Computational experiments with a linear approximation of second-order cone optimization

Image Technical Report 0001

François Glineur*
Service de Mathématique et de Recherche Opérationnelle,
Faculté Polytechnique de Mons,
Rue de Houdain, 9, B-7000 Mons, Belgium,
`Francois.Glineur@fpms.ac.be`.

January 2000, revised November 2000

*The author is a research fellow supported by the F.N.R.S. (Belgian National Fund for Scientific Research). This research was partly carried out during a visit to the SSOR department of the Delft University of Technology supported by a grant of the Communauté Française de Belgique.

Abstract

In this article, we present and improve a polyhedral approximation of the second-order cone due to Ben-Tal and Nemirovski [BTN98]. We also discuss several ways of reducing the size of this approximation. This construction allows us to approximate second-order cone optimization problems with linear optimization.

We implement this scheme and conduct computational experiments dealing with two classes of second-order cone problems: the first one involves truss-topology design and uses a large number of second-order cones with relatively small dimensions, while the second one models convex quadratic optimization problems with a single large second-order cone.

Keywords. second-order cone optimization, approximation, linear optimization, quadratic optimization.

Contents

1 Introduction

Many useful classes of optimization problems one encounters both in theory and in practice are convex, such as linear optimization, convex quadratic optimization, semidefinite optimization and geometric optimization. This fundamental characteristic guarantees some nice properties: every local minima is a global minima, existence of strongly related dual problem and availability of efficient algorithms to solve these problems.

Conic optimization is a powerful setting that relies on convex cones to formulate convex problems: the standard conic primal-dual pair is the following [SW70, Stu99a]

$$\inf_{x \in \mathbb{R}^n} \; c^T x \qquad \text{s.t.} \qquad Ax = b \text{ and } x \in \mathcal{C} \tag{1.1}$$

$$\sup_{y \in \mathbb{R}^m, x^* \in \mathbb{R}^n} \; b^T y \qquad \text{s.t.} \qquad A^T y + x^* = c \text{ and } x^* \in \mathcal{C}^* \;, \tag{1.2}$$

where x and (y, x^*) are the primal and dual variables, A is a $m \times n$ matrix, b and c are m and n-dimensional column vectors, $\mathcal{C} \subseteq \mathbb{R}^n$ is a closed convex cone and $\mathcal{C}^* \subseteq \mathbb{R}^n$ is its dual cone, defined by $\mathcal{C}^* = \{x^* \in \mathbb{R}^n \mid x^T x^* \geq 0 \; \forall x \in \mathcal{C}\}$.

Different types of convex cones lead to different classes of problems: for example, linear optimization uses the nonnegative orthant $\mathbb{R}^n_+$ while semidefinite programming relies on the set of positive semidefinite matrices $\mathbb{S}^n_+$. In this article, we will focus the *second-order* cone, also known as *Lorentz* cone or *ice-cream cone*, which leads to *second-order cone optimization.* It is defined as follows:

Definition 1.1. *The second order cone $\mathbb{L}^n$ is the subset of $\mathbb{R}^{n+1}$ defined by*

$$\mathbb{L}^n = \{(r, x) \in \mathbb{R} \times \mathbb{R}^n \mid \|x\| \leq r\} \;,$$

where $\|\cdot\|$ denotes the usual Euclidean norm on $\mathbb{R}^n$.

It is indeed straightforward to check that this is a closed convex cone (it is in fact the epigraph of the Euclidean norm). Another interesting property of $\mathbb{L}^n$ is the fact that it is self-dual, i.e. $(\mathbb{L}^n)^* = \mathbb{L}^n$.

The standard second-order cone problems are based on the cartesian product of several second-order cones, which can be formalized using r constants $n_k \in \mathbb{N}, 1 \leq k \leq r$ such that $\sum_{k=1}^r (n_k + 1) = n$ and defining $\mathcal{C} = \mathbb{L}^{n_1} \times \mathbb{L}^{n_2} \times \cdots \times \mathbb{L}^{n_r}$. This set is obviously also a self-dual closed convex cone, which allows us to rewrite problems (1.1) and (1.2) as

$$\inf \; c^T x \qquad \text{s.t.} \qquad Ax = b \text{ and } x^k \in \mathbb{L}^{n_k} \; \forall k = 1, 2, \ldots, r \tag{1.3}$$

$$\sup \; b^T y \qquad \text{s.t.} \qquad A^T y + x^* = c \text{ and } x^{k*} \in \mathbb{L}^{n_k} \; \forall k = 1, 2, \ldots, r \;, \tag{1.4}$$

where vectors x and x^* have been split into r subvectors $(x^1, x^2, \ldots, x^r)$ and $(x^{1*}, x^{2*}, \ldots, x^{r*})$ with $x^{k*}, x^k \in \mathbb{R}^{n_k+1}$ for all $k = 1, \ldots, r$. It is usually more practical to pivot out variables x^* in the dual problem (1.4), i.e. write them as a function of vector y. Splitting matrix A into $(A^1, A^2, \ldots, A^r)$ with $A^k \in \mathbb{R}^{m \times (n_k+1)}$ and vector c into $(c^1, c^2, \ldots, c^r)$ with $c^k \in \mathbb{R}^{n_k+1}$, we have $x^{k*} = c^k - A^{kT} y$. The last step is to isolate the first column in A^k and the first component in c^k, i.e. letting $A^k = (f^k, G^k)$ with $f^k \in \mathbb{R}^m$ and $G^k \in \mathbb{R}^{m \times n_k}$ and $c^k = (d^k, h^k)$ with $d^k \in R$ and $h^k \in \mathbb{R}^{n_k}$, we can rewrite the dual problem (1.4) as

$$\max b^T y \quad \text{s.t.} \quad \left\|G^{kT} y + h^k\right\| \leq f^{kT} y + d^k \; \forall k = 1, 2, \ldots, r \;,$$

which is more convenient to formulate real-world problems.

Second-order optimization admits many different well-known classes of optimization problems as special cases, such as linear optimization, linearly and quadratically constrained convex quadratic optimization, robust linear optimization, matrix-fractional problems and problems with hyperbolic constraints (see the survey [LVBL98]). Applications arise in various fields such as engineering (antenna array design, finite impulse response filter design, truss design) and finance (portfolio optimization), see again [LVBL98].

From the computational point of view, second-order cone optimization is a relatively young field if compared to linear and quadratic optimization (for example, the leading commercial linear and quadratic solvers do not yet offer the option of solving second-order cone optimization problems). This observation led Ben-Tal and Nemirovski to develop an interesting alternative approach to solving second-order cone problems: they show in [BTN98] that it is possible to write a polyhedral approximation of the second order cone $\mathbb{L}^n$ with a prescribed accuracy ϵ using a number of variables and constraints that is polynomial in n and $\log \frac{1}{\epsilon}$. This implies that second-order cone optimization problems can be approximated with an arbitrarily prescribed accuracy by linear optimization problems using this polyhedral approximation.

This potentially allows the approximate resolution of large-scale second-order cone problems using state of the art linear solvers, capable of handling problems with hundreds of thousands of variables and constraints.

This paper is organized as follows: Section 2 presents a polyhedral approximation of the second-order cone. This construction relies on a decomposition scheme based on three-dimensional second order cones. We present first an efficient way to approximate these cones and then show how to combine them in order to approximate a second-order cone of higher dimension, which ultimately gives a method to approximate any second-order cone optimization problem with a linear problem. Section 3 reports our computational experiments with this scheme. After a presentation of our implementation and some related issues, we describe two classes of second-order problems: truss-topology design problems and convex quadratic optimization problems. We present and discuss the results of our computational experiments, highlighting when necessary the particular features of each class of problems (guaranteed accuracy, alternative formulations). We conclude this article with a few remarks and suggestions for further research.

2 Approximating second-order cone optimization

In this section, we present a polyhedral approximation of the second-order cone $\mathbb{L}^n$ which allows us to derive a linearizing scheme for second-order cone optimization. It is a variation of the construction of Ben-Tal and Nemirovski that features slightly better properties.

2.1 Principle

The principle that lies behind their approximation is twofold:

a. **Decomposition**. Since the Lorentz cone $\mathbb{L}^n$ is a $n+1$-dimensional subset, any circumscribed polyhedral cone around $\mathbb{L}^n$ is bound to have its number of facets growing exponentially with the dimension n, i.e. will need an exponential number of linear inequalities to be defined. The remedy is to decompose the second-order cone into a polynomial number of smaller second-order cones with fixed dimension, for which a good polyhedral approximation can be found. In the present case, $\mathbb{L}^n$ can be decomposed into $n-1$ three-dimensional second-order cones $\mathbb{L}^2$, at the price of introducing $n-2$ additional variables (see Section 2.2).

b. **Projection**. Even the three-dimensional second-order cone $\mathbb{L}^2$ is not too easy to approximate: the most obvious way to proceed, a regular circumscribed polyhedral cone, requires hundreds of inequalities even for an approximation with modest accuracy (see Section 2.3). The key idea to lower the number of inequalities is to introduce several additional variables, i.e. lift the approximating polyhedron into a higher dimensional space and consider its projection onto a $(n+1)$-dimensional subspace as the approximation of $\mathbb{L}^n$ (see Section 2.4).

To summarize, the introduction of a certain number of additional variables, combined with a projection, can be traded against a much lower number of inequality constraints defining the polyhedron. We first concentrate on the decomposition of $\mathbb{L}^n$ into smaller second-order cones.

