C. R. Acad. Sci. Paris, Ser. I 350 (2012) 639–641

Contents lists available at SciVerse ScienceDirect

C. R. Acad. Sci. Paris, Ser. I

www.sciencedirect.com

Numerical Analysis

Round-off estimates for second-order conic feasibility problems

Estimations d'arrondi pour des problèmes de faisabilité coniques du second ordre

Felipe Cucker [a,1], Javier Peña [b], Vera Roshchina [a]

[a] *Department of Mathematics, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong*
[b] *Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA*

ARTICLE INFO

Article history:
Received 23 May 2012
Accepted 26 June 2012
Available online 4 July 2012

Presented by Philippe G. Ciarlet

ABSTRACT

We present the analysis of an interior-point method to decide feasibility problems of second-order conic systems. A main feature of this algorithm is that arithmetic operations are performed with finite precision. Bounds for both the number of arithmetic operations and the finest precision required are exhibited.

RÉSUMÉ

Nous présentons une analyse de la méthode des points intérieurs pour résoudre les problèmes de faisabilité des systèmes coniques du second ordre. Une caractéristique principale de cet algorithme est que les opérations arithmétiques sont effectuées en précision finie. Des estimations du nombre des opérations arithmétiques et de la précision requise sont obtenues.

1. Second-order homogeneous feasibility problems

A common form (after homogenizing, if necessary) of linear conic feasibility problem is the alternative pair

$$Ax = 0, \quad x \in K \tag{P}$$

and

$$-A^{\mathrm{T}} y \in K^*. \tag{D}$$

Here $A \in \mathbb{R}^{m\times n}$ is a matrix, $K \subset \mathbb{R}^n$ is a closed convex cone and $K^* \subset \mathbb{R}^n$ is the dual cone of K. We sometimes write $x \succcurlyeq_K 0$ instead of $x \in K$ and we write $x \succ_K 0$ to denote that x is in the relative interior of K.

It is well known that each of (P) and (D) above has a strict solution (one where the membership is in the relative interior of the cone) if and only if the other has no solutions at all (the point 0 is not considered as a solution for these problems). The feasibility problem consists in, given A, deciding which of the two systems, if any, has a strict solution.

E-mail addresses: macucker@math.cityu.edu.hk (F. Cucker), jfp@andrew.cmu.edu (J. Peña), vera.roshchina@gmail.com (V. Roshchina).
[1] Partially supported by CityU SRG grant #7008053.

1631-073X/$ – see front matter
http://dx.doi.org/10.1016/j.crma.2012.06.013

Paramount examples of K are the positive orthant $[0,\infty)^n$—which gives rise to linear programming—, the cone of positive semidefinite matrices—at the basis of semidefinite programming—, and the second-order cone. The latter is defined to be the product of a finite number of Lorentz cones $\mathcal{L}_p \subset \mathbb{R}^{p+1}$ which, in turn, are given by

$$\mathcal{L}_p := \left\{ x \in \mathbb{R}^{p+1} \;\middle|\; x_0 \geqslant \left\| (x_1, \ldots, x_p) \right\| \right\}.$$

Problems induced by second-order cones, known as second-order programming problems, are a particular case of semidefinite programming and contain as a particular case all instances of linear programming. They have lately received considerable attention due to their many applications [11] and since they appear to be at the boundary of the problems for which interior-point methods can solve large instances, a fact that is linked to the appearance of commercial software for the solution of second-order programs such as MOSEK[2] or CPLEX.[3]

2. Interior-point methods

An analysis of interior-point methods for SOCP can be found in [13] (see also the references therein). A theoretical analysis of interior-point methods for general convex cones can be found in [9]. As it was remarked in [1,11], however, solving SOCP problems by reducing them to semidefinite programs (or to more general conic problems) is not a good idea since interior-point methods that solve the SOCP directly have a much better complexity (both theoretically and in practice) than when applied to these more general reformulations.

