Top (2012) 20:791–809
DOI 10.1007/s11750-010-0161-9

ORIGINAL PAPER

An investigation of feasible descent algorithms for estimating the condition number of a matrix

Carmo P. Brás · William W. Hager · Joaquim J. Júdice

Received: 23 October 2009 / Accepted: 12 October 2010 / Published online: 4 November 2010

Abstract Techniques for estimating the condition number of a nonsingular matrix are developed. It is shown that Hager's 1-norm condition number estimator is equivalent to the conditional gradient algorithm applied to the problem of maximizing the 1-norm of a matrix-vector product over the unit sphere in the 1-norm. By changing the constraint in this optimization problem from the unit sphere to the unit simplex, a new formulation is obtained which is the basis for both conditional gradient and projected gradient algorithms. In the test problems, the spectral projected gradient algorithm yields condition number estimates at least as good as those obtained by the previous approach. Moreover, in some cases, the spectral gradient projection algorithm, with a careful choice of the parameters, yields improved condition number estimates.

Keywords Condition number · Numerical linear algebra · Nonlinear programming · Gradient algorithms

Mathematics Subject Classification (2000) 15A12 · 49M37 · 90C30

Research of W.W. Hager is partly supported by National Science Foundation Grant 0620286.

C.P. Brás (✉)
CMA, Departamento de Matemática, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
e-mail: mb@fct.unl.pt

W.W. Hager
Department of Mathematics, University of Florida, Gainesville, FL 32611-8105, USA
e-mail: hager@math.ufl.edu

J.J. Júdice
Departamento de Matemática, Universidade de Coimbra and Instituto de Telecomunicações, 3001-454 Coimbra, Portugal
e-mail: joaquim.judice@co.it.pt

1 Introduction

The role of condition number is well established in a variety of numerical analysis problems (see, for example Demmel 1997; Golub and Loan 1996; Hager 1998). In fact, the condition number is useful for assessing the complexity of a linear system and the error in the solution since it provides information about the sensitivity of the solution to perturbations in the data. For large matrices, the calculation of the exact value is impracticable as it involves a large number of operations. Therefore, techniques have been developed to approximate $\|A^{-1}\|_1$ without computing the inverse of the matrix A. Hager's 1-norm condition number estimator (Hager 1984) is implemented in the MATLAB builtin function CONDEST. This estimate is a gradient method for computing a stationary point of the problem

$$\begin{aligned} &\text{Maximize } \|Ax\|_1 \\ &\text{subject to } \ \|x\|_1 = 1. \end{aligned} \tag{1}$$

The algorithm starts from $x = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$, evaluates a subgradient, and then moves to a vertex $\pm e_j$, $j = 1, 2, \dots, n$, which results in the largest increase in the objective function based on the subgradient. The iteration is repeated until a stationary point is reached, often within two or three iterations. We show that this algorithm is essentially the conditional gradient (CG) algorithm.

Since the solution to (1) is achieved at a vertex of the feasible set, an equivalent formulation is

$$\begin{aligned} &\text{Maximize } \|Ax\|_1 \\ &\text{subject to } \ e^T x = 1, \\ &\qquad\qquad\quad x \geq 0 \end{aligned} \tag{2}$$

where e is the vector with every element equal to 1. We propose conditional gradient and gradient projected algorithms for computing a stationary point of (2).

The paper is organized as follows. In Sect. 2 Hager's algorithm is described and its equivalence with a CG algorithm is shown. The Simplex formulation of the condition number problem and a CG algorithm for solving this nonlinear program are introduced in Sect. 3. The SPG algorithm is discussed in Sect. 4. Computational experience is reported in Sect. 5 and some conclusions are drawn in the last section.

2 Hager's algorithm for condition number estimation

Definition 1 The 1-norm of a square matrix $A \in \mathbb{R}^{n \times n}$ is defined by

$$\|A\|_1 = \max_{x \in \mathbb{R}^n \setminus \{0\}} \frac{\|Ax\|_1}{\|x\|_1} \tag{3}$$

where $\|x\|_1 = \sum_{j=1}^n |x_j|$.

In practice the computation of the 1-norm of a matrix A is not difficult since

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|.$$

In this paper we are interested in the computation of the 1-norm of the inverse of a nonsingular matrix A. This norm is important for computing the condition number of a matrix A, a concept that is quite relevant in numerical linear algebra (Gill et al. 1991; Higham 1996). We recall the following definition.

Definition 2 The condition number for the 1-norm of a nonsingular matrix A is given by

$$\mathrm{cond}_1(A) = \|A\|_1 \left\|A^{-1}\right\|_1$$

where A^{-1} is the inverse of matrix A.

In practice, computing the inverse of a matrix involves too much effort, particularly when the order is large. Instead, the condition number of a nonsingular matrix is estimated by

$$\mathrm{cond}_1(A) = \|A\|_1 \beta$$

where β is an estimate of $\|A^{-1}\|_1$. To find a β that is close to the true value of $\|A^{-1}\|_1$, the following optimization problem is considered in (Hager 1984):

$$\begin{aligned} &\text{Maximize } F(x) = \|Ax\|_1 = \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \\ &\text{subject to } \ x \in K \end{aligned} \tag{4}$$

where

$$K = \left\{ x \in \mathbb{R}^n : \|x\|_1 \le 1 \right\}.$$

The following properties hold (Higham 1988).