2.2 Decomposition

Let us start with the following equivalent definition of $\mathbb{L}^n$

$$\mathbb{L}^n = \left\{(r, x_1, x_2, \ldots, x_n) \in \mathbb{R}_+ \times \mathbb{R}^n \mid \sum_{i=1}^{n} x_i^2 \leq r^2\right\}.$$

Introducing a vector of $\lfloor \frac{n}{2} \rfloor$ additional variables $y = (y_1, y_2, \ldots, y_{\lfloor \frac{n}{2} \rfloor})$, we consider the set $\mathbb{L}^{n\prime}$ defined by

$$\left\{(r, x, y) \in \mathbb{R}_+ \times \mathbb{R}^{n+\lfloor \frac{n}{2} \rfloor} \mid x_{2i-1}^2 + x_{2i}^2 \leq y_i^2,\ 1 \leq i \leq \lfloor \frac{n}{2} \rfloor,\ \begin{cases} \sum_{i=1}^{\frac{n}{2}} y_i^2 & \leq r^2 \ (n \text{ even}) \\ \sum_{i=1}^{\frac{n}{2}} y_i^2 + x_n^2 & \leq r^2 \ (n \text{ odd}) \end{cases}\right\}.$$

It it straightforward to prove that the projection of this set on the subspace of its first $n+1$ variables $(r, x_1, \ldots, x_n)$ is equal to $\mathbb{L}^n$, i.e.

$$(r, x) \in \mathbb{L}^n \quad \Leftrightarrow \quad \exists y \in \mathbb{R}^{\lfloor \frac{n}{2} \rfloor} \quad \text{s.t.} \quad (r, x, y) \in \mathbb{L}^{n\prime}.$$

It is also worth to point out that all the constraints defining $\mathbb{L}^{n\prime}$ are second-order cone constraints, i.e. that $\mathbb{L}^{n\prime}$ can also be written as

$$\left\{(r, x, y) \in \mathbb{R}_+ \times \mathbb{R}^{n+\lfloor \frac{n}{2} \rfloor} \mid (y_i, x_{2i-1}, x_{2i}) \in \mathbb{L}^2,\ 1 \leq i \leq \lfloor \frac{n}{2} \rfloor,\ \begin{cases} (r, y) & \in \mathbb{L}^{\lceil \frac{n}{2} \rceil} \ (n \text{ even}) \\ (r, y, x_n) & \in \mathbb{L}^{\lceil \frac{n}{2} \rceil} \ (n \text{ odd}) \end{cases}\right\}. \tag{2.1}$$

This means that $\mathbb{L}^n$ can be decomposed into $\lfloor \frac{n}{2} \rfloor$ small 3-dimensional second-order cones and a single $\mathbb{L}^{\lceil \frac{n}{2} \rceil}$ second-order cone, at the price of introducing $\lfloor \frac{n}{2} \rfloor$ auxiliary variables. This procedure can be applied recursively to the largest of the remaining second-order cone $\mathbb{L}^{\lceil \frac{n}{2} \rceil}$ until it also becomes equal to $\mathbb{L}^2$.

It is not too difficult to see that there are in the final expression $n-1$ second-order cones $\mathbb{L}^2$ and $n-2$ additional y_i variables. Indeed, the addition of each small cone $\mathbb{L}^2$ reduces the size of the largest cone by one, since we remove two variables from this cone (the last two variables in $\mathbb{L}^2$) but replace them with a single new variable (the first variable in $\mathbb{L}^2$).

Since we start with this largest cone equal $\mathbb{L}^n$ and stop when its size is equal to 2, we need $n-2$ small cones along with $n-2$ auxiliary variables to reduce the cone to $\mathbb{L}^2$. But this last $\mathbb{L}^2$ cone also has to be counted, which gives then a total number of cones equal to $n-1$.

The existence of this decomposition implies that any second-order cone optimization problem can be transformed into a problem using only 3-dimensional second-order cones, using the construction above. We note however that strictly speaking, the resulting formulation is not a conic problem, since some variables belong to two different cones at the same time. It is nonetheless possible to add a extra variable for each shared variable, along with a constraint to make them equal on the feasible region, to convert this formulation into the strict conic format (1.1)–(1.2).

2.3 A first approximation of $\mathbb{L}^2$

The previous section has shown that we can focus our attention on approximations of the 3-dimensional second-order. Moreover, it seems reasonable to require this approximation to be a cone itself too. Taking into account this additional assumption, we can take advantage of the homogeneity property of these cones to write

$$(r, x_1, x_2) \in \mathbb{L}^2 \Leftrightarrow (1, \frac{x_1}{r}, \frac{x_2}{r}) \in \mathbb{L}^2 \ , \tag{2.2}$$

which basically means we can fix $r = 1$ and look for a polyhedral approximation of the resulting set

$$\left\{x \in \mathbb{R}^2 \mid (1, x) \in \mathbb{L}^2\right\} = \left\{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1\right\} = \mathcal{B}_2(1) \ ,$$

which is exactly the disc of radius one in $\mathbb{R}^2$. Any approximating polyhedron for $\mathcal{B}_2(2)$ will be then later straightforwardly converted into a polyhedral cone approximating $\mathbb{L}^2$, using the additional homogenizing variable r.

At this point, we have to introduce a measure of the quality of our approximations. A natural choice for this measure is to state that a polyhedron $\mathcal{P} \subseteq \mathbb{R}^2$ is a ϵ-approximation of $\mathcal{B}_2(1)$ if and only we have the double inclusion $\mathcal{B}_2(1) \subseteq \mathcal{P} \subseteq \mathcal{B}_2(1+\epsilon)$, i.e. the polyhedron contains the unit disc but lies entirely within the disc of radius $1+\epsilon$.

The most obvious approximation of the unit disc is the regular m-polyhedron $\mathcal{P}_m$, which is described by m linear inequalities. We have the following theorem:

Theorem 2.1. *The regular polyhedron with m sides is an approximation of the unit disc $\mathcal{B}_2(1)$ with accuracy $\epsilon = \cos(\frac{\pi}{m})^{-1} - 1$.*

Proof. The proof is quite straightforward: looking at Figure 1 (which represents the case $m = 8$), we see that angle $\angle AOM$ is equal to $\frac{\pi}{m}$ and thus that $|OA| \cos(\frac{\pi}{m}) = |OM| = 1$. Our measure of quality is then equal to $\epsilon = |OA| - 1 = \cos(\frac{\pi}{m})^{-1} - 1$, as announced. □

This result is not very satisfying: since $\cos(x)^{-1} \approx 1 + \frac{x^2}{2}$ when x is small, we have that $\epsilon \approx \frac{\pi^2}{2m^2}$ when m is large, which means that doubling the number of inequalities only divides the accuracy by four. For example, approximating $\mathcal{B}_2(1)$ with the relatively modest accuracy 10^{-4} would already take a 223-sided polyhedron, i.e. more than 200 linear inequalities.

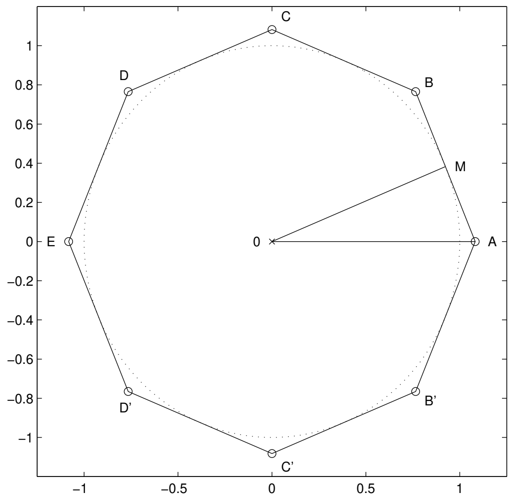

Figure 1: Approximating $\mathcal{B}_2(1)$ with a regular octagon.

2.4 A better approximation of $\mathbb{L}^2$

As outlined in Section 2.1, the key idea introduced by Ben-Tal and Nemirovski to obtain a better polyhedral approximation is to consider the projection of a polyhedron belonging to a higher dimensional space. The construction we are going to present here is a variation of the one described in [BTN98], featuring slightly better parameters and a more transparent proof.

Let us introduce an integer parameter $k \geq 2$ and consider the set $\mathcal{D}_k \subseteq \mathbb{R}^{2k+2}$ defined as

$$\left\{(\alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_k) \in \mathbb{R}^{2k+2} \mid \left\{ \begin{array}{rcrcr} \alpha_{i+1} & = & \alpha_i \cos \frac{\pi}{2^i} & + & \beta_i \sin \frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \beta_i \cos \frac{\pi}{2^i} & - & \alpha_i \sin \frac{\pi}{2^i} \\ -\beta_{i+1} & \leq & \beta_i \cos \frac{\pi}{2^i} & - & \alpha_i \sin \frac{\pi}{2^i} \\ 1 & = & \alpha_k \cos \frac{\pi}{2^k} & + & \beta_k \sin \frac{\pi}{2^k} \end{array} \right. \forall\, 0 \leq i < k \right\}.$$

This set is obviously a polyhedron[1], since its defining constraints consist in $k+1$ linear equalities and $2k$ inequalities. The following theorem gives some insight about the structure of this set.

Theorem 2.2. *The projection of the set $\mathcal{D}_k$ on the subspace of its two variables (α_0, β_0) is equal to the regular 2^k-sided polyhedron, i.e. we have*

$$(\alpha_0, \beta_0) \in \mathcal{P}_{2^k} \;\Leftrightarrow\; \exists(\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k) \in \mathbb{R}^{2k} \mid (\alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_k) \in \mathcal{D}_k \;.$$

Proof. To fix ideas, we are going to present some figures corresponding to the case $k = 3$, but our reasoning will of course be valid for all $k \geq 2$. Looking at Figure 1, which depicts $\mathcal{P}_{2^3}$, we see that the last equality in the definition of $\mathcal{D}_k$ describes the line AM. Indeed, we have $A = (\cos(\frac{\pi}{2^k})^{-1}, 0)$ and $M = (\cos(\frac{\pi}{2^k}), \sin(\frac{\pi}{2^k}))$ and it is straightforward that both of these points satisfy the last equality in the definition of $\mathcal{D}_k$.

