are positive reals.
   This sequence converges towards a limit $x : g(\lambda^0) = -f(x) + \frac{\lambda^0}{2}(\phi^2(x) - 1)$
3. Compute the constant $k = \lambda^0(g(\lambda^0) + \frac{\lambda^0}{2}), \quad \lambda^* = \sqrt{-2k}$.
4. Solve $(P_{\lambda^*}) : inf\{-f(x) + \frac{\lambda^*}{2}(\phi^2(x) - 1) : x \in \mathbf{R}^n\}$ by the sub-gradient algorithm with regularisation presented in Section 2.3. The solution x^* of (P_{λ^*}) is a solution of (P) and the maximum eigenvalue of A is $\alpha^* = f(x^*)$

The two DC algorithms presented above are applied to the Lagrangian function. We now consider DC algorithms directly related to the initial problem (P):

4.3. Let us consider the following optimization problem:

$$(P) : max\, \{f(x) = \sqrt{< Ax, x >} : \phi(x) = \|x\| \leq 1\}$$

where A denotes a positive semi-definite symmetric matrix and $\|.\|$ is the Euclidean norm.

It can be easly verified that (P) is equivalent to the following problem (Q):

$$(Q) : min\, \{-\sqrt{< Ax, x >} + \chi_E(x) : x \in \mathbf{R}^n\}$$

where $\chi_E(x)$ stands for the indicatrix function related to the set $E = \{x \in \mathbf{R}^n : \|x\| \leq 1\}$.

The problem (Q) may be written on the DC form:

$$(Q) : min\, \{[\frac{\mu}{2}\|x\|^2 + \chi_E(x)] - [\sqrt{< Ax, x >} + \frac{\mu}{2}\|x\|^2] : x \in \mathbf{R}^n\}$$

where μ is an arbitrary positive real.

Application of the simple form of the sub-gradient algorithm given in Section 2.3. to solve (Q) leads to the following formula:

$$(\mathbf{Proj1}) : \qquad x^{k+1} = Proj_E(y^k) = \begin{cases} y^k & \text{if } \|y^k\| \preceq 1 \\ \dfrac{y^k}{\|y^k\|} & \text{otherwise} \end{cases}$$

where $y^k = [\mu x^k + \dfrac{Ax^k}{\sqrt{< Ax^k, x^k >}}]$

The sequence $\{x^k\}$ converges to x^* solution of (P) and the maximum eigenvalue of A is $\alpha^* = [f(x^*)]^2$.

4.4. Consider now the optimization problem:

$$(P) : max\, \{f(x) = \|Ax\| : \phi(x) = \|x\| \leq 1\}$$

Problem (P) is equivalent to problem (Q):

$$(Q) : min\, \{-A\|x\| + \chi_E(x) : x \in \mathbf{R}^n\}$$

which can be written in the following DC form:

$$(Q) : min\{[\frac{\mu}{2}\|x\|^2 + \chi_E(x)] - [\|Ax\| + \frac{\mu}{2}\|x\|^2] : x \in \mathbf{R}^n\}$$

where μ is an arbitrary positive real.

The simple form of the sub-gradient algorithm (see Section 2.3.) used to solve problem (Q) leads to the following formula:

$$(\mathbf{Proj2}): \qquad x^{k+1} = Proj_E\,[\mu x^k + \frac{A^t A x^k}{\|Ax^k\|}]$$

The sequence $\{x^k\}$ converges to x^* solution of (P) and the maximum eigenvalue of A is $\alpha^* = f(x^*)$.

4.5. The optimization problem:

$$(P) : max\,\{f(x) = \frac{1}{2} < Ax, x >: \phi(x) = \|x\| \leq 1\}$$

(P) is equivalent to the following problem (Q):

$$(Q) : min\,\{[\frac{\mu}{2}\|x\|^2 + \chi_E(x)] - [\frac{1}{2} < Ax, x > + \frac{\mu}{2}\|x\|^2] : x \in \mathbf{R}^n\}$$

where μ is a given positive real number.

To solve (Q), we apply the simple form of the sub-gradient algorithm and obtain the following relation:

$$(\mathbf{Proj3}): \qquad x^{k+1} = Proj_E\,[(A + \mu I)x^k]$$

The sequence $\{x^k\}$ converges to x^* solution of (P) and the maximum eigenvalue of A is $\alpha^* = \|Ax^*\|$.