(i) F is differentiable at all $x \in K$ such that $(Ax)_i \neq 0, \ \forall i$.
(ii) If

$$\left\|\nabla F(x)\right\|_\infty \le \nabla F(x)^T x, \tag{5}$$

then x is a stationary point of F on K.
(iii) F is convex on K and the global maximum of F is attained at one of the vertices e^j of the convex set K.

The optimal solution of the program (4) is not in general unique and there may exist an optimal solution which is not one of the unit vectors e^i. In fact, the following result holds.

Theorem 1 *Let J be a subset of $\{1, \dots, n\}$ such that $\|A\|_1 = \|A_{.j}\|_1$, for all $j \in J$ and $A_{.j} \geq 0$ for all $j \in J$. If J is nonempty, then any convex combination $\bar{x}$ of the canonical vectors e^j, $j \in J$ also satisfies*

$$\|A\|_1 = \|A\bar{x}\|_1.$$

Proof Let

$$\bar{x} = \sum_{j \in J} \bar{x}_j e^j$$

where $\bar{x}_j \geq 0$, for all $j \in J$ and $\sum_{j \in J} \bar{x}_j = 1$. Then

$$(A\bar{x})_i = \sum_{j \in J} a_{ij} \bar{x}_j, \quad i = 1, 2, \dots, n.$$

Since $A_{.j} \geq 0$ for all $j \in J$, we have

$$\begin{aligned} \|A\bar{x}\|_1 &= \sum_{i=1}^{n} \left| \sum_{j \in J} a_{ij} \bar{x}_j \right| \\ &= \sum_{i=1}^{n} \sum_{j \in J} a_{ij} \bar{x}_j \\ &= \sum_{j \in J} \bar{x}_j \left(\sum_{i=1}^{n} a_{ij} \right) \\ &= \sum_{j \in J} \bar{x}_j \left| \sum_{i=1}^{n} a_{ij} \right| \\ &= \left(\sum_{j \in J} \bar{x}_j \right) \|A\|_1 = \|A\|_1. \end{aligned}$$

□

For instance, let

$$A = \begin{bmatrix} 1 & 4 & 0 \\ 2 & 0 & -1 \\ 3 & 2 & 4 \end{bmatrix}.$$

Hence $J = \{1, 2\}$. If $\bar{x} = \frac{2}{3} e^1 + \frac{1}{3} e^2 = [\frac{2}{3}\ \frac{1}{3}\ 0]^T$, then $\|A\bar{x}\|_1 = \|A\|_1 = 6$.

Hager's algorithm (Hager 1984) has been designed to compute a stationary point for the optimization problem (4). The algorithm investigates only vectors of the canonical basis with exception of the initial point, and it moves through these vectors using search directions based on the gradient of F. If F is not differentiable at one of these vectors, then a subgradient of F can be easily computed (Hager 1984). The inequality (5) provides a stopping criterion for the algorithm. The steps of the algorithm are presented below, where $\text{sign}(y)$ denotes a vector with components

$$\big(\text{sign}(y)\big)_i = \begin{cases} 1 & \text{if } y_i \geq 0, \\ -1 & \text{if } y_i < 0. \end{cases}$$

HAGER'S ESTIMATOR FOR $\|A\|_1$

Step 0: *Initialization*
Let $x = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$.

Step 1: *Determination of the Gradient or Subgradient of F, z*

Compute $y = Ax$,

$$\xi = \text{sign}(y),$$

$$z = A^T \xi.$$

Step 2: *Stopping Criterion*
If $\|z\|_\infty \leq z^T x$, stop with $\gamma = \|y\|_1$ the estimate for $\|A\|_1$.

Step 3: *Search Direction*
Let r be such that $\|z\|_\infty = |z_r| = \max_{1 \leq i \leq n} |z_i|$. Set $x = e^r$ and return to Step 1.

This algorithm can now be used to find an estimation of $\|B^{-1}\|_1$ by setting $A = B^{-1}$. In this case, the vectors y and z are computed by

$$y = B^{-1}x \quad \Leftrightarrow \quad By = x,$$
$$z = \left(B^{-1}\right)^T \xi \quad \Leftrightarrow \quad B^T z = \xi.$$

So, in each iteration of Hager's method for estimating $\|B^{-1}\|_1$ two linear systems with the matrices B and B^T are required to be solved. If the LU decomposition of B is known, then each iteration of the algorithm corresponds to the solution of four triangular systems. Since the algorithm only guarantees a stationary point of the function $F(x) = \|Ax\|_1$ on the convex set K, the procedure is applied from different starting points in order to get a better estimate for $\text{cond}_1(B)$. The whole procedure is discussed in (Higham 1996) and has been implemented in MATLAB (Moler et al. 2001). Furthermore, the algorithm provides an estimator for $\text{cond}_1(B)$ that is smaller than or equal to the true value of the condition number of the matrix B in the 1-norm. The purpose of this paper is to investigate Hager's algorithm and to show how it might be improved.