Recall now that the application

$$\mathcal{R}_\theta : \mathbb{R}^2 \mapsto \mathbb{R}^2 : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \mathcal{R}_\theta \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \cos\theta + y \sin\theta \\ -x \sin\theta + y \cos\theta \end{pmatrix}$$

[1] Strictly speaking, this set is not a full-dimensional polyhedron in $\mathbb{R}^{2k+2}$ because of the additional linear constraints but this has no incidence on our purpose.

is a clockwise rotation around the origin with angle θ. Calling P_i the point of $\mathbb{R}^2$ whose coordinates are (α_i, β_i) and $\hat{P}_i = (\hat{\alpha}_i, \hat{\beta}_i)$ the image of P_i by rotation $\mathcal{R}_{\pi/2^i}$, we have that the first three constraints in the definition of $\mathcal{D}_k$ are equivalent to $\alpha_{i+1} = \hat{\alpha}_i$, $\beta_{i+1} \geq \hat{\beta}_i$ and $-\beta_{i+1} \leq \hat{\beta}_i$. These last two inequalities rewritten as $-\beta_{i+1} \leq \hat{\beta}_i \leq \beta_{i+1}$ immediately imply that β_{i+1} has to be nonnegative. Under this assumption, we call $\bar{P}_i$ the points whose coordinates are $(\alpha_i, -\beta_i)$ and find that these three constraints are equivalent to saying that $\hat{P}_i \in [P_{i+1}\ \bar{P}_{i+1}]$. In other words, the point $\hat{P}_i$ has to belong to a vertical segment $[P_{i+1}\ \bar{P}_{i+1}]$ such that P_{i+1} has its second coordinate nonnegative. Since $\hat{P}_i$ is the image of P_i by a rotation of angle $\pi/2^i$, saying that $\hat{P}_i$ belongs to some set is equivalent to saying that P_i belongs to the image of this set by the inverse rotation. In our case, this means *in fine* that P_i has to belong to the image by a rotation of angle $-\pi/2^i$ of a segment $[P_{i+1}\ \bar{P}_{i+1}]$ such that P_{i+1} has its second coordinate nonnegative.

We can now specialize this result to $i = k - 1$. Recall that P_k is known to belong to the line AB. According to the above discussion, we have first to restrict this set to its points with a nonnegative β_k, which gives the half line $[AB$. Taking the union of all segments $[P_k\ \bar{P}_k]$ for all possible P_k's gives the region bounded by half lines $[AB$ and $[AB'$. Taking finally the image of this set by a rotation of angle $-\pi/2^{k-1}$, we find that P_{k-1} has to belong to the region bounded by half lines $[BA$ and $[BC$.

We can now iterate this procedure and describe the set of points P_{k-2}, P_{k-3}, etc. Indeed, using exactly the same reasoning, we find that the set of points P_{i-1} can be deduced from the set of points P_i with a three-step procedure:

a. Restrict the set of points P_i to those with a nonnegative β_i coordinate.

b. Consider the union of segments $[P_i\ \bar{P}_i]$ where P_i belongs to the above restricted set, i.e. add for each point (α_i, β_i) the set of points (α_i, x) for all x ranging from $-\beta_i$ to β_i.

c. Rotate this union counterclockwise around the origin with an angle equal to $\pi/2^i$ to find the set of points P_{i-1}.

In the case of our example with $k = 3$, we have already shown that the set $\{P_k\} = \{P_3\} = [AB$ and $\{P_{k-1}\} = \{P_2\}$ is the region bounded by $[BA \cup [BC$. Going on with the procedure described above, we readily find that $\{P_{k-2}\} = \{P_1\}$ is the region bounded by the polygonal line $[ABCDE]$ while $\{P_{k-3}\} = \{P_0\}$ is the complete octagon $ABCDED'C'B'A$, which is the expected result (see Figure 2 for the corresponding pictures). It is not difficult to see that in the general case $\{P_{k-i}\}$ is a set bounded by 2^i consecutive sides of $\mathcal{P}_{2^k}$, which means we always end up with $\{P_0\}$ equal to the whole regular 2^k-sided polyhedron $\mathcal{P}_{2^k}$. This completes the proof since the set of points P_0 is the projection of $\mathcal{D}_k$ on the subspace of the two variables (α_0, β_0). □

This theorem allows us to derive quite easily a polyhedral approximation for $\mathcal{B}_2(1)$.

Corollary 2.1. *The projection of $\mathcal{D}_k$ on the subspace of its two variables (α_0, β_0) is a polyhedral approximation of $\mathcal{B}_2(1)$ with accuracy $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$.*

Proof. Straightforward application of Theorems 2.1 and 2.2. □

This approximation is much better than the previous one: we have here that $\epsilon \approx \frac{\pi^2}{2^{2k+1}}$, which means that dividing the accuracy by four can be achieved by increasing k by 1, which corresponds to adding 2 variables, 1 equality and 2 inequality constraints (compare to the

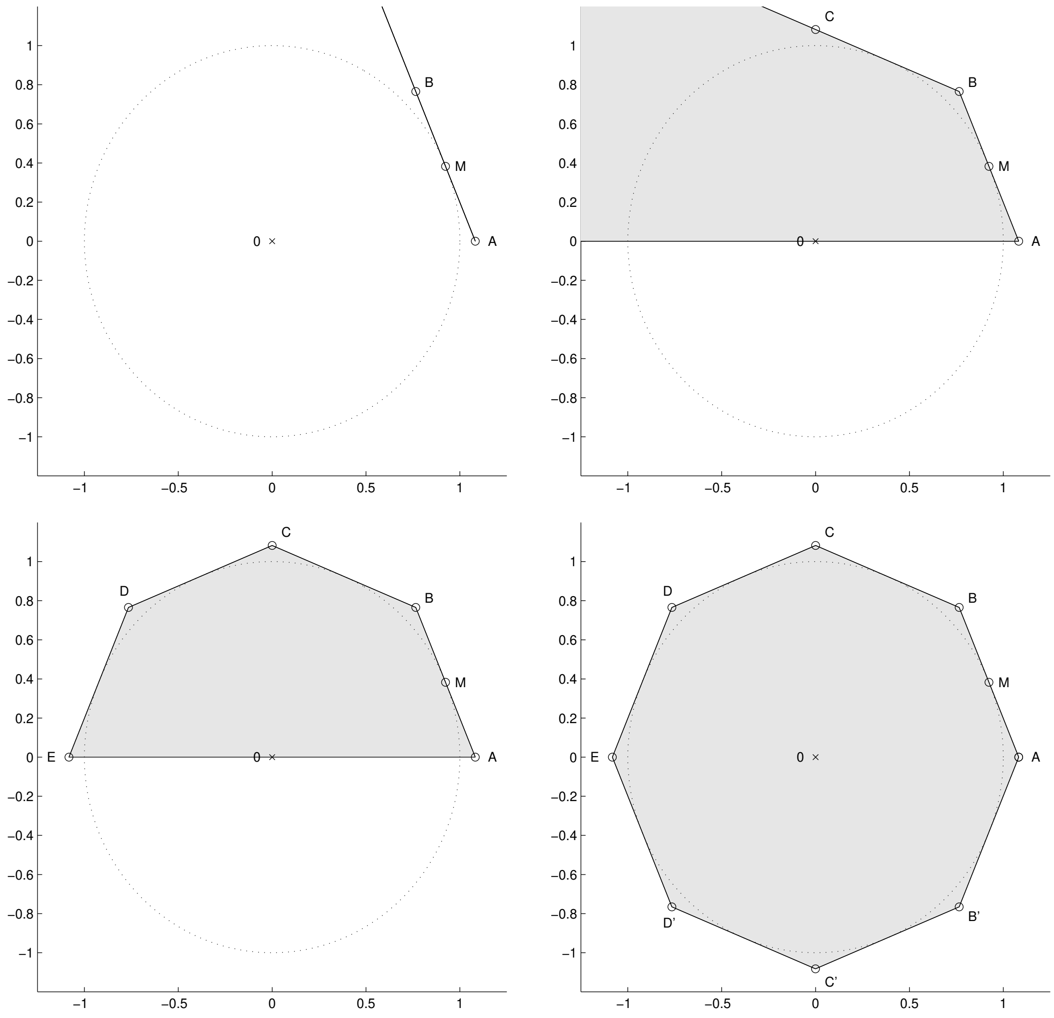

Figure 2: The sets of points P_3, P_2, P_1 and P_0 when $k = 3$.

previous situation which needed to double the number of inequalities to reach to same goal). For example, an accuracy of $\epsilon = 10^{-4}$ can be obtained with $k = 8$, i.e. with 16 inequalities, 9 equalities and 18 variables (as opposed to 223 inequalities with the previous approach).

We are now in position to convert this polyhedral approximation of $\mathcal{B}_2(1)$ into an approximation of $\mathbb{L}^2$. We define the set $\mathcal{L}_k \in \mathbb{R}^{2k+3}$ as

$$\left\{(r, \alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_k) \in \mathbb{R}^{2k+3} \mid \left\{ \begin{array}{rcrcr} \alpha_{i+1} & = & \alpha_i \cos \frac{\pi}{2^i} & + & \beta_i \sin \frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \beta_i \cos \frac{\pi}{2^i} & - & \alpha_i \sin \frac{\pi}{2^i} \\ -\beta_{i+1} & \leq & \beta_i \cos \frac{\pi}{2^i} & - & \alpha_i \sin \frac{\pi}{2^i} \\ r & = & \alpha_k \cos \frac{\pi}{2^k} & + & \beta_k \sin \frac{\pi}{2^k} \end{array} \right. \forall\ 0 \leq i < k \right\}.$$

Note the close resemblance between this set and $\mathcal{D}_k$, the only difference being the introduction of an additional variable r in the last equality constraint. This set $\mathcal{L}_k$ is our final polyhedral approximation of $\mathbb{L}^2$. Obviously, before we give proof of this fact, we need a measure of the quality of an approximation in the case of a second-order cone. This is the purpose of the next definition.