4.6. The optimization problem:

$$(P) : max\,\{f(x) = \frac{1}{2}\|Ax\|^2 : \phi(x) = \|x\| \leq 1\}$$

(P) is equivalent to problem (Q):

$$(Q) : min\,\{[\frac{\mu}{2}\|x\|^2 + \chi_E(x)] - [\frac{1}{2}\|Ax\|^2 + \frac{\mu}{2}\|x\|^2] : x \in \mathbf{R}^n\}$$

where μ is a positive real.

The simple form of the sub-gradient algorithm applied to (Q) leads to the following expression:

$$(\mathbf{Proj4}): \qquad x^{k+1} = Proj_E\,[(A^t A + \mu I)x^k]$$

The sequence $\{x^k\}$ converges to x^* solution of (P) and the maximum eigenvalue of A is $\alpha^* = \|Ax^*\|$.

5. Linpack technique for computing the smallest eigenvalue

Let A be a real symmetric matrix, λ_1 its smallest eigenvalue and λ a real positive number strictly superior to $-\lambda_1$ (in general, $\lambda = \|A\|_1$). Using Choleski method, we write $(A + \lambda I) = R^t R$, where I denotes the unit matrix of order n and R an upper triangular matrix. We wish to estimate the vector associated to the smallest eigenvalue of A, which corresponds to evaluating the vector z^* such that:

$$\|Rz^*\| = min\,\{\|Rx\| \,:\, \|x\| = 1\}$$

We find it not reasonable to solve directly this minimization problem to calculate $z*$. We compute an approximation of z^* by using the so-called Linpack technique proposed by Cline & al (1979). This method consists in determining the vector w by solving the linear system $R^t w = e$ with $e = (\pm 1, \pm 1, ..., \pm 1)^t$.

The sign of each component of vector e is choosen in such a way that the norm of w is large enough. Several approaches leading to determine the vector e in these conditions have been proposed in the literature (Moré & Sorenson, 1983; Cline & al, 1979). After calculating w, the equation $Rw = w$ is solved, leading to $z^* = \frac{v}{\|v\|}$. The Linpack technique may be summarized as follows:

1. Choose a real $\lambda > -\lambda_1$.
2. Decompose $(A + \lambda I) = R^t R$.
3. Determine vector e and solve $R^t w = e$.
4. Solve $Rv = w$ and $z^* = \frac{v}{\|v\|}$.

In order to compute the approximate maximum eigenvalue of A, we apply the Linpack technique on $-A$.

6. Numerical results for some examples

In this section, we present comparative numerical results related to the algorithms previously defined for computation of the maximum eigenvalue (of a symmetric matrix).

(1): A is a full, symmetric, positive semi-definite, "Hilbert" matrix:

$$A_{ij} = \frac{1}{i + j - 1} \qquad \forall i = 1, ..., n$$

(2): A is a hollow tridiagonal symmetric matrix

$$A_{ii} = 2 \quad \forall i = 1, .., n; \quad A_{ii+1} = -1 \quad \forall i = 1, .., n-1;$$

$$A_{i-1i} = -1 \quad \forall i = 2, .., n \; and \; A_{ij} = 0 \quad \text{otherwise.}$$

Cases (3), (4) and (5): A is a full, positive semi-definite symmetric

The computations were run on a SUN work station (SPARC station SLC). Information on the CPU time related to different techniques used are given, together with the matrix size, in tables 1 to 5:

DC1, DC2, Proj1, Proj2, Proj3 and Proj4 denote the different DC sub-gradient algorithms used.

Power is the iterative Power method (Chatelin, 1988) and Liniter corresponds to calculation of the approximate value of α^* by the Linpack technique (Moré & Sorenson, 1983; Cline & al, 1979) followed by the use of the iterative Power method.