In (Bertsekas 2003), the conditional gradient algorithm is discussed to deal with nonlinear programs with simple constraints. The algorithm is a descent procedure that possess global convergence toward a stationary point under reasonable hypothesis (Bertsekas 2003). To describe an iteration of the procedure consider again the nonlinear program (4) and suppose that F is differentiable. If $\bar{x} \in K$, then a search direction is computed as

$$d = \bar{y} - \bar{x}$$

where $\bar{y} \in K$ is the optimal solution of the convex program

$$\begin{aligned} OPT\text{:} \quad & \text{Maximize } \nabla F(\bar{x})^T (y - \bar{x}) \\ & \text{subject to } \ y \in K. \end{aligned}$$

Two cases may occur:

(i) the optimal value is nonpositive and $\bar{y}$ is a stationary point of F on K;
(ii) the optimal value is positive and d is an ascent direction of F at $\bar{x}$.

In order to compute this optimal solution $\bar{y}$, we introduce the following change of variables:

$$\begin{aligned} & y_i = u_i - v_i, \\ & u_i \geq 0, \qquad v_i \geq 0, \\ & u_i v_i = 0, \quad \text{for all } i = 1, 2, \dots, n. \end{aligned} \tag{6}$$

Then OPT is equivalent to the following Mathematical Program with Linear Complementarity Constraints:

$$\begin{aligned} & \text{Maximize } \nabla F(\bar{x})^T (u - v - \bar{x}) \\ & \text{subject to } e^T u + e^T v \leq 1, \\ & \qquad u \geq 0, \qquad v \geq 0, \\ & \qquad u^T v = 0, \end{aligned}$$

where e is a vector of ones of order n. As discussed in (Murty 1976), the complementarity constraint is redundant and OPT is equivalent to the following linear program:

$$\begin{aligned} LP\text{:} \quad & \text{Maximize } \nabla F(\bar{x})^T (u - v - \bar{x}) \\ & \text{subject to } \ e^T u + e^T v \leq 1, \\ & \qquad u \geq 0, \qquad v \geq 0. \end{aligned}$$

Suppose that

$$\left|\nabla_r F(\bar{x})\right| = \max_{1 \leq i \leq n} \left|\nabla_i F(\bar{x})\right|.$$

An optimal solution $(\bar{u}, \bar{v})$ of LP is given by

$$\text{(i)} \quad \nabla_r F(\bar{x}) > 0 \quad \Rightarrow \quad \bar{u}_j = \begin{cases} 1 & \text{if } j = r, \\ 0 & \text{otherwise,} \end{cases} \qquad \bar{v} = 0,$$

$$\text{(ii)} \quad \nabla_r F(\bar{x}) < 0 \quad \Rightarrow \quad \bar{v}_j = \begin{cases} 1 & \text{if } j = r, \\ 0 & \text{otherwise,} \end{cases} \qquad \bar{u} = 0.$$

It then follows from (6) that

$$\bar{y} = \operatorname{sign}\bigl(\nabla_r F(\bar{x})\bigr)e^r.$$

After the computation of the search direction, a stepsize can be computed by an exact line-search

$$\begin{aligned}&\text{Maximize } F(\bar{x} + \alpha d) = g(\alpha)\\ &\text{subject to } \alpha \in [0, 1].\end{aligned}$$

Since F is convex on $\mathbb{R}^n$, g is convex on $[0, 1]$ and the maximum is achieved at $\alpha = 1$ (Bazaraa et al. 1993). Hence the next iterate computed by the algorithm is given by

$$\tilde{x} = \bar{x} + \bar{y} - \bar{x} = \bar{y} = \operatorname{sign}\bigl(\nabla_r F(\bar{x})\bigr)e^r.$$

Hence, the conditional gradient algorithm is essentially the same as Hager's gradient method for (4). Except for signs, the same iterates are generated and the algorithms terminate within n steps at the same iteration.

3 A simplex formulation

In this section we introduce a different nonlinear program for $\|A\|_1$ estimation

$$\begin{aligned}&\text{Maximize } F(x) = \|Ax\|_1 = \sum_{i=1}^{n}\left|\sum_{j=1}^{n} a_{ij}x_j\right|\\ &\text{subject to } e^T x = 1,\\ &\qquad\qquad\quad x \geq 0.\end{aligned} \tag{7}$$

Note that the constraint set of this nonlinear program is the ordinary simplex Δ. Since Δ has the extreme points $e^1, e^2, \ldots, e^n$, the maximum of the F function is attained at one of them, which gives the 1-norm of the matrix A. As in Theorem 1, this maximum can also be attained at a point that is not a vertex of the simplex.

As before the difficulty for computing $\|A\|_1$ is due to the concavity of program (7). If the matrix has nonnegative elements the problem can be easily solved. In fact, if $A \geq 0$ and $x \geq 0$, then

$$F(x) = \sum_{i=1}^{n}\left|\sum_{j=1}^{n} a_{ij}x_j\right| = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_j = \sum_{j=1}^{n}\left(\sum_{i=1}^{n} a_{ij}\right)x_j.$$

Hence

$$F(x) = d^T x$$

with

$$d = A^T e.$$

Therefore the program (7) reduces to the linear program

$$\begin{aligned} &\text{Maximize } d^T x \\ &\text{subject to } e^T x = 1, \\ &\qquad x \geq 0 \end{aligned}$$

which has the optimal solution

$$x = e^r, \quad \text{with } d_r = \max\{d_i : i = 1, \dots, n\}.$$

Hence

$$\|A\|_1 = d_r.$$

This observation also appears in (Hager 1984).