Definition 2.1. *A set $S \subseteq \mathbb{R}^{n+1}$ is said to be an ϵ-approximation of the second-order cone $\mathbb{L}^n$ if and only if we have*

$$\mathbb{L}^n \subseteq S \subseteq \mathbb{L}^n_\epsilon = \{(r, x) \in \mathbb{R} \times \mathbb{R}^n \mid \|x\| \leq (1+\epsilon) r\}$$

where $\mathbb{L}^n_\epsilon$ is an ϵ-relaxed second-order cone.

This definition extends our definition of ϵ-approximation for the unit disc $\mathcal{B}_2(1)$. The next theorem demonstrates how Corollary 2.1 on the accuracy of the polyhedral approximation $\mathcal{D}_k$ for $\mathcal{B}_2(1)$ can be converted into a result on the accuracy of $\mathcal{L}_k$ for $\mathbb{L}^2$.

Theorem 2.3. *The projection of $\mathcal{L}_k$ on the subspace of its three variables (r, α_0, β_0) is a polyhedral approximation of $\mathbb{L}^2$ with accuracy $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$.*

Proof. Assuming $r > 0$ for the moment, we first establish a link between $\mathcal{D}_k$ and $\mathcal{L}_k$. It is indeed straightforward to check using the corresponding definitions that the following equivalence holds

$$(r, \alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_k) \in \mathcal{L}_k \;\Leftrightarrow (\frac{\alpha_0}{r}, \ldots, \frac{\alpha_k}{r}, \frac{\beta_0}{r}, \ldots, \frac{\beta_k}{r}) \in \mathcal{D}_k\ , \tag{2.3}$$

since

$$\left\{ \begin{array}{rcrcl} \alpha_{i+1} & = & \alpha_i \cos\frac{\pi}{2^i} & + & \beta_i \sin\frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \beta_i \cos\frac{\pi}{2^i} & - & \alpha_i \sin\frac{\pi}{2^i} \\ -\beta_{i+1} & \leq & \beta_i \cos\frac{\pi}{2^i} & - & \alpha_i \sin\frac{\pi}{2^i} \\ r & = & \alpha_k \cos\frac{\pi}{2^k} & + & \beta_k \sin\frac{\pi}{2^k} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{rcrcl} \frac{\alpha_{i+1}}{r} & = & \frac{\alpha_i}{r} \cos\frac{\pi}{2^i} & + & \frac{\beta_i}{r} \sin\frac{\pi}{2^i} \\ \frac{\beta_{i+1}}{r} & \geq & \frac{\beta_i}{r} \cos\frac{\pi}{2^i} & - & \frac{\alpha_i}{r} \sin\frac{\pi}{2^i} \\ -\frac{\beta_{i+1}}{r} & \leq & \frac{\beta_i}{r} \cos\frac{\pi}{2^i} & - & \frac{\alpha_i}{r} \sin\frac{\pi}{2^i} \\ 1 & = & \frac{\alpha_k}{r} \cos\frac{\pi}{2^k} & + & \frac{\beta_k}{r} \sin\frac{\pi}{2^k} \end{array} \right.$$

which means that $\mathcal{L}_k$ is nothing more than the homogenized polyhedral cone corresponding to $\mathcal{D}_k$.

Let us now suppose $(r, x_1, x_2) \in \mathbb{L}^2$. Equivalence (2.2) implies $(\frac{x_1}{r}, \frac{x_2}{r}) \in \mathcal{B}_2(1)$, which in turn implies by Corollary 2.1 that there exists a vector $(\alpha, \beta) \in \mathbb{R}^{2k}$ such that $(\frac{x_1}{r}, \alpha, \frac{x_2}{r}, \beta)$ belongs to $\mathcal{D}_k$. Using the link (2.3), this last inclusion is equivalent to $(r, x_1, r\alpha, x_2, r\beta) \in \mathcal{L}_k$, which means that (r, α_0, β_0) belongs to the projection of $\mathcal{L}_k$ on the subspace (r, α_0, β_0). We have thus shown that this projection is a relaxation of $\mathbb{L}^2$, the first condition for it to be an ϵ-approximation of $\mathbb{L}^2$.

Supposing now (r, x_1, x_2) belongs to the projection of $\mathcal{L}_k$, there exists a vector $(\alpha, \beta) \in \mathbb{R}^{2k}$ such that $(r, x_1, \alpha, x_2, \beta) \in \mathcal{L}_k$. The equivalence (2.3) implies then that $(\frac{x_1}{r}, \frac{\alpha}{r}, \frac{x_2}{r}, \frac{\beta}{r}) \in \mathcal{D}_k$, which means that $(\frac{x_1}{r}, \frac{x_2}{r})$ belongs to the projection of $\mathcal{D}_k$ on its subspace (α_0, β_0). Using now Corollary 2.1, which states that this projection is an ϵ-approximation of $\mathcal{B}_2(1)$ with $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$, we can write that $\left\|(\frac{x_1}{r}, \frac{x_2}{r})\right\| \leq 1 + \epsilon$, which can be rewritten as $\|(x_1, x_2)\| \leq (1+\epsilon) r$, which is the exactly the second condition for this projection to be an ϵ-approximation of $\mathbb{L}^2$.

The last task we have to accomplish is to check what happens in the case where $r \leq 0$. Suppose $(r, x_1, \alpha, x_2, \beta) \in \mathcal{L}_k$. Looking at the definition of $\mathcal{L}_k$, and using the same reasoning as in the proof of Theorem 2.2, it is straightforward to show that the variables α_0 and β_0 can only be equal to 0 when $r = 0$, and that they cannot satisfy the constraints if $r < 0$ (i.e. in the first case the set $\{P_0\}$ is equal to $\{(0, 0)\}$ while in the second case $\{P_0\} = \emptyset$). Since this is also the situation of the second-order cone $\mathbb{L}^2$, our approximation is exact when $r \leq 0$, and we can conclude that the projection of $\mathcal{L}_k$ on the subspace of its three variables (r, α_0, β_0) is an ϵ-approximation of the three dimensional second-order cone $\mathbb{L}^2$ with $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$. □

2.5 Reducing the approximation

Our polyhedral approximation $\mathcal{L}_k$ features $2k+3$ variables, $2k$ linear inequalities and $k+1$ linear equalities. It is possible to reduce these numbers by pivoting out a certain number of variables. Namely, using the set of constraints $\alpha_{i+1} = \alpha_i \cos \frac{\pi}{2^i} + \beta_i \sin \frac{\pi}{2^i}$ for $0 \le i < k$, we can replace α_k by a linear combination of α_{k-1} and β_{k-1}, then replace α_{k-1} by a linear combination of α_{k-2} and β_{k-2}, and so on until all variables α_i have been replaced except α_0 (which cannot and should not be pivoted out since it belongs to the projected approximation). The last equality $r = \alpha_k \cos \frac{\pi}{2^k} + \beta_k \sin \frac{\pi}{2^k}$ can also be used to pivot out β_k.

The resulting polyhedron has then $k+2$ variables $(r, \alpha_0, \beta_0, \dots, \beta_{k-1})$, $2k$ linear inequalities and no linear equality. However, it should be noted that the constraint matrix describing the reduced polyhedron is denser than in the original approximation, i.e. it contains many more nonzero elements, as depicted on Figure 3 in the case $k = 15$ (which also mentions the number of nonzero elements in each case).

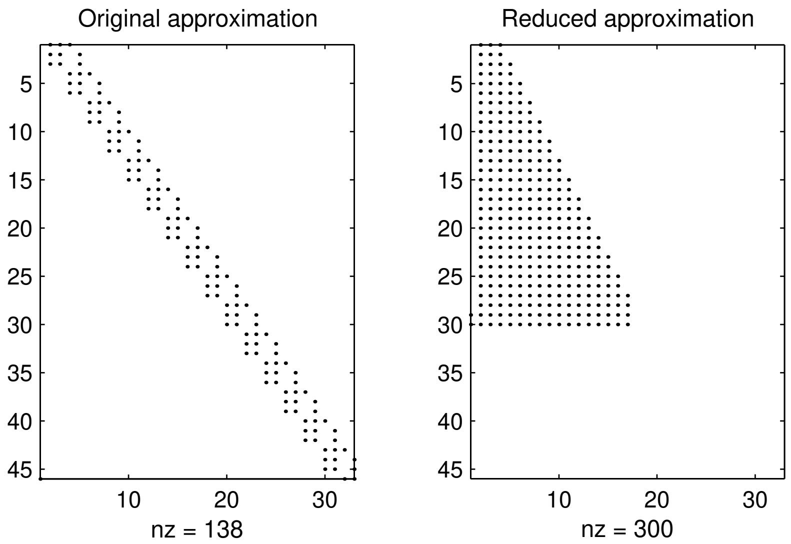

Figure 3: Constraint matrices for $\mathcal{L}_{15}$ and its reduced variant.

This denser constraint matrix has of course a negative impact on the efficiency of the algorithm used to solve the approximation problems, so that computational experiments are needed to decide whether this is enough to counterbalance the advantage of a reduced number of equalities and variables Indeed, preliminary testing on a few problems representative of the ones we are going to consider in Section 3 led us to the conclusion that pivoting out the variables is beneficial, leading roughly to a 20% reduction of computing times.