Size	DC1	DC2	Proj1	Proj2	Proj3	Proj4	Power	Liniter
50	0.2	0.2	0.3	0.3	0.3	0.3	0.4	0.4
100	0.5	0.6	1	1	1	1	1	1.5
150	1	1.2	1.8	2	1.8	1.8	2	4.5
200	2	2.3	4	4	3.5	3.5	6	8.5
250	2.5	3	5	6	5	5	9	14
300	3.5	4.5	7	7.5	6	5.5	14	20
350	4.5	6	9	10	8.5	7.5	20	28
400	6	8	13	13	11	10	29	37
450	8	10	15	17	14	12	44	49
500	10	13	20	21	17	16	58	62
1000	48	73	105	117	95	91	256	274

Table 1. CPU time (in seconds) corresponding to different methods used for the Hilbert matrix (case 1)

7. Accuracy of the method

In Tables 6 to 10, the computed values of the maximum eigenvalue α^* related to the different algorithms used in our calculations are presented. The last column concerns the exact value of α^*. It may be noted that Proj1 and Proj3 (resp. Proj2 and Proj4) lead to the same approximate values of α^* as DC1 (resp. DC2).

Size	DC1	DC2	Proj1	Proj2	Proj3	Proj4	Power	Liniter
50	3	6	3	6	3	6	6	12
100	12	24	13	26	13	26	36	50
150	28	55	31	60	29	61	82	114
200	50	98	55	106	54	106	145	203
250	77	153	86	166	85	152	228	318
300	111	221	122	240	119	232	329	455
350	156	301	166	325	165	316	443	618
400	203	393	218	425	214	412	585	806
450	258	496	278	546	268	523	741	1020
500	316	613	342	672	335	665	940	1285
1000	1364	2687	1468	3094	1428	2860	4230	5479

Table 2. CPU time (in seconds) related to different methods used for case 2

Size	DC1	DC2	Proj1	Proj2	Proj3	Proj4	Power	Liniter
50	4	6	5	7	6	6	5	5
100	12	18	13	19	13	19	9	19
150	23	35	30	36	29	36	47	75
200	54	105	54	106	52	105	148	185
250	84	156	84	150	81	150	185	254
300	121	236	121	236	116	236	305	393
350	156	305	157	305	157	307	448	580
400	204	398	204	398	204	400	480	614
450	258	505	259	505	258	505	748	925
500	318	623	320	625	318	622	924	1232
1000	1382	2714	1408	2875	1324	2884	3818	4785

Table 3. CPU time (in seconds) for a random positive semi-definite symmetric matrix (case 3)

Size	DC1	DC2	Proj1	Proj2	Proj3	Proj4	Power	Liniter
50	4	6	4	6	4	6	8	9
100	10	16	13	23	13	21	7	13
150	29	57	30	57	30	57	17	32
200	51	102	51	101	53	100	75	102
250	79	160	80	157	85	158	115	172
300	115	231	118	226	127	236	173	245
350	156	318	160	310	167	311	248	356
400	204	412	210	401	217	401	372	468
450	260	510	267	510	274	509	446	584
500	318	630	324	626	336	628	612	708
1000	1384	2820	1406	2840	1434	2812	2450	2872

Table 4. CPU time (in seconds) for case 4.

Size	DC1	DC2	Proj1	Proj2	Proj3	Proj4	Power	Liniter
50	3	6	3	6	3	6	9	12
100	14	27	14	27	14	27	37	58
150	31	60	31	61	30	60	82	115
200	42	65	55	81	53	81	150	179
250	86	170	88	175	85	171	241	298
300	124	252	130	252	123	251	360	457
350	163	270	177	335	169	334	445	609
400	219	440	228	457	221	446	604	816
450	279	540	296	556	281	554	789	1065
500	346	668	361	696	346	693	957	1268
1000	1392	2868	1565	3212	1476	3103	3861	4453

Table 5. CPU time (in seconds) for case 5.

Size	DC1	DC2	Power	Liniter	α^*
50	2.0762967	2.0762967	2.0762967	2.0762967	2.0762967
100	2.1826961	2.1826961	2.1826961	2.1826961	2.1826961
150	2.2378812	2.2378812	2.2378812	2.2378812	2.2378812
200	2.2742670	2.2742670	2.2742670	2.2742670	2.2742670
250	2.3010352	2.3010352	2.3010352	2.3010352	2.3010352
300	2.3220199	2.3220199	2.3220199	2.3220199	2.3220199
350	2.3391705	2.3391705	2.3391705	2.3391705	2.3391705
400	2.3536064	2.3536064	2.3536064	2.3536064	2.3536064
450	2.3660270	2.3660270	2.3660270	2.3660270	2.3660270
500	2.3768965	2.3768965	2.3768965	2.3768965	2.3768965
1000	2.4258645	2.4258645	2.4258585	2.4258645	2.4258645

Table 6. Computed values of α^* for the algorithms used in our calculations. The last column concerns the exact value of α^* (case 1).