Like the previous algorithm, the importance of this procedure is due to the possibility of computing a good matrix condition number estimator. Let A be a Minkowski matrix, i.e., $A \in P$ and the nondiagonal elements are all nonnegative. Then $A^{-1} \geq 0$ (Cottle et al. 1992) and

$$\left\|A^{-1}\right\|_1 = d_r = \max\{d_i, i = 1, \dots, n\},$$

with d the vector given by

$$A^T d = e. \tag{8}$$

It should be noted that this type of matrices appears very often in the solution of second order ordinary and elliptic partial differential equations by finite elements and finite differences (Johnson 1990). So the condition number of a Minkowski matrix can be computed by solving a unique system of linear equations (8) and setting

$$\text{cond}_1(A) = \|A\|_1 d_r$$

where d is the unique solution of (8).

In the general case let us assume that F is differentiable. Then the following result holds.

Theorem 2 *If F is differentiable on an open set containing the simplex Δ and if $\max_i \nabla_i F(x) \leq \nabla F(x)^T x$, then x is a stationary point of F on Δ.*

Proof For each $y \in \Delta$,

$$\begin{aligned} \nabla F(x)^T y = \sum_{i=1}^n \nabla_i F(x) y_i &\leq \sum_{i=1}^n \big(\max \nabla_i F(x)\big) y_i \\ &\leq \max_{1\leq i\leq n} \nabla_i F(x) \left(\sum_{i=1}^n y_i\right) = \max_{1\leq i\leq n} \nabla_i F(x). \end{aligned}$$

Hence, by hypothesis,

$$\nabla F(x)^T(y-x) \leq 0, \forall y \in \Delta$$

and x is a stationary point of F on Δ. □

Next we describe the conditional gradient algorithm for the solution of the nonlinear program (7). As before, if $\bar{x} \in \Delta$ then the search direction d is given by

$$d = \bar{y} - \bar{x}$$

where $\bar{y} \in \Delta$ is the optimal solution of the linear program

$$\begin{aligned} LP: \quad & \text{Maximize } \nabla F(\bar{x})^T(y-\bar{x}) \\ & \text{subject to } e^T y = 1, \\ & \qquad\qquad y \geq 0. \end{aligned}$$

Two cases may occur.

(i) Maximum value is nonpositive and $\bar{y}$ is a stationary point of F on Δ.
(ii) Maximum value is positive and d is an ascent direction of F at $\bar{x}$.

Furthermore $\bar{y}$ is given by

$$\bar{y}_i = \begin{cases} 1 & \text{if } i = r, \\ 0 & \text{otherwise} \end{cases}$$

where r is the index satisfying

$$\nabla_r F(\bar{x}) = \max_{1 \leq i \leq n} \{\nabla_i F(\bar{x})\}.$$

An exact line-search along this direction leads to $\alpha = 1$ and the next iterate is

$$\tilde{x} = \bar{x} + \bar{y} - \bar{x} = \bar{y} = e^r.$$

Hence the conditional gradient algorithm for estimating $\|A\|_1$ can be described as follows.

CONDITIONAL GRADIENT ESTIMATOR FOR $\|A\|_1$

Step 0: *Initialization*
Let $x = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$.

Step 1: *Determination of the Gradient or Subgradient of F, z*
Compute $y = Ax$,
$\xi = \text{sign}(y)$,
$z = A^T \xi$.

Step 2: *Stopping Criterion*
If $\max_{1 \leq i \leq n} z_i \leq z^T x$, stop the algorithm with $\gamma = \|y\|_1$ the estimate for $\|A\|_1$.

Step 3: *Search Direction*
Consider $z_r = \max_{1 \le i \le n} z_i$. Set $x = e^r$ and return to Step 1.

As discussed in (Bertsekas 2003) this algorithm possesses global convergence toward a stationary point of the function F on the simplex. Furthermore, the following result holds:

Theorem 3 *If $\bar{x}$ is not a stationary point, then $F(e^r) > F(\bar{x})$, for r such that*

$$\nabla_r F(\bar{x}) = \max\{\nabla_i F(\bar{x}) : i = 1, \dots, n\}.$$

Proof

$$\begin{aligned} F(e^r) &\ge F(\bar{x}) + \nabla F(\bar{x})^T (e^r - \bar{x}) = F(\bar{x}) + [\nabla_r F(\bar{x}) - \nabla F(\bar{x})^T \bar{x}] \\ &= F(\bar{x}) + \Big[\max_i \nabla_i F(\bar{x}) - \nabla F(\bar{x})^T \bar{x}\Big] > F(\bar{x}) \end{aligned}$$

by Theorem 2. □

Since the simplex has n vertices the algorithm should converge to a stationary point of F in a finite number of iterations. Furthermore the algorithm is strongly polynomial, as the computational effort per iteration is polynomial in the order of the matrix A.

As mentioned before, the estimation of $\|A^{-1}\|_1$ can be done by setting $A = B^{-1}$ for this algorithm. If the LU decomposition of B is available, then, as in Hager's method, the algorithm requires the solution of four triangular systems in each iteration. Unfortunately, the asymptotic rate of convergence of the conditional gradient method is not very fast when Δ is a polyhedron (Bertsekas 2003) and gradient projection methods often converge faster. In the next section we discuss the use of the so-called Spectral Projected Gradient algorithm (Birgin et al. 2000; Júdice et al. 2008) to find an estimate of $\|A\|_1$.