Another interesting remark can be made when we have to approximate a second-order cone whose components are restricted to be nonnegative. Namely, if we know beforehand that x_1 and x_2 cannot be negative, the polyhedral approximation of $\mathbb{L}^2$ can be reduced. Indeed, looking back at the proof of Theorem 2.2, we see that the set of points P_2 is bounded by 2^{k-2} consecutive sides of the regular 2^k-sided polyhedron (see for example the set $\{P_2\}$ depicted in Figure 2). Combining this with the restriction that α_2 and β_2 are nonnegative, we have that the set $\{P_2\}$ is exactly equal to the restriction of $\mathcal{P}_{2^k}$ to the positive orthant, and is thus a valid ϵ-approximation of $\mathbb{L}^2$ on this positive orthant with $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$.

This observation leads to the formulation of a reduced polyhedral approximation $\mathcal{L}'_k$ defined by

$$\left\{(r,\alpha_2,\ldots,\alpha_k,\beta_2,\ldots,\beta_k)\in\mathbb{R}^{2k-1} \mid \left\{\begin{array}{rcrcr} \alpha_{i+1} & = & \alpha_i\cos\frac{\pi}{2^i} & + & \beta_i\sin\frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \beta_i\cos\frac{\pi}{2^i} & - & \alpha_i\sin\frac{\pi}{2^i} \\ -\beta_{i+1} & \leq & \beta_i\cos\frac{\pi}{2^i} & - & \alpha_i\sin\frac{\pi}{2^i} \\ r & = & \alpha_k\cos\frac{\pi}{2^k} & + & \beta_k\sin\frac{\pi}{2^k} \end{array}\right. \forall\ 2\leq i<k\right\},$$

whose projection of the subspace of (r,α_2,β_2) approximates the nonnegative part $\mathbb{L}^2$. This approximation features $2k-1$ variables, $2k-4$ linear inequalities and $k-1$ linear equalities and can be reduced to k variables, $2k-4$ linear inequalities and no linear equality if we perform the pivoting described above.

At this stage, we would like to compare our approximation with the one presented in [BTN98]. Both of these feature the same accuracy $\epsilon=\cos(\frac{\pi}{2^k})^{-1}-1$ (with parameter ν in [BTN98] equal to $k-1$ in our setting). However, Ben-Tal and Nemirovski do not make explicit that the projection of their polyhedral approximation is equal to the regular 2^k-sided polyhedron in $\mathbb{R}^2$ and only prove the corresponding accuracy result.

Table 1 compares the sizes of the polyhedral approximations in three cases: the original approximation, the reduced approximation where variables α_i are pivoted out and the nonnegative approximation $\mathcal{L}'_k$ (also with variables α_i pivoted out).

Table 1: Comparison of our approximation $\mathcal{L}_k$ with [BTN98].

	Original		Reduced		Nonnegative	
	[BTN98]	$\mathcal{L}_k$	[BTN98]	$\mathcal{L}_k$	[BTN98]	$\mathcal{L}'_k$
Variables	$2k+3$	$2k+3$	$k+4$	$k+2$	$k+2$	k
Inequalities	$2k+4$	$2k$	$2k+4$	$2k$	$2k$	$2k-4$
Equalities	$k-1$	$k+1$	0	0	0	0

Our version uses 4 less inequality constraints in all three cases. It also features 2 more equality constraints in the original approximation, which turns out to be an advantage since it allows us to pivot out more variables in the reduced versions. Both the reduced and the nonnegative versions of $\mathcal{L}_k$ use 2 less variables than their counterparts in the original article of Ben-Tal and Nemirovski.

2.6 An approximation of $\mathbb{L}^n$

We are now going to use the decomposition presented in Section 2.2 and our polyhedral approximation $\mathcal{L}_k$ for $\mathbb{L}^2$ to build an approximation for $\mathbb{L}^n$. Recall that expression (2.1) decomposed $\mathbb{L}^n$ into $\lfloor n/2\rfloor$ three-dimensional second-order cones $\mathbb{L}^2$ and a single larger cone $\mathbb{L}^{\lceil\frac{n}{2}\rceil}$. Applying this decomposition recursively, we can decompose $\mathbb{L}^{\lceil\frac{n}{2}\rceil}$ into $\lfloor\lceil n/2\rceil/2\rfloor$ second-order cones $\mathbb{L}^2$ with a remaining larger cone $\mathbb{L}^{\lceil\lceil n/2\rceil/2\rceil}$, which can be again decomposed into $\lfloor\lceil\lceil n/2\rceil/2\rceil/2\rfloor$ cones $\mathbb{L}^2$, etc.

Calling q_k the number of three-dimensional second-order cones appearing in the decomposition at each stage of this procedure and r_k the corresponding size of the remaining cone, we have initially $q_0 = 0$, $r_0 = n$ and

$$q_k = \lfloor \frac{r_{k-1}}{2} \rfloor \text{ and } r_k = \lceil \frac{r_{k-1}}{2} \rceil \; \forall \; k > 0 \; .$$

Obviously, r_k is strictly decreasing and we must eventually end up with r_k equal to 2. Indeed, it is easy to see that $2^{i-1} < r_{k-1} \leq 2^i$ implies $2^{i-2} < r_k \leq 2^{i-1}$ and a simple recursive argument shows then that if $2^{m-1} < n \leq 2^m$ we have $2^{m-k-1} < r_k \leq 2^{m-k}$ and thus that $r_{m-1} = 2$. At this stage, the remaining second-order cone is $\mathbb{L}^2$, which we can add to the decomposition in the last stage with $q_m = 1$ to have $r_m = 0$. Our decomposition has thus in total m stages. We also showed in Section 2.2 that the total of $\mathbb{L}^2$ cones in the final decomposition is equal to $\sum_{i=1}^m q_i = n - 1$, and we also note for later use that $2^{m-k} < r_{k-1} \leq 2^{m-k+1}$ implies $2^{m-k-1} \leq q_k \leq 2^{m-k}$.

We ask ourselves now what happens if each of the second-order cones appearing in this decomposition is replaced by an ϵ-approximation. Namely, suppose each of the $\lfloor n/2 \rfloor$ second-order cones $\mathbb{L}^2$ in expression (2.1) is replaced by an $\epsilon^{(i)}$-approximation ($0 \leq i \leq \lfloor n/2 \rfloor$), while the remaining larger cone is replaced by an ϵ'-approximation. We end up with the set

$$\left\{ (r, x, y) \in \mathbb{R}_+ \times \mathbb{R}^{n + \lfloor \frac{n}{2} \rfloor} \mid (y_i, x_{2i-1}, x_{2i}) \in \mathbb{L}^2_{\epsilon^{(i)}}, \; 1 \leq i \leq \lfloor \frac{n}{2} \rfloor, \begin{cases} (r, y) \in \mathbb{L}^{\lceil n/2 \rceil}_{\epsilon'} & (n \text{ even}) \\ (r, y, x_n) \in \mathbb{L}^{\lceil n/2 \rceil}_{\epsilon'} & (n \text{ odd}) \end{cases} \right\},$$

whose constraints are equivalent to

$$x_{2i-1}^2 + x_{2i}^2 \leq (1 + \epsilon^{(i)})^2 y_i^2, \; 1 \leq i \leq \lfloor \frac{n}{2} \rfloor, \begin{cases} \sum_{i=1}^{\frac{n}{2}} y_i^2 & \leq (1 + \epsilon')^2 r^2 \; (n \text{ even}) \\ \sum_{i=1}^{\frac{n}{2}} y_i^2 + x_n^2 & \leq (1 + \epsilon')^2 r^2 \; (n \text{ odd}) \end{cases} \Bigg\} \; .$$

Ideally, we would like this decomposition to be an ϵ-approximation of $\mathbb{L}^n$. We already know that it is a relaxation of $\mathbb{L}^n$, since each approximation of $\mathbb{L}^2$ is itself a relaxation. We have thus to concentrate on the second condition defining an ϵ-approximation, $\|x\| \leq (1 + \epsilon) r$. Writing

$$\sum_{i=1}^{2\lfloor n/2 \rfloor} x_i^2 \leq \sum_{i=1}^{\lfloor n/2 \rfloor} (1 + \epsilon^{(i)})^2 y_i^2 \; ,$$

we would like to bound the quantity on the right hand-side. Unfortunately, we only know a bound on the sum of y_i^2's, which forces us to write

$$\sum_{i=1}^{2\lfloor n/2 \rfloor} x_i^2 \leq (1 + \max_i \; \epsilon^{(i)})^2 \sum_{i=1}^{\lfloor n/2 \rfloor} y_i^2 \; \Rightarrow \; \sum_{i=1}^n x_i^2 \leq (1 + \max_i \; \epsilon^{(i)})^2 (1 + \epsilon')^2 r^2 \; .$$

This shows that our decomposition is an approximation of $\mathbb{L}^n$ with accuracy $\epsilon = (1 + \max_i \; \epsilon_i)(1 + \epsilon')$. This immediately implies that there is no point in approximating with different accuracies the $\lfloor n/2 \rfloor$ small second-order cones $\mathbb{L}^2$ appearing in the decomposition, since only the largest of these accuracies has an influence on the resulting approximation for $\mathbb{L}^n$. Applying now our decomposition recursively to the remaining cone, and choosing at each stage k a unique accuracy ϵ_k for all the $\mathbb{L}^2$ cones, we find that $1 + \epsilon = \prod_{k=1}^m (1 + \epsilon_k)$,, i.e. that the final accuracy of our polyhedral approximation is the product of the accuracies chosen at each stage of the decomposition (note that, unlike the situation for a single stage, there is no reason here to choose all ϵ_k accuracies to be equal to each other).