Size	DC1	DC2	Power	Liniter	α^*
50	3.9958160	3.9962050	3.9848430	3.9962067	3.9962067
100	3.9963850	3.9983211	3.9945270	3.9991150	3.9990326
150	3.9964000	3.9983267	3.9945670	3.9997980	3.9995672
200	3.9964840	3.9983267	3.9945670	3.9999710	3.9997557
250	3.9964840	3.9983267	3.9945670	3.9999990	3.9998433
300	3.9964840	3.9983267	3.9945670	3.9999995	3.9998911
350	3.9964840	3.9983270	3.9945670	3.9999999	3.9999199
400	3.9964840	3.9983270	3.9945670	4.0000000	3.9999386
450	3.9964840	3.9983270	3.9945670	4.0000000	3.9999515
500	3.9964840	3.9983270	3.9945670	4.0000000	3.9999607
1000	3.9996800	3.9997460	3.9964980	4.0000000	3.9999902

Table 7. Computed values of α^* for the algorithms used in our calculations. The last column concerns the exact value of α^* (case 2).

Size	DC1	DC2	Power	Liniter	α^*
50	24.500000	24.500000	24.500000	24.500000	24.500
100	49.000007	49.000007	49.000005	49.000007	49.000
150	74.500004	75.500004	74.500016	74.500009	74.500
200	99.499985	99.500008	99.500349	99.500010	99.500
250	124.50002	124.50002	124.50015	124.50002	124.50
300	149.49994	149.50001	149.49997	149.50002	149.50
350	174.49993	174.49998	174.50177	174.50009	174.50
400	199.49992	199.49998	199.50642	199.49999	199.50
450	222.99290	222.99994	222.99940	223.00025	223.00
500	249.49978	249.49986	249.50048	249.50005	249.50
1000	499.99870	499.99890	500.00568	500.00481	500.00

Table 8. Computed values of α^* for the algorithms used in our calculations. The last column concerns the exact value of α^* (case 3).

Size	DC1	DC2	Power	Liniter	α^*
50	10.979992	10.979983	10.980057	10.980008	10.98
100	11.900003	11.900003	11.900450	11.900003	11.90
150	12.959972	12.959936	12.960091	12.960016	12.96
200	13.979965	13.979950	13.979931	13.979996	13.98
250	14.979750	14.979998	14.980236	14.980005	14.98
300	15.978305	15.979891	15.979961	15.980051	15.98
350	16.959991	16.959994	16.960505	16.959994	16.96
400	17.977001	17.979740	17.980043	17.979900	17.98
450	18.977501	18.979830	18.978129	18.980017	18.98
500	19.959985	19.959987	19.959858	19.959959	19.96
1000	24.979873	24.979920	24.976581	24.979850	24.98

Table 9. Computed values of α^* for the algorithms used in our calculations. The last column concerns the exact value of α^* (case 4).

Size	DC1	DC2	Power	Liniter	α^*
50	12.449974	12.450153	12.450053	12.450001	12.45
100	14.949786	14.949998	14.950497	14.949999	14.95
150	17.449664	17.450004	17.450358	17.450008	17.45
200	19.900002	19.900002	19.900071	19.900003	19.90
250	22.449171	22.449996	22.450415	22.450008	22.45
300	24.898710	24.899984	24.900003	24.899994	24.90
350	27.399991	27.399991	27.401263	27.399999	27.40
400	29.949990	29.950005	29.951348	29.950001	29.95
450	32.299007	32.299888	32.299760	32.300006	32.30
500	34.899620	34.900004	34.899611	34.900007	34.90
1000	52.499540	52.499620	52.499580	52.500042	52.50

Table 10. Computed values of α^* for the algorithms used in our calculations. The last column concerns the exact value of α^* (case 5).

For the matrix of size 500, the graphs of the function $g(\lambda) = \frac{-\lambda}{2} + \frac{k}{\lambda}$, are plotted in Figs. 1 through 5.