4 A spectral projected gradient algorithm

As discussed in (Birgin et al. 2000, 2001), the Spectral Projected Gradient (SPG) algorithm uses in each iteration k a vector $x_k \in \Delta$ and moves along the projected gradient direction defined by

$$d_k = P_\Delta\big(x_k + \eta_k \nabla F(x_k)\big) - x_k \tag{9}$$

where $P_\Delta(w)$ is the projection of $w \in \mathbb{R}^n$ onto Δ. In each iteration k, x_k is updated by $x_{k+1} = x_k + \delta_k d_k$, where $0 < \delta_k \le 1$ is computed by a line search technique. The algorithm converges to a stationary point of F (Birgin et al. 2000). Next we discuss the important issues for the application of the SPG algorithm to the simplex formulation discussed in (Júdice et al. 2008).

(i) Computation of Stepsize: As before exact line search gives $\alpha_k = 1$ in each iteration k.

(ii) Computation of the Relaxation Parameter η_k:

Let $z_k = \nabla F(x_k)$ and $\eta_{\min}$ and $\eta_{\max}$ be a small and a huge positive real number, respectively, and as before, let $P_X(x)$ denote the projection of x over a set X. Then (Júdice et al. 2008)

(I) For $k = 0$,

$$\eta_0 = P_{[\eta_{\min},\eta_{\max}]}\left(\frac{1}{\| P_\Delta(x_0 + z_0) - x_0\|_\infty}\right).$$

(II) For any $k > 0$, let x_k be the corresponding iterate, z_k the gradient (subgradient) of F at x_k and

$$\begin{cases} s_{k-1} = x_k - x_{k-1}, \\ y_{k-1} = z_{k-1} - z_k. \end{cases}$$

Then

$$\eta_k = \begin{cases} P_{[\eta_{\min},\eta_{\max}]} \frac{\langle s_{k-1}, s_{k-1}\rangle}{\langle s_{k-1}, y_{k-1}\rangle} & \Leftarrow \langle s_{k-1}, y_{k-1}\rangle > \varepsilon, \\ \eta_{\max} & \Leftarrow \text{otherwise} \end{cases} \tag{10}$$

where ε is a quite small positive number.

(iii) Computation of the projection over Δ:

The projection onto Δ that is required in each iteration of the algorithm is computed as follows:

(I) Find $u = x_k + \eta_k z_k$.

(II) The vector $P_\Delta(u)$ is the unique optimal solution of the strictly convex quadratic problem

$$\begin{aligned} \underset{w\in\mathbb{R}^n}{\text{Minimize}} \quad & \frac{1}{2}\|u - w\|_2^2 \\ \text{subject to} \quad & e^T w = 1, \\ & w \geq 0. \end{aligned}$$

As suggested in (Júdice et al. 2008), a very simple and strongly polynomial Block Pivotal Principal Pivoting Algorithm is employed to compute this unique global minimum solution.

The steps of the SPG algorithm can be presented as follows.

SPECTRAL PROJECTED GRADIENT ESTIMATOR FOR $\|A^{-1}\|_1$

Step 0: *Initialization*

Let $x_0 = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$ and $k = 0$.

Step 1: *Determination of the Gradient or Subgradient of F, z_k*
Compute $Ay_k = x_k$,
$\xi = \text{sign}(y_k)$,
$A^T z_k = \xi$.

Step 2: *Stopping Criterion*
If $\max_{1\le i\le n}(z_k)_i \le z_k^T x_k$, stop the algorithm with $\gamma = \|y_k\|_1$ the estimate for $\|A^{-1}\|_1$.

Step 3: *Search Direction*
Update $x_{k+1} = P_\Delta(x_k + \eta_k z_k)$ and go to Step 1 with $k \leftarrow k+1$.

5 Computational experience

In this section we report some computational experience with the methods previously described to estimate $\|A^{-1}\|_1$. All the estimators have been implemented in MATLAB (Moler et al. 2001), version 7.1. The test problems have been taken from well-known sources.

Type I. Minkowski matrices (Johnson 1990), with the following structure:

$$\left[\begin{array}{cccc|cccc|c} 4 & -1 & & & & & & & \\ -1 & 4 & -1 & & & & & & \\ & -1 & 4 & -1 & & -I_4 & & & \\ & & -1 & 4 & & & & & \\ & & & & 4 & -1 & & & \\ & & & & -1 & 4 & -1 & & \\ & -I_4 & & & & -1 & 4 & -1 & -I_4 \\ & & & & & & -1 & 4 & \\ & & & & & -I_4 & & & \ddots \end{array}\right].$$

Type II. Matrices (Higham 1988) $A = \alpha I + e \times e^T$, where $e = [1\ 1\ \dots\ 1]^T$ and the parameter α takes values close to zero.

Type III. Triangular bidiagonal unitary matrices (Higham 1988), where the diagonal and the first lower diagonal are unitary and the remaining elements are zero. Hence these matrices have the form

$$\begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ 0 & 1 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Type IV. Pentadiagonal matrices (Murty 1988) with

$$m_{ii} = 6,$$

$$m_{i,i-1} = m_{i-1,i} = -4,$$
$$m_{i,i-2} = m_{i-2,i} = 1$$

and all other elements equal to zero.

Type V. Murty matrices (Murty 1988) with the structure

$$\begin{bmatrix} 1 & & & & \\ 2 & 1 & & & \\ 2 & 2 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ 2 & 2 & 2 & \dots & 1 \end{bmatrix}.$$

Type VI. Fathy matrices (Murty 1988) of the form

$$F = M^T M$$

where M is a *Murty* matrix.