In this Note we present an analysis of a finite-precision algorithm for solving the SOCP feasibility problem. This algorithm is designed to work with variable precision, that is, the machine precision can be re-adjusted along the way. Our Main Theorem shows bounds for the finest machine precision and the maximal number of iterations (and hence, of arithmetic operations) needed by this algorithm. We next define the main ingredients of these bounds.

2.1. Condition and well-posedness

Our bounds depend on Renegar's condition number [8,9], which is consistent with similar bounds obtained for the polyhedral case in [5]. Let $\rho_P(A)$ and $\rho_D(A)$ be the *distance to infeasibility* of (P) and (D) respectively defined by

$$\rho_P(A) = \inf\left\{ \|\Delta A\|:\ (A+\Delta A)x = 0,\ x \succ_K 0 \text{ is infeasible} \right\}$$

and

$$\rho_D(A) = \inf\left\{ \|\Delta A\|:\ -(A+\Delta A)^{\mathrm{T}} y \succ_K 0,\ y \in \mathbb{R}^m \text{ is infeasible} \right\}.$$

Note, we used in this definition the fact that a second-order cone K is self-dual, i.e., $K = K^*$. Then Renegar's *condition number* $C(A)$ is defined as the reciprocal of the relative distance to ill-posedness of the pair (P)–(D):

$$C(A) := \frac{\|A\|}{\max\{\rho_P(A), \rho_D(A)\}}.$$

Although any equivalent matrix norm can be used to define $C(A)$, in our analysis we choose to use the standard operator norm induced by the Euclidean scalar product. We say that the problem is ill-posed if both $\rho_P(A) = \rho_D(A) = 0$ and hence $C(A) = \infty$.

2.2. Finite precision and approximate solutions

A common feature of all studies on SOCP made up to now is the assumption of infinite precision in all the performed arithmetic operations. In this Note we will assume instead a context of finite precision in which the following quantity plays the key role.

The *round-off unit* or *machine precision* of a machine is a number $u \in \mathbb{R}$, $0 < u < 1$, such that real numbers x in the machine are systematically replaced by approximations $\mathrm{round}(x)$ satisfying $|\mathrm{round}(x) - x| \leqslant u|x|$. Roughly, $|\log u|$ corresponds with the number of digits of the mantissa in the floating-point representation of $\mathrm{round}(x)$.

We consider variable precision algorithms. In this context the machine precision, though finite, varies during the computation. It is initially small and subsequently needs to be sharpened, but remains bounded.

The assumption of finite precision sets some limitations on the kind of solutions we may obtain. If system (D) has strict solutions, then we will obtain, after sufficiently refining the precision, a strict solution $y \in \mathbb{R}^m$. On the other hand, if the system having solution is (P), then there is no hope of exactly computing one such solution, since the set of solutions is thin in $\mathbb{R}^n$. In such a case there is no way to ensure that the errors produced by the use of finite precision will not move any candidate solution out of the set. We can, however, compute good approximations, namely, forward-approximate solutions.

[2] http://www.mosek.com/.
[3] http://www.ilog.com/products/cplex/.

Definition 1. Let $\gamma \in (0,1)$. A point $\hat{x} \in \mathbb{R}^n$ is a *γ-forward solution* of the system $Ax = 0$, $x \succ_K 0$, if $\hat{x} \succ_K 0$, and there exists $\breve{x} \in \mathbb{R}^n$ such that

$$A\breve{x} = 0, \quad \breve{x} \succ_K 0$$

and

$$\|\hat{x} - \breve{x}\| \leqslant \gamma \|\hat{x}\|.$$

The point $\breve{x}$ is said to be an *associated solution* for $\hat{x}$. A point is a *forward-approximate solution* of $Ax = 0$, $x \succcurlyeq_K 0$, if it is a γ-forward solution of the system for some $\gamma \in (0,1)$.

3. Statement of the main result

We estimate both the number of iterations of the algorithm and the precision required as functions of m, n, r and the condition number $\mathcal{C}(A)$. Our main result (recall, K is the product of r Lorentz cones) is the following (see [6] for a proof).