2.7 Optimizing the approximation

The previous section has shown how to build a polyhedral approximation of $\mathbb{L}^n$ and how its quality depends on the accuracy of the approximations used at each stage of the decomposition. Our goal is here to optimize these quantities, i.e. given a target accuracy ϵ for $\mathbb{L}^n$, find the values of ϵ_k ($1 \leq k \leq m$) that lead to the smallest polyhedral approximation, i.e. the one with the smallest number of variables and constraints.

Let us suppose we use at stage k the approximation $\mathcal{L}_{u_k}$ with $u_k + 2$ variables and $2u_k$ linear inequalities (i.e. with variables α pivoted out of the formulation), which has an accuracy $\epsilon_k = \cos(\frac{\pi}{2^{u_k}})^{-1} - 1$. Recalling notation q_k for the number of cones $\mathbb{L}^2$ introduced at stage k of the decomposition, the final polyhedral approximation has thus an accuracy equal to $\prod_{k=1}^m \cos(\frac{\pi}{2^{u_k}})^{-1}$ with $2\sum_{k=1}^m q_k u_k$ inequalities and $n + \sum_{k=1}^m q_k u_k$ variables. Indeed, we have n original x_i variables and u_k additional variables for each of the q_k approximations at stage k, since the first two variables in these approximations are coming from the previous stage. We observe that the main quantity to be minimized is $\sum_{k=1}^m q_k u_k$ for both the number of variables and inequalities, which leads to the following optimization problem:

$$\sigma_{n,\epsilon} = \min_{u \in \mathbb{N}^m} \sum_{k=1}^m q_k u_k \quad \text{s.t.} \quad \prod_{k=1}^m \cos(\frac{\pi}{2^{u_k}})^{-1} \leq 1 + \epsilon \ . \tag{2.4}$$

A possible choice for variables u_k is to take them all equal. Plugging this unique value into the accuracy constraint, we readily find that u_k has to be equal to

$$u_k = \lceil \log_2 \left(\pi / \arccos(1+\epsilon)^{-1/m} \right) \rceil$$

and that when the dimension of the cone $\mathbb{L}^n$ (and thus m) tends to $+\infty$, we have $u_k = \mathcal{O}\left(\log \frac{m}{\epsilon}\right)$ and $\sigma_{n,\epsilon} = \mathcal{O}\left(n \log \frac{m}{\epsilon}\right)$.

This obviously does not lead to an optimal solution of (2.4). Indeed, since the number of approximations is decreasing as we move from one stage of the decomposition to the next, it is intuitively clear that trading a lower accuracy for first stages against a higher accuracy for the last stages will be beneficial, since the lowering of the number of variables and inequalities in the first stages will affect many more constraints than the increase of size for the last stages. This implies that the components u_k of any optimal solution of (2.4) will have to be in increasing order.

Finding a closed form optimal solution of (2.4) does not appear to be possible, but we can find a good suboptimal solution using some approximations. We first introduce variables v_k such that $v_k = 4^{-u_k} \Leftrightarrow u_k = -\log_4 v_k$ and rewrite problem (2.4) as

$$\sigma_{n,\epsilon} = \min_{v \in \mathbb{R}^m} -\log_4 \prod_{k=1}^m v_k^{q_k} \quad \text{s.t.} \quad \sum_{k=1}^m \log(\cos(\pi\sqrt{v_k})^{-1}) \leq \log(1+\epsilon) \ .$$

Since $u_k \geq 2$, we have $\pi\sqrt{v_k} \leq \frac{\pi}{4}$ and we can use the easily proven[2] inequality

$$\log(\cos(x)^{-1}) \leq (\frac{3x}{4})^2$$

valid for all $0 \leq x \leq \frac{\pi}{4}$ to write

$$\sigma_{n,\epsilon} = -\max_{v \in \mathbb{R}^m} \log_4 \prod_{k=1}^m v_k^{q_k} \quad \text{s.t.} \quad \sum_{k=1}^m v_k \leq \frac{16}{9}\pi^{-2}\log(1+\epsilon) = K(\epsilon) \ ,$$

[2] This inequality can be easily checked by plotting the graphs of its two sides on the interval $[0\ \frac{\pi}{4}]$.

which is thus a restriction of our original problem. It amounts to maximizing a product of variables whose sum is bounded, a problem whose optimality conditions are well-known. In our case, they can written as

$$\frac{v_1}{q_1} = \frac{v_2}{q_2} = \ldots \frac{v_m}{q_m} = \frac{\sum_{k=1}^m v_k}{\sum_{k=1}^m q_k} = \frac{K(\epsilon)}{n-1} \Rightarrow v_k = \frac{q_k}{n-1} K(\epsilon) \Leftrightarrow u_k = \log_4 \frac{n-1}{q_k} - \log_4 K(\epsilon) \; .$$

However u_k must be integer, so that we have to degrade this solution further and round it towards a larger integer. Using that fact that $n-1 \leq 2^m$ and $q_k \geq 2^{m-k-1}$, we have

$$\frac{n-1}{q_k} \leq \frac{2^m}{2^{m-k-1}} = 2^{k+1} \Rightarrow \log_4 \frac{n-1}{q_k} \leq \log_4 2^{k+1} = \frac{k+1}{2} \; ,$$

so that we can take $u_k = \lceil \frac{k+1}{2} \rceil - \lfloor \log_4 K(\epsilon) \rfloor$ as our suboptimal integer solution for (2.4).

Let us plug now these values into the objective function $\sigma_{n,\epsilon}$: we find

$$\begin{aligned}
\sum_{k=1}^m q_k u_k &= \sum_{k=1}^m q_k \lceil \frac{k+1}{2} \rceil - \sum_{k=1}^m q_k \lfloor \log_4 K(\epsilon) \rfloor \\
&\leq \sum_{k=1}^m q_k (\frac{k}{2} + 1) - (n-1) \lfloor \log_4 K(\epsilon) \rfloor \quad (\text{using} \sum_{k=1}^m q_k = n-1) \\
&\leq \sum_{k=1}^m 2^{m-k} (\frac{k}{2} + 1) - (n-1) \lfloor \log_4 K(\epsilon) \rfloor \quad (\text{using } q_k \leq 2^{m-k}) \\
&\leq 2^{m+1} - \frac{m}{2} - 2 - (n-1) \lfloor \log_4 K(\epsilon) \rfloor \\
&\leq 4(n-1) - (n-1) \lfloor \log_4 K(\epsilon) \rfloor \quad (\text{using } 2^{m-1} \leq n-1) \\
&\leq (n-1) \lceil 4 - \log_4 K(\epsilon) \rceil = (n-1) \lceil 4 - \log_4 \frac{16}{9} + \log_4 \pi^2 - \log_4 \log(1+\epsilon) \rceil \\
&\leq (n-1) \lceil 5.3 - \log_4 \log(1+\epsilon) \rceil
\end{aligned}$$

(where we have used at the fourth line the fact that $\sum_{k=1}^m 2^{m-k}(\frac{k}{2}+1) = 2^{m+1} - \frac{m}{2} - 2$, which is easily proved recursively). We can wrap this result into the following theorem:

Theorem 2.4. *For every $\epsilon < \frac{1}{2}$, there exists a polyhedron with no more than*

$$2 + (n-1) \lceil 5.3 - \log_4 \log(1+\epsilon) \rceil = \mathcal{O}\big(n \log \frac{1}{\epsilon}\big) \textit{ variables}$$

and

$$2 + 2(n-1) \lceil 4.3 - \log_4 \log(1+\epsilon) \rceil = \mathcal{O}\big(n \log \frac{1}{\epsilon}\big) \textit{ inequalities}$$

whose projection on a certain subspace of $n+1$ variables is an ϵ-approximation of the second-order cone $\mathbb{L}^n \subseteq \mathbb{R}^{n+1}$.

Proof. This a consequence of the previous derivation, which showed that choosing $u_k = \lceil \frac{k+1}{2} \rceil - \lfloor \log_4 K(\epsilon) \rfloor$ lead to an ϵ-approximation of $\mathbb{L}^n$ with $n + \sigma_{n,\epsilon}$ variables and $2\sigma_{n,\epsilon}$ linear inequalities, with $\sigma_{n,\epsilon} = (n-1) \lceil 5.3 - \log_4 \log(1+\epsilon) \rceil$. However, the size of this polyhedron can be further reduced using $\mathcal{L}'_k$, the polyhedral approximation of the nonnegative part of $\mathbb{L}^2$. Indeed, looking at the decomposition (2.1), we see that all the y variables used in the second stage of the decomposition are guaranteed to be nonnegative, since we have in our approximation $(y_i, x_{2i-1}, x_{2i}) \in \mathcal{L}_{u_1}$ which implies $y_i \geq 0$. This means that we can use for the second stage and the following our reduced approximation $\mathcal{L}'_{u_k}$, known to be valid when its first two variables are restricted to the nonnegative orthant $\mathbb{L}^2$, which uses 2 less variables and 4 less inequalities per cone. Since there are $\frac{n}{2}$ cones in the first stage of the decomposition and $n-1$ cone in total, we can use $\frac{n}{2}-1$ reduced approximations $\mathcal{L}'$, which give us a total saving of $n-2$ variables and $2n-4$ constraints[3]. Combining this with the value of $\sigma_{n,\epsilon}$, we find that our approximation has $2+(n-1)\lceil 5.3-\log_4 \log(1+\epsilon)\rceil$ variables and $2+\lceil 4.3-\log_4 \log(1+\epsilon)\rceil$ inequalities.