Type VII. Matrices (Higham 1988) whose inverses can be written as

$$A^{-1} = I + \theta * C$$

where C is an arbitrary matrix such that

$$Ce = C^T e = 0,$$

and the parameter θ taking values close to zero.

Type VIII. Matrices from the `MatrixMarket` collection (Boisvert et al. 1998) that covers all of the `Harwell-Boeing` matrices. In Table 1 some specifications of the selected matrices are displayed.

Table 2 includes the numerical results of the performance of the three algorithms discussed in this paper for the estimation of the 1-norm condition number of the matrices mentioned before. The notations N, VALUE and IT are used to denote the order of the matrices, the value of the condition number estimator and the number of iterations provided by the three algorithms, which is the number of iterates visited by the method. Furthermore CN represents the true condition number of these matrices.

In general the SPG algorithm converges to a vertex of the simplex. In Table 2 we use the notation $(*)$ to indicate the problems where the algorithm converges to a point that is not a vertex of Δ. Finally column $\eta_{\max}$ indicates the value of $\eta_{\max}$ that leads to the best performance for the SPG algorithm.

The results show that the three algorithms are able to obtain similar estimations for the condition number of the matrices. The number of iterations for the three algorithms is always quite small. Furthermore for the best values of $\eta_{\max}$ the SPG algorithm finds in general the best estimate for the condition number.

The main drawback of the SPG algorithm relies on its dependence of the choice of the interval $[\eta_{\min}, \eta_{\max}]$. Our experience has shown that a value of $\eta_{\min} = 10^{-3}$ is in general a good choice. However the choice of $\eta_{\max}$ has an important factor on the computational effort. The results show that a value of $\eta_{\max}$ of 10^4 or 10^5 works

Table 1 Properties of the Type VIII matrices

Name	Original name	Origin	Order	Number of nonzeros elements	Type
MATRIX_1	s1rmq4m1	Finite-Elements	5489	143300	SPD
MATRIX_2	s2rmq4m1	Finite-Elements	5489	143300	SPD
MATRIX_3	s3rmq4m1	Finite-Elements	5489	143300	SPD
MATRIX_4	s1rmt3m1	Finite-Elements	5489	112505	SPD
MATRIX_5	s2rmt3m1	Finite-Elements	5489	112505	SPD
MATRIX_6	s3rmt3m1	Finite-Elements	5489	112505	SPD
MATRIX_7	bcsstk14	Structural Engineering	1806	32630	SPD
MATRIX_8	bcsstk15	Structural Engineering	3948	60882	SPD
MATRIX_9	bcsstk27	Structural Engineering	1224	28675	SPD
MATRIX_{10}	bcsstk28	Structural Engineering	4410	111717	Indefinite
MATRIX_{11}	orani678	Economic Modeling	2529	90158	Unsymmetric
MATRIX_{12}	sherman5	Petroleum Modeling	3312	20793	Unsymmetric
MATRIX_{13}	nos7	Finite Differences	729	2673	SPD
MATRIX_{14}	nos2	Finite Differences	957	2547	SPD
MATRIX_{15}	e20r5000	Dynamics of Fluids	4241	131556	Unsymmetric
MATRIX_{16}	fidapm37	Finite Elements Modeling	9152	765944	Unsymmetric
MATRIX_{17}	fidap037	Finite Elements Modeling	3565	67591	Unsymmetric

very well for many cases, but for other problems $\eta_{\max}$ should be chosen larger. It is important to note that the choice of $\eta_{\max}$ has even an impact on the stationary point obtained by the SPG algorithm, and of the corresponding condition number estimate.

As suggested in (Hager 1984), we decided to apply the SPG algorithm more than once starting in each iteration with an initial point that is the barycenter of the non visited vertices of the canonical basis in the previously iterations. The results are presented in Table 3, where NAP represents the number of applications of the SPG algorithm and NIT the total number of iterations for these experiments. These results show that the algorithm can improve the estimate in some cases, when the true condition number has not been obtained by the SPG algorithm.

6 Conclusions

The formulation of the condition number estimation problem was changed from a minimization over a unit sphere in the 1-norm to a minimization over the unit simplex. For this new constraint set, it is relatively easy to project a vector onto the feasible set. Hence, a projected gradient algorithm could be used to estimate the 1-norm condition number. Numerical experiments indicate that the so-called spectral projected gradient (SPG) algorithm (Birgin et al. 2000) can yield a better condition number estimate than the previous algorithm, while the number of iterations increased by at most one.