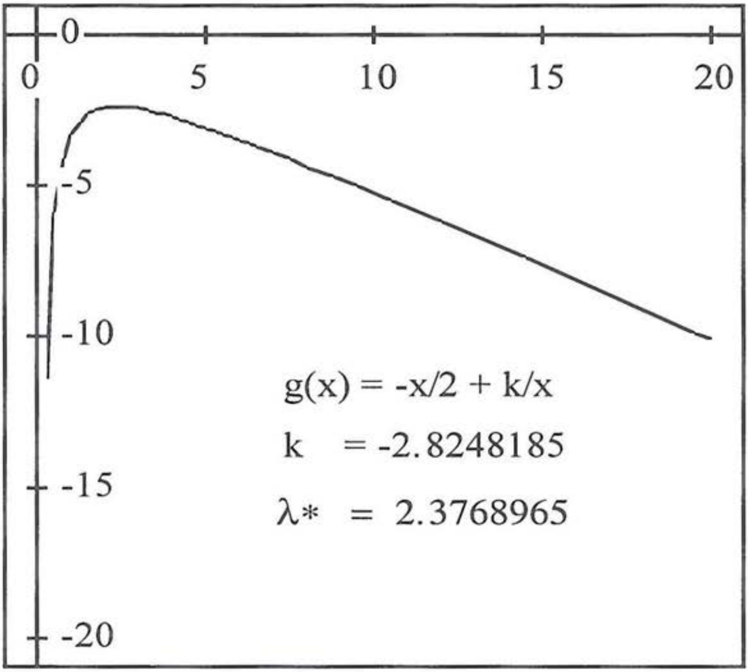

Figure 1. Graph of the function $g(\lambda) = -\frac{\lambda}{2} - \frac{2.8248185}{\lambda}$ $\lambda^* = 2.3768965$ (case 1)

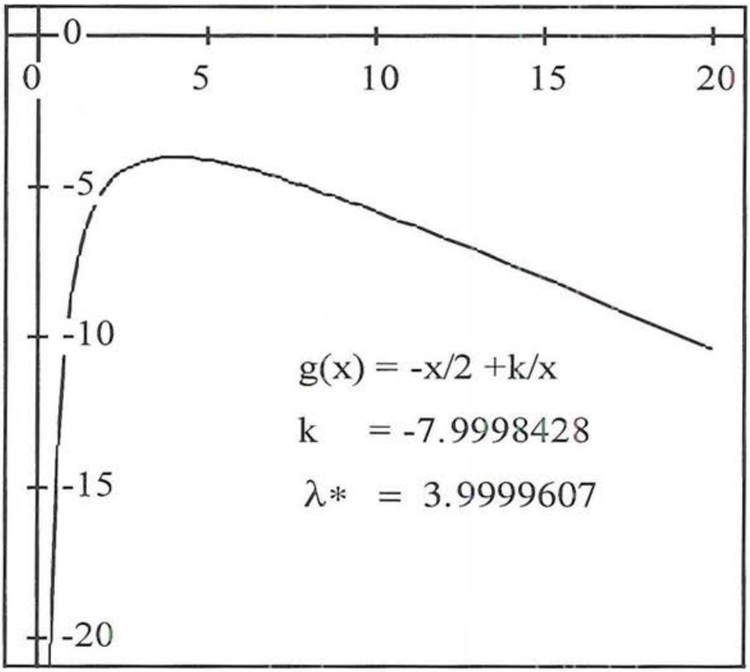

Figure 2. Graph of the function $g(\lambda) = -\frac{\lambda}{2} - \frac{7.9998428}{\lambda}$ $\lambda^* = 3.9999607$ (case 2)