Main Theorem. *There exists a finite precision algorithm which, with input a matrix $A \in \mathbb{R}^{m\times n}$ and a number $\gamma \in (0,1)$, finds either a strict γ-forward solution $x \in \mathbb{R}^n$ of $Ax = 0$, $x \succcurlyeq_K 0$, or a strict solution $y \in \mathbb{R}^m$ of the system $A^{\mathrm{T}} y \preccurlyeq_K 0$. The machine precision varies during the execution of the algorithm. The finest required precision is*

$$u = \frac{1}{\mathbf{c}(m+n)^{5/2}\mathbf{r}^{20}\mathcal{C}(A)^5},$$

where $\mathbf{c}$ is a universal constant and $\mathbf{r} = r + 2$. The number of main (interior-point) iterations of the algorithm is bounded by

$$O\left(r^{1/2}\left(\log(r) + \log\bigl(\mathcal{C}(A)\bigr) + |\log\gamma|\right)\right)$$

if (P) is strictly feasible and by the same expression without the $|\log\gamma|$ term if (D) is.

Each such iteration is performed with $O(n^3)$ arithmetic operations.

3.1. A final remark

In the numerical analysis literature, fixed precision is used more commonly than variable precision. We note here that from our variable precision analysis we can obtain a fixed precision one. Indeed, assume the precision u is fixed. Then our algorithm could run until the point at which it should get a precision finer than u. If it found the answer before this point it could return it (and this answer would be guaranteed to be correct). If not, it could halt and return a failure message. Furthermore, the only reason for u to be insufficient is that $\mathcal{C}(A)$ is too large. Solving the bound for u in the Main Theorem we obtain a lower bound C_u for $\mathcal{C}(A)$. Thus, the failure message could be something like "*The condition of the data is larger than C_u. To solve the problem I need more precision*."

Finite precision analyses are pervasive in Numerical Linear Algebra. They are much less common in optimization. While the effects of finite precision when solving linear programming problems had been early noticed (e.g. [2,4,7,10,12,15]) there was no condition-based round-off analysis of linear programming problems until recently. This was done for the feasibility problem for polyhedral conic systems [5], for the optimal value of linear programs [14], and for the computation of optimal basis and optimal solutions of linear programs [3]. To the best of our knowledge, the Main Theorem above is the first such analysis for nonlinear cones.

References

[1] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Program. 95 (2003) 3–51.
[2] R.H. Bartels, A stabilization of the simplex method, Numer. Math. 16 (1971) 414–434.
[3] D. Cheung, F. Cucker, Solving linear programs with finite precision: II. Algorithms, J. Complexity 22 (2006) 305–335.
[4] R.J. Clasen, Techniques for automatic tolerance control in linear programming, Comm. ACM 9 (1966) 802–803.
[5] F. Cucker, J. Peña, A primal-dual algorithm for solving polyhedral conic systems with a finite-precision machine, SIAM J. Optim. 12 (2002) 522–554.
[6] F. Cucker, J. Peña, V. Roshchina, Solving second-order conic systems with finite-precision, preprint. Available at arXiv:1104.1352.
[7] W. Ogryczak, The simplex method is not always well behaved, Linear Algebra Appl. 109 (1988) 41–57.
[8] J. Renegar, Linear programming, complexity theory and elementary functional analysis, Math. Program. 70 (1995) 279–351.
[9] J. Renegar, A Mathematical View of Interior-Point Methods in Convex Optimization, SIAM, 2000.
[10] S.M. Robinson, A characterization of stability in linear programming, Oper. Res. 25 (1977) 435–447.
[11] M. Sousa Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear Algebra Appl. 284 (1998) 193–228.
[12] S. Storoy, Error control in the simplex-technique, BIT 7 (1967) 216–225.
[13] T. Tsuchiya, A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming, Optim. Methods Softw. 11 (1999) 141–182.
[14] J.R. Vera, On the complexity of linear programming under finite precision arithmetic, Math. Program. 80 (1998) 91–123.
[15] P. Wolfe, Error in the solution of linear programming problems, in: L.R. Ball (Ed.), Error in Digital Computation, John Wiley & Sons, 1965, pp. 271–284.