Table 2 Condition number estimators

Matrix		Hager's estimator		CG estimator		SPG estimator			CN
		Value	NIT	Value	NIT	Value	NIT	$\eta_{\max}$	
TYPE I	N								
	50	2.31E+001	2	2.31E+001	2	2.31E+001	2	1.0E+08	2.31E+001
	250	2.40E+001	2	2.40E+001	2	2.40E+001	3*	1.0E+08	2.40E+001
	500	2.40E+001	2	2.40E+001	2	2.40E+001	3*	1.0E+08	2.40E+001
	1000	2.40E+001	2	2.40E+001	2	2.40E+001	2*	1.0E+08	2.40E+001
	2000	2.40E+001	2	2.40E+001	2	2.40E+001	3*	1.0E+04	2.40E+001
	4000	2.40E+001	2	2.40E+001	2	2.40E+001	2*	1.0E+04	2.40E+001
TYPE II	α	$(N = 4000)$							
	0.50	1.60E+004	2	1.60E+004	2	1.60E+004	2	1.0E+16	1.60E+004
	0.25	3.20E+004	2	3.20E+004	2	3.20E+004	2	1.0E+16	3.20E+004
	0.125	6.40E+004	2	6.40E+004	2	6.40E+004	2	1.0E+16	6.40E+004
	0.01	8.00E+005	2	8.00E+005	2	8.00E+005	3	1.0E+12	8.00E+005
	1.0E-03	8.00E+006	2	8.00E+006	2	8.00E+006	3	1.0E+12	8.00E+006
	1.0E-04	8.00E+007	2	8.00E+007	2	8.00E+007	3*	1.0E+10	8.00E+007
	1.0E-05	8.00E+008	2	8.00E+008	2	8.00E+008	3	1.0E+10	8.00E+008
TYPE III	N								
	50	9.80E+001	2	9.80E+001	2	9.80E+001	3	1.0E+04	1.00E+002
	250	4.98E+002	2	4.98E+002	2	4.98E+002	3	1.0E+04	5.00E+002
	500	9.98E+002	2	9.98E+002	2	9.98E+002	3	1.0E+04	1.00E+003
	1000	2.00E+003	2	2.00E+003	2	2.00E+003	3	1.0E+04	2.00E+003
	2000	4.00E+003	2	4.00E+003	2	4.00E+003	3	1.0E+04	4.00E+003
	4000	8.00E+003	2	8.00E+003	2	8.00E+003	3	1.0E+04	8.00E+003

Table 2 (*Continued*)

Matrix		Hager's estimator		CG estimator		SPG estimator			CN
		Value	NIT	Value	NIT	Value	NIT	$\eta_{\max}$	
TYPE IV	N								
	50	3.04E+005	2	3.04E+005	2	3.04E+005	3	1.0E+11	3.04E+005
	250	1.68E+008	2	1.68E+008	2	1.68E+008	3	1.0E+06	1.68E+008
	500	2.65E+009	2	2.65E+009	2	2.65E+009	3	1.0E+06	2.65E+009
	1000	4.20E+010	2	4.20E+010	2	4.20E+010	2	1.0E+06	4.20E+010
	2000	6.69E+011	2	6.69E+011	2	6.69E+011	2	1.0E+06	6.69E+011
	4000	1.07E+013	2	1.07E+013	2	1.07E+013	2	1.0E+06	1.07E+013
TYPE V	N								
	50	9.80E+003	2	9.80E+003	2	9.80E+003	2	1.0E+04	9.80E+003
	250	2.49E+005	2	2.49E+005	2	2.49E+005	2	1.0E+04	2.49E+005
	500	9.98E+005	2	9.98E+005	2	9.98E+005	2	1.0E+04	9.98E+005
	1000	4.00E+006	2	4.00E+006	2	4.00E+006	2	1.0E+04	4.00E+006
	2000	1.60E+007	2	1.60E+007	2	1.60E+007	2	1.0E+04	1.60E+007
	4000	6.40E+007	2	6.40E+007	2	6.40E+007	2	1.0E+04	6.40E+007
TYPE VI	N								
	50	2.50E+007	2	2.50E+007	2	2.50E+007	2	1.0E+04	2.50E+007
	250	1.56E+010	2	1.56E+010	2	1.56E+010	2	1.0E+04	1.56E+010
	500	2.50E+011	2	2.50E+011	2	2.50E+011	2	1.0E+04	2.50E+011
	1000	4.00E+012	2	4.00E+012	2	4.00E+012	2	1.0E+04	4.00E+012
	2000	6.40E+013	2	6.40E+013	2	6.40E+013	2	1.0E+04	6.40E+013
	4000	1.02E+015	2	1.02E+015	2	1.02E+015	2	1.0E+04	1.02E+015

Table 2 (*Continued*)

Matrix		Hager's estimator		CG estimator		SPG estimator			CN
		Value	NIT	Value	NIT	Value	NIT	η_{max}	
TYPE VII	θ	$(N = 4000)$							
	0.50	5.44E+013	3	5.43E+013	3	8.14E+013	3	1.0E+02	8.14E+013
	0.25	2.37E+013	3	2.37E+013	3	2.73E+013	3	1.0E+02	2.75E+013
	0.125	1.01E+013	4	1.01E+013	4	1.09E+013	3	1.0E+02	1.10E+013
	0.01	1.67E+011	3	1.67E+011	3	2.12E+011	3	1.0E+05	2.14E+011
	1.0E-03	5.68E+009	3	5.68E+009	3	6.89E+009	3	1.0E+06	6.99E+009
	1.0E-04	1.27E+008	3	1.27E+008	3	2.74E+008	4	1.0E+08	2.90E+008
	1.0E-05	7.14E+006	4	7.14E+006	4	8.10E+006	4	1.0E+08	8.15E+006
TYPE VIII	NAME								
	MATRIX_1	7.47E+004	2	7.47E+004	2	7.47E+004	2	1.0E+05	7.47E+004
	MATRIX_2	7.66E+005	2	7.66E+005	2	7.66E+005	2	1.0E+05	7.66E+005
	MATRIX_3	4.54E+007	3	4.45E+007	3	4.45E+007	3	1.0E+05	4.54E+007
	MATRIX_4	4.49E+004	2	4.49E+004	2	4.49E+004	2	1.0E+05	4.93E+004
	MATRIX_5	9.22E+005	2	9.22E+005	2	9.22E+005	2	1.0E+05	9.22E+005
	MATRIX_6	4.39E+007	2	4.39E+007	2	4.39E+007	2	1.0E+05	4.62E+007
	MATRIX_7	1.11E+010	2	1.11E+010	2	1.11E+010	2*	1.0E+05	1.11E+010
	MATRIX_8	7.66E+009	2	7.66E+009	2	7.66E+009	2*	1.0E+05	7.66E+009
	MATRIX_9	1.23E+004	2	1.23E+004	2	1.23E+004	3	1.0E+05	1.23E+004
	MATRIX_{10}	4.49E+004	2	4.49E+004	2	4.49E+004	3	1.0E+05	4.49E+004
	MATRIX_{11}	1.00E+007	2	1.00E+007	2	1.00E+007	2	1.0E+05	1.00E+007
	MATRIX_{12}	3.90E+005	2	3.90E+005	2	3.90E+005	2	1.0E+05	3.90E+005
	MATRIX_{13}	3.00E+008	2	3.00E+008	2	3.00E+008	3*	1.0E+08	3.00E+008