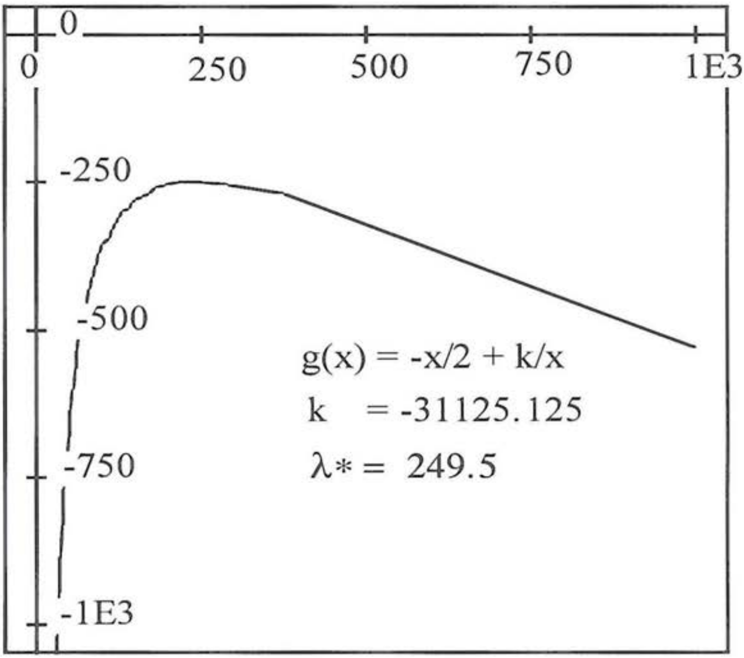

Figure 3. Graph of the function $g(\lambda) = -\dfrac{\lambda}{2} - \dfrac{31123.125}{\lambda}$ $\lambda^* = 249.5$ (case 3)

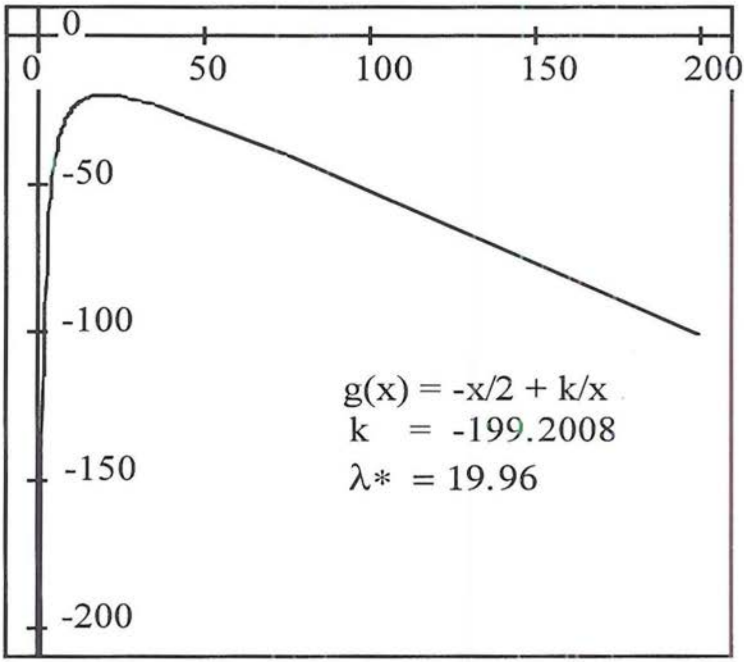

Figure 4. Graph of the function $g(\lambda) = -\dfrac{\lambda}{2} - \dfrac{199.2008}{\lambda}$ $\lambda^* = 19.96$ (case 4)

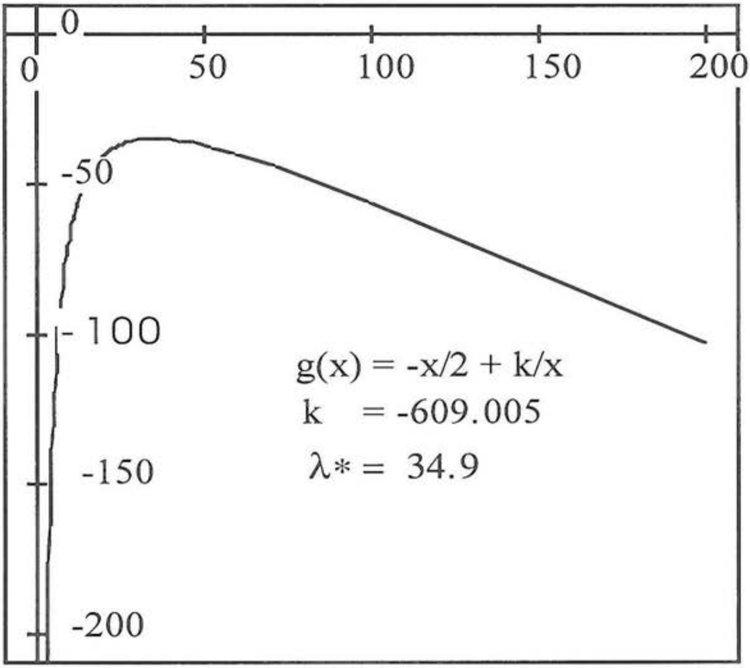

Figure 5. Graph of the function $g(\lambda) = -\frac{\lambda}{2} - \frac{609.005}{\lambda}$ $\lambda^* = 34.90$ (case 5)