We also have to prove the asymptotic behaviour of $\sigma_{n,\epsilon}$ when n tends to infinity. Indeed, we have $\log(1+\epsilon) \geq \frac{\epsilon}{2}$ when $\epsilon < \frac{1}{2}$, which implies $-\log_4 \log(1+\epsilon) \leq \log_4 \frac{1}{\epsilon}$. This leads to $\lceil 5.3 - \log_4 \log(1+\epsilon)\rceil = \mathcal{O}\big(\log \frac{1}{\epsilon}\big)$, which is enough to prove the theorem. □

This result is better than the one we previously obtained choosing all u_k's equal to each other: indeed, we had in that case $\sigma_{n,\epsilon} = \mathcal{O}\big(n \log \frac{m}{\epsilon}\big) = \mathcal{O}\big(n \log \frac{\log n}{\epsilon}\big)$ while we have here $\mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$. For a fixed accuracy ϵ, our first choice translates into $\sigma_{n,\epsilon} = \mathcal{O}\left(n \log \log n\right)$ when n tends to infinity while our optimized choice of u_k's leads to $\sigma_{n,\epsilon} = \mathcal{O}\left(n\right)$, which is better. We note however that if we fix n and let ϵ tends to 0, the asymptotic behaviour is the same in both cases , namely we have $\sigma_{n,\epsilon} = \mathcal{O}\big(\log \frac{1}{\epsilon}\big)$.

Ben-Tal and Nemirovski achieve essentially the same result in [BTN98], albeit in the special case when n is a power of two. Our proof has the additional advantage of providing a closed form for the parameters u_k as well as for the total size of the polyhedral approximation, for all values of n. They also prove that the number of inequalities of a ϵ-approximation of $\mathbb{L}^n$ must be greater that $\mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$, i.e. that the order of the result of Theorem 2.4 is not improvable.

2.8 An approximation of second-order cones optimization

The previous sections have proven the existence of a polyhedral approximation of the second-order cone with a moderate size (growing linearly with the dimension of the cone and the logarithm of the accuracy). However, we have to point out that these polyhedrons are not strictly speaking approximations of the second-order cone: more precisely, it is their projection on a certain subspace than is an ϵ-approximation of $\mathbb{L}^n$.

This does not pose any problem when trying to approximate a second-order cone optimization problem with linear optimization. Let us suppose we want to approximate problem (1.3), which we recall here for convenience,

$$\inf\ c^T x \quad \text{s.t.} \quad Ax = b \text{ and } x^k \in \mathbb{L}^{n_k}\ \forall k = 1, 2, \ldots, r \tag{1.3}$$

with ϵ-approximations of the second order cones $\mathbb{L}^{n_k}$. Theorem 2.4 implies the existence of a polyhedron

$$\mathcal{Q}_k = \Big\{ (x^k, y^k) \in \mathbb{R}^{n_k+1} \times \mathbb{R}^{\mathcal{O}\left(n_k \log \frac{1}{\epsilon}\right)} \mid A_k (x^k, y^k)^T \geq 0 \Big\}$$

with

$$A_k \in \mathbb{R}^{\mathcal{O}\left(n_k \log \frac{1}{\epsilon}\right) \times \mathcal{O}\left(n_k \log \frac{1}{\epsilon}\right)} ,$$

[3]This reasoning was made for an even n. In the case of an odd n, we have one cone in the decomposition for which only the first variable is known to be nonnegative. It is possible to show that there exists a polyhedral approximation adapted to this situation that uses 1 less variable and 2 less inequalities than the regular approximation, which allows us to write exactly the same results as for an even n.

whose projection on the subspace (r, x) is an ϵ-approximation of $\mathbb{L}^{n_k}$, which allows us to write the following linear optimization problem[4]

$$\min\ c^T x \quad \text{s.t.} \quad Ax = b \text{ and } A_k(x^k, y^k)^T \geq 0\ \forall k = 1, 2, \ldots, r\ . \tag{2.5}$$

We note that the fact that our approximations are projections is handled in a seamless way by this formulation: the only difference with the use of a direct approximation of the cones $\mathbb{L}^{n_k}$ is the addition of the auxiliary variables y^k to the formulation. This problem features $\sum_{k=1}^r \mathcal{O}\big(n_k \log \frac{1}{\epsilon}\big) = \mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$ variables, m equality constraints and $\sum_{k=1}^r \mathcal{O}\big(n_k \log \frac{1}{\epsilon}\big) = m + \mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$ homogeneous inequality constraints. We also point out as a minor drawback of this formulation the fact that it involves irrational coefficients, namely the quantities $\sin \frac{\pi}{2^i}$ and $\cos \frac{\pi}{2^i}$ occurring in the definition of $\mathcal{L}_k$. However, it rational coefficients are really needed (for example if one wants to work with a complexity model based on exact arithmetic), it is possible to replace those quantities with rational approximations while keeping an essentially equivalent accuracy for the resulting polyhedral approximation, i.e. featuring the same asymptotic behaviour.

To conclude this section, we are going to compare the algorithmic complexity of solving problem (1.3) either directly or using our polyhedral approximation. The best complexity obtained so far[5] for solving a linear program with v variables up to accuracy ϵ is $\mathcal{O}\big(v^{3.5} \log(\frac{1}{\epsilon})\big)$ arithmetic operations (using for example a short-step path-following method, see e.g. [NN94]). In our case, assuming we solve the approximate problem (2.5) up to the same accuracy that the one used to approximate the second-order cones, this leads to a complexity equal to $\mathcal{O}\big(n^{3.5} \log(\frac{1}{\epsilon})^{4.5}\big)$.

On the other hand, solving problem (1.3) can be done using $\mathcal{O}\big(\sqrt{r}\, n^3 \log(\frac{1}{\epsilon})\big)$ arithmetic operations, using for example a potential reduction approach, see e.g. [LVBL98]. If $r = \mathcal{O}(1)$, i.e. if the number of cones used in the formulation is bounded, the second complexity is better, both if $n \to +\infty$ or $\epsilon \to 0$. However, if $r = \mathcal{O}(n)$, which means that the dimension of the cones used in the formulation is bounded, both complexity become equivalent from the point of view of the dimension n, but the second one is still better when letting the accuracy tend to 0. We conclude that the direct solving of (1.3) as a second-order cone problem is superior from the point of view of algorithmic complexity. The purpose of the second part of this article will be to test whether this claim is also valid for computational experiments.

2.9 Accuracy of the approximation

The linearizing scheme for second-order cone optimization presented in the previous section is based on a polyhedral approximation whose accuracy is guaranteed in the sense of Definition 2.1. It is important to realize that this bound on the accuracy of the approximation does not imply a bound on the accuracy of the solutions (or the objective value) of the approximated problem.

Indeed, let us consider the following set:

$$\big\{(r, x_1, x_2) \in \mathbb{R}^3 \mid r - x_2 = \frac{1}{2} \text{ and } (r, x_1, x_2) \in \mathbb{L}^2\big\}\ .$$

[4] We could replace the inf of problem (1.3) by a min since it is well-known that linear optimization problems always attain their optimal objectives.

[5] Using standard linear algebra and without partial updating.

This set can be seen as the feasible region of a second-order cone problem. Using the fact that

$$x_1^2 + x_2^2 \leq r^2 \Leftrightarrow x_1^2 + x_2^2 \leq (x_2 + \frac{1}{2})^2 \Leftrightarrow x_1^2 - \frac{1}{4} \leq x_2 \ ,$$

we find that the projection of this set on the subspace (x_1, x_2) is the epigraph of the parabola $x \mapsto x^2 - \frac{1}{4}$. Let us now replace $\mathbb{L}^2$ by the polyhedral approximation $\mathcal{L}_k$. Since the resulting set will be polyhedral, its projection on the subspace (x_1, x_2) will also be polyhedral, and we can deduce without difficulties that it is the epigraph of a piecewise linear function, as shown by Figure 4 (depicting the cases $k = 1$, 2, 3 and 4).

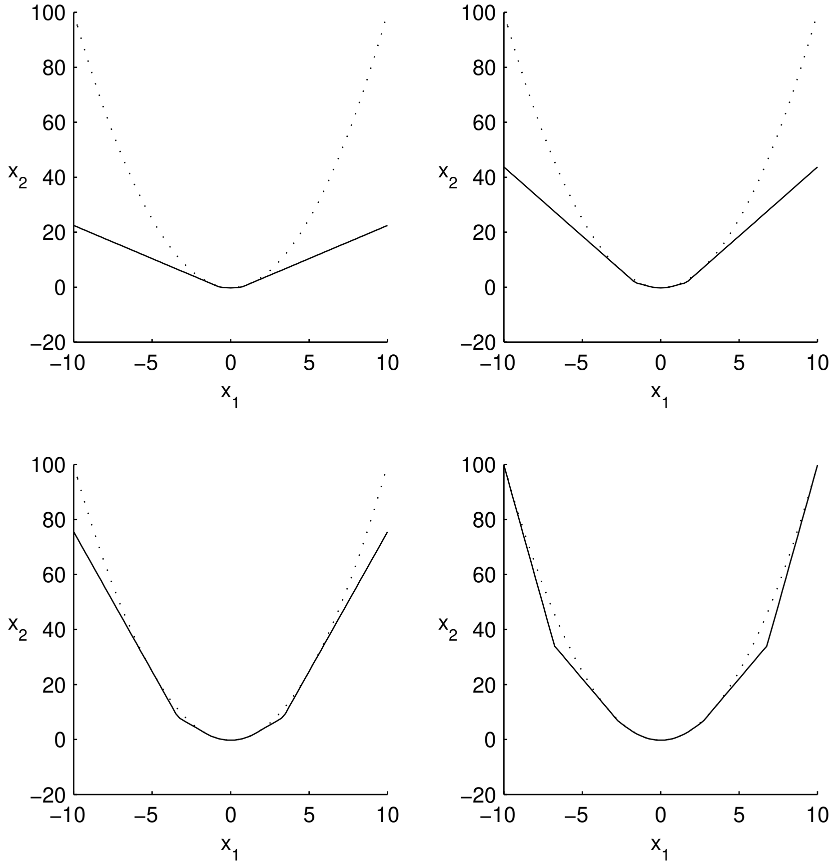

Figure 4: Linear approximation of a parabola using $\mathcal{L}_k$ for $k = 1, 2, 3, 4$.