Table 2 (*Continued*)

Matrix	Hager's estimator		CG estimator		SPG estimator			CN
	Value	NIT	Value	NIT	Value	NIT	$\eta_{\max}$	
MATRIX$_{14}$	3.71E+005	2	3.71E+005	2	3.71E+005	3*	1.0E+05	3.71E+005
MATRIX$_{15}$	1.84E+010	2	1.84E+010	2	1.84E+010	2	1.0E+05	1.84E+010
MATRIX$_{16}$	3.05E+010	2	3.05E+010	2	3.05E+010	2	1.0E+05	3.05E+010
MATRIX$_{17}$	2.26E+002	2	2.26E+002	2	2.26E+002	2*	1.0E+05	2.26E+002

Table 3 Performance of the SPG algorithm with a different initial point

Matrix		Previous estimative	SPG estimator			CN
			Value	NIT	NAP	
TYPE III	N					
	50	9.80E+001	1.00E+002	5	2	1.00E+002
	250	4.98E+002	5.00E+002	5	2	5.00E+002
	500	9.98E+002	1.00E+003	5	2	1.00E+003
TYPE VII	θ	$(N = 4000)$				
	0.25	2.73E+013	2.73E+013	6	2	2.75E+013
	0.125	1.09E+013	1.09E+013	23	7	1.10E+013
	0.01	2.12E+011	2.12E+011	6	2	2.14E+011
	1.0E-03	6.89E+009	6.89E+009	9	3	6.99E+009
	1.0E-04	2.74E+008	2.74E+008	16	4	2.90E+008
	1.0E-05	8.10E+006	8.10E+006	12	3	8.15E+006
TYPE VIII	NAME					
	MATRIX3	4.45E+07	4.54E+07	5	2	4.54E+07
	MATRIX4	4.49E+04	4.93E+04	8	4	4.93E+04
	MATRIX6	4.39E+07	4.60E+07	8	3	4.62E+07

References

Bazaraa MS, Sherali HD, Shetty C (1993) Nonlinear programming: theory and algorithms, 2nd edn. Wiley, New York

Bertsekas DP (2003) Nonlinear programming, 2nd edn. Athena Scientific, Belmont

Birgin EG, Martínez JM, Raydan M (2000) Nonmonotone spectral projected gradient methods on convex sets. SIAM J Optim 10:1196–1211

Birgin EG, Martínez JM, Raydan M (2001) Algorithm 813: SPG-software for convex constrained optimization. ACM Trans Math Softw 27:340–349

Boisvert R, Pozo R, Remington K, Miller B, Lipman R (1998) Mathematical and Computational Sciences Division of Information Technology Laboratory of National Institute of Standards and Technology. Available at http://math.nist.gov/MatrixMarket/formats.html

Cottle R, Pang J, Stone R (1992) The linear complementarity problem. Academic Press, New York

Demmel JW (1997) Applied numerical linear algebra. SIAM, Philadelphia

Gill P, Murray W, Wright M (1991) Numerical linear algebra and optimization. Addison-Wesley, New York

Golub GH, Loan CFV (1996) Matrix computations, 3rd edn. John Hopkins University Press, Baltimore

Hager WW (1984) Condition estimates. SIAM J Sci Stat Comput 5(2):311–316

Hager WW (1998) Applied numerical linear algebra. Prentice Hall, New Jersey

Higham NJ (1988) FORTRAN codes for estimating the one-norm of a real or complex matrix with applications to condition estimation. ACM Trans Math Softw 14(4):381–396

Higham NJ (1996) Accuracy and stability of numerical algorithms. SIAM, Philadelphia

Johnson C (1990) Numerical solution of partial differential equations by the finite element method. Cambridge University Press, London

Júdice JJ, Raydan M, Rosa SS, Santos SA (2008) On the solution of the symmetric eigenvalue complementarity problem by the spectral projected gradient algorithm. Numer Algorithms 44:391–407

Moler C, Little JN, Bangert S (2001) Matlab user's guide—the language of technical computing. The MathWorks, Sherborn

Murty K (1976) Linear and combinatorial programming. Wiley, New York

Murty K (1988) Linear complementarity, linear and nonlinear programming. Heldermann, Berlin