8. Discussion and concluding remarks

According to the results, the following remarks can be made:

1. The numerical results obtained for all the examples reported previously confirm the stability, the robustness and the superiority of the sub-gradient algorithms (particularly DC1), when compared to other classical methods.
2. In sub-gradient algorithms DC1 and DC2, the choice of regularisation parameters λ and μ is delicate and significant. Indeed, changes of values of λ and μ do not affect the value of the computed solution, but influence significantly the performance of the algorithm in time. The different numerical tests performed led us to conclude that the best values of λ and μ are those in the 1-10 range. For rather small (resp. rather large) values of λ and μ, the convergence becomes slow. It should also be pointed out that, in some cases (for example the case $n^\circ 2$), the algorithm remains insensitive to variations of λ and μ : the results remain the same for any positive value of λ and μ.
3. For the other sub-gradient algorithms (Proj1, Proj2, Proj3 and Proj4), the best values of μ are found be in the 0-1 range.
4. The number of iterations before convergence was not reported for the algorithms used in our calculations because the complexity of the iterative procedure differs from one method to another. In the author's opinion, the significant features concern performance in time and accuracy of the method under consideration, which corresponds to the global CPU time and the computed approximate value of α^*.
5. The use of the Linpack technique for the iterate power method leads to increase of the global CPU time but improves the accuracy.
6. When using the projected Newton method and the Rayleigh quotient algorithm, it cannot be defined, a priori, what is the best-adapted norm for calculation of the extreme eigenvalues.
7. Results concerning the use of the projected Newton and the Rayleigh quotient algorithms (involving or not the Linpack technique) for the calculation of the maximum eigenvalue are not presented in the paper since these methods generally lead to determination of an eigenvalue which is not necessarily the maximum value. Three examples related to this problem are presented in the Appendix.

References

AUSLENDER, A. (1976) *Optimisation, méthodes numériques.* Masson, Paris.

CHATELIN, F. (1983) *Spectral Approximation of Linear Operators.* Academic Press, New Work.

CHATELIN, F. (1984) Simultaneous Newton's iterations for the eigenproblem. *Computing, Suppl.*, **5**, 67-74.

CHATELIN, F. (1988) *Valeurs propres d'une matrice.* Masson, Paris.

CLINE, A., MOLER, C., STEWART, G., WILKINSON, J.H. (1979) An estimate for the condition number of matrix. *SIAM. J. Numer. Analysis*, **16**, 368-375.

DAVIDSON, E.R. (1983) Matrix eigenvector methods. *Methods in Computational Molecular Physics*, 95-113, Reidel, Boston.

GÉRADIN, M., KILL, N. (1984) Analyse dynamique des machines tournantes. Principes et applications. Rapport VF-50, labo. Aéronaut. Université de Liège, Belgique.

GOLUB, G.H., VAN LOAN, C. (1989) *Matrix Computations*. North Oxford Academic, Oxford.

HIRIART URRUTY, J.B. (1989) How to regularize a difference of two functions?. Seminar of numerical analysis. Université Paul Sabatier. Toulouse.

HIRIART URRUTY, J.B & LEMARÉCHAL, C. (1990) Testing necessary and sufficient conditions for global optimality in the problem of maximizing a convex quadratic over a convex polyhedron. Seminar of numerical analysis. Université Paul Sabatier. Toulouse.

MORÉ, J.J., SORENSEN, D.C. (1983) Computing a trust region step. *SIAM J. Sc. Stat. Comput*, **4**, 553-572.

MORISHIMA, M. (1971) *Marx's Economics. A dual theory of value and growth.* Cambridge University Press.

NEUMANN, J. VON (1946) A model of general economic equilibrium. *Rev. Econ. Stud.*, **13**, 10-18.