Because a polyhedron has a finite number of vertices, this piecewise linear function must have a finite number of segments. Considering the rightmost piece, i.e. whose x_1 values span an interval of the type $[\alpha + \infty[$, it is obvious that it cannot approximate the parabola with a guaranteed accuracy. Indeed, the difference between the approximation and the parabola grows quadratically on this segment, which means that even the ratio of variable x_2 between the parabola and its linear approximation is not bounded.

Let us now consider the following parameterized family of second-order cone optimization problems

$$\min\ x_2 \quad \text{s.t.} \quad x_1 = \lambda,\ r - x_2 = \frac{1}{2} \text{ and } (r, x_1, x_2) \in \mathbb{L}^2 \ , \qquad (\text{PB}_\lambda)$$

which is using the same feasible set as above with an additional constraint fixing the variable x_1 to λ. Denoting the optimal objective value of (PB_λ) by $p^*(\lambda)$, we have in light of the previous discussion that $p^*(\lambda) = \lambda^2 - \frac{1}{4}$. However, we also showed that the optimal objective value $p_k^*(\lambda)$ of the approximated problem

$$\min\ x_2 \quad \text{s.t.} \quad x_1 = \lambda,\ r - x_2 = \frac{1}{2} \text{ and } (r, x_1, x_2, y) \in \mathcal{L}_k$$

must be a piecewise linear function of λ with a finite number of segments. Indeed, simple computations[6] show that the endpoints of these segments occur for

$$\lambda = \frac{\sin(i\theta)}{2\cos\frac{\theta}{2} - 2\cos(i\theta)} \text{ for } i = 1, 2, \ldots, 2^k - 1 \text{ with } \theta = \frac{\pi}{2^{k-1}}\ ,$$

which shows that $p^*(\lambda)$ is linear as soon as $\lambda \geq \frac{\sin\theta}{2\cos\frac{\theta}{2} - 2\cos\theta}$.

The discrepancy between the real optimum $p^*(\lambda)$ and the approximated optimum $p_k^*(\lambda)$ is thus unbounded when λ goes to infinity. Moreover, we have that the relative accuracy of $p_k^*(\lambda)$ tends to 1, the worst possible value, i.e.

$$\frac{p^*(\lambda) - p_k^*(\lambda)}{p^*(\lambda)} \to 1\ .$$

Another interesting feature of this small example is that performing a complete parametric analysis for parameter λ ranging from $-\infty$ to $+\infty$ would lead to $2^k - 1$ different break points.

We conclude that we cannot give an *a priori* bound on the accuracy of the optimal objective value of the linear approximation of a second-order cone optimization problem (this remark is also valid for accuracy of the optimal solution itself, since we have in our example (PB_λ) that the optimal value of x_2 is equal to p^*).

3 Computational experiments

In this section, we present computational experiments with an implementation of the linearizing scheme for second-order cone optimization we have just described.

3.1 Implementation

The computer used to conduct those experiments is an Intel 500 MHz Pentium III with 128 megabytes of memory. We chose to use the MATLAB programming environment, developed by The MathWorks, for the following reasons:

- MATLAB is a flexible and modular environment for technical computing, two very important characteristics when developing research code. Although MATLAB may be somehow slower than a pure C or FORTRAN approach, we think that this loss of performance is more than compensated by the ease of development (especially from the point of view of graphic capabilities and debugging). Moreover, the critical (i.e. time consuming) parts of the algorithms can be coded separately in C or FORTRAN and

[6]Simply observe that the extremal rays of $\mathcal{L}_k$ obey to the relation $x_2 = x_1 \tan i\theta$ with $i = 1, 2, \ldots, 2^k$ and $\theta = \pi/2^{k-1}$.

used in MATLAB via MEX files (this is the approach taken by the solvers we mention below), which allows a well designed MATLAB program to be nearly as efficient as an equivalent pure C or FORTRAN program.

- ◇ Efficient interior-point solvers are available on the MATLAB platform. Indeed, we used in our experiments

  - The MOSEK optmization toolbox for MATLAB by EKA Consulting ApS [AA99, ART00], a full-featured optimization package including a simplex solver and primal-dual interior-point solvers for linear optimization, convex linearly and quadratically constrained optimization, second-order cone optimization, linear least square problems, linear l_1 and l_∞-norm optimization and geometric and entropy optimization. When compared with the standard optimization toolbox from MATLAB, MOSEK is particularly efficient on large-scale and sparse problems. MOSEK can be downloaded for research and evaluation purposes at `http://www.mosek.com`.
  - SeDuMi by Jos Sturm [Stu99b], another primal-dual interior-point solver which is able to handle linear, second-order cone and semidefinite optimization problems. SeDuMi is designed to take into account sparsity and complex values, and has the advantage of dealing with the very important class of semidefinite optimization, but is a little more restrictive than MOSEK concerning the input format, since problems must be entered in the standard conic form (1.3). SeDuMi can be downloaded at `http://www.unimaas.nl/~sturm/`.

The main routines we implemented are the following:

- ◇ `PolySOC2(k)` generates the polyhedron $\mathcal{L}_k$ with accuracy $\epsilon_k = \cos(\frac{\pi}{2^{u_k}})^{-1} - 1$. Variables α_i are pivoted out, so that this routine returns a polyhedron with $k+2$ variables and $2k$ inequalities. An optional parameter is available to use the reduced approximation $\mathcal{L}'_k$, valid on the nonnegative restriction of $\mathbb{L}^2$.

- ◇ `Steps(q, e)` computes the optimal choice for the size of the cones at each stage of the decomposition of $\mathbb{L}^n$. Indeed, `q` contains our vector q (i.e. the number of cones at each stage) and `e` is the target accuracy ϵ.

- ◇ `PolySOCN(n, e)` generates a e-approximation of $\mathbb{L}^n$. It uses the output of `PolySOC2` and the optimal sizes for the cones computed by `Steps`.

- ◇ `PolySOCLP(p, e)` linearizes the second-order cone optimization problem `p`, replacing each second-order cone constraint with a polyhedral e-approximation using `PolySOCN` and outputting a linear optimization problem.

The procedure `Steps` we implemented features some improvements when compared with the theory we presented in the previous section. Indeed, Theorem 2.4 shows that the choice $u_k = \lceil \frac{k+1}{2} \rceil - \lfloor \log_4 K(\epsilon) \rfloor$ leads to a polyhedron of size $\mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$, but is not optimal for two reasons:

- ◇ We approximated the formula giving the accuracy of the approximation to derive u_k (namely, we used $\log(\cos(x)^{-1}) \leq (\frac{3x}{4})^2$).

- ◇ The optimal solution for this approximated accuracy was not guaranteed to be integer and had to be rounded to the smallest greater integer.

However, one can easily improve this choice in practice as follows. Let us suppose theory predicts some optimal values v_k for the sizes of the cones at stage k, which have to be rounded to $\lceil v_k \rceil$. Because of this rounding, the actual accuracy of the approximation will be much better than our target ϵ. Recalling now that this accuracy is equal to $(1+\epsilon_1)(1+\epsilon')$, where ϵ_1 is the accuracy of the cones in the first stage, and is thus equal to $\cos(\frac{\pi}{2\lceil v_1 \rceil})^{-1} - 1$, and ϵ' is the accuracy from the cone modelled by all the remaining stages $\mathbb{L}^{\lceil n/2 \rceil}$, we can compute an upper bound for ϵ', according to

$$(1+\epsilon_1)(1+\epsilon') \leq \epsilon \Leftrightarrow \epsilon' \leq \frac{\epsilon}{1+\epsilon_1} - 1$$

which will be better (i.e. higher) than in the theoretical derivation since it takes into account the exact accuracy ϵ_1 of the first stage, rounding included. We can now apply this procedure to the second stage, i.e. computing a theoretical value for ϵ_2 and an upper bound on the accuracy of $\mathbb{L}^{\lceil \lceil n/2 \rceil /2 \rceil}$, and so on, obtaining in the end a smaller polyhedral approximation, since the required accuracies for the cones at every stage (except the first one) are higher and hence need less constraints and variables.

Still, this improved rounding does not address the first reason why our u_k's are not optimal, the fact that we do not optimize the actual formula for the accuracy. Since it seems impossible to deal with it in closed form, we implemented a dynamic programming approach to optimize it. This algorithm uses the theoretical suboptimal solution described above (including the improved rounding procedure) to provide bounds on the optimal solution and therefore reduce the computing time.

Figure 5 presents the graphs of the size (measured by $\sigma_{n,\epsilon}$) of our approximations in two situations: fixed dimension ($n = 50, 200, 800$) with accuracy ranging from 10^{-1} to 10^{-8} and fixed accuracy ($\epsilon = 10^{-2}, 10^{-5}, 10^{-8}$) with dimension ranging from 10 to 1000. The asymptotic behaviour of $\sigma_{n,\epsilon}$ is very clear on these graphs: we have a linear increase when n tends to $+\infty$ for a fixed accuracy and a logarithmic increase when ϵ tends to 0 for a fixed dimension (since the first graph has a logarithmic scale of abscissas).