PETERS, G., WILKINSON, J.H. (1979) Inverse iteration, ill-conditioned equations and Newton's method. *SIAM Review*, **21**, 339-360.

PHAM DINH, T. (1984) Convergence of subgradient method for computing the bound norm of matrices. *Linear Alg. and Its Appl.*, **62**, 163-182.

PHAM DINH, T. (1986) Algorithms for solving a class of non convex optimization problems. Methods of subgradients. Fermat days 85. *Mathematics for Optimization*, Hiriart Urruty J.B. (ed.), Elsevier Science Publishers B.V. North-Holland.

PHAM DINH, T. & EL BERNOUSSI, S. (1988) Duality in difference of convex functions. Subgradient methods. *Trends in Mathematical Optimization, International series of Numer. Math.*, **84**, Birkhauser, 277-293.

RAPPAZ, J. (1979) *Spectral approximation by finite elements and applications.* Academic Press, New York, 311-318.

ROCKAFELLAR, R.T. (1970) *Convex Analysis.* Princeton University. Princeton.

TOLAND, J.F. (1979) A duality principle for non-convex optimization and calculus of variations. *Arch. Rational. Mech. Analysis.* **71**.

YASSINE, A., PHAM DINH, T. (1992) Algorithmes de sous-gradient pour la résolution du problème d'analyse multidimensionnelle des tableaux de dissimilarité. Rapport de Recherche 30, Les Prépublications de l'Institut Elie Cartan, Université de Nancy I, France.

YASSINE, A. (1995) Algorithmes de sous-gradient pour calculer les valeurs propres extrêmes d'une matrice symétrique réelle. Rapport de Recherche 5, Les Prépublications de l'Institut Elie Cartan, Université de Nancy I, France.

YASSINE, A. (1997) Sub-gradient Algorithms for solving Multi-Dimensional Analysis Problems of Dissimilarity Data. *Journal of Applied Mathematics and Computer Science*, **7**, 3, 421-443.

Appendix

On the use of projected Newton and Rayleigh quotient algorithms for the search of the maximal eigenvalue of a matrix A.

EXAMPLE 8.1 *(Golub and Van Loan, 1989)*

$$A = \begin{bmatrix} 100 & 1 & 1 & 1 \\ 1 & 99 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Let $\sigma(A)$ be the spectrum of the matrix A. Hence:

$$\sigma(A) = \{0.37982076, 2.579377773, 98.38412988, 100.65667158\}$$

Applying the above-mentioned methods, we obtain the eigenvalue $\lambda_2 = 2.579377773$, starting from the initial point $x^{(0)} = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})^t$.

EXAMPLE 8.2 *(Golub and Van Loan, 1989)*

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 6 & 10 & 15 & 21 \\ 1 & 4 & 10 & 20 & 35 & 56 \\ 1 & 5 & 15 & 35 & 70 & 126 \\ 1 & 6 & 21 & 56 & 126 & 252 \end{bmatrix}$$

$$\sigma(A) = \{0.00300439, 0.06429432, 0.48933883, 2.04357378, 15.55347327,$$

$$332.84631541\}$$

Starting from $x^{(0)} = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})^t$, the Rayleigh quotient algorithm converges to $\lambda_5 = 15.55347327$ and the projected Newton method converges to $\lambda_4 = 2.04357378$.

EXAMPLE 8.3 *(Chatelin, 1988)*

Let us consider the n-symmetric tridiagonal matrix A given by:

and $A_{ij} = 0$ *otherwise, for* $n = 5$:

$$\sigma(A) = \{0.26794919,\ 1,\ 2,\ 3,\ 3.73205081\}$$

The methods considered here lead to the eigenvalue $\lambda_1 = 0.26794919$ *when using the Euclidean norm and* $\lambda_2 = 1$ *for the infinite-norm.*

for $n = 10$:

$$\sigma(A) = \{0.08101405, 0.31749293, 0.69027853, 1.16916997,$$

$$1.71537032, ..., 3.91898595\}$$

The methods lead to eigenvalue $\lambda_2 = 0.31749293$ when using the Euclidean norm and, for the infinite norm, the Rayleigh quotient algorithm converges to $\lambda_4 = 1.16916997$ and the projected Newton method converges to $\lambda_5 = 1.71537032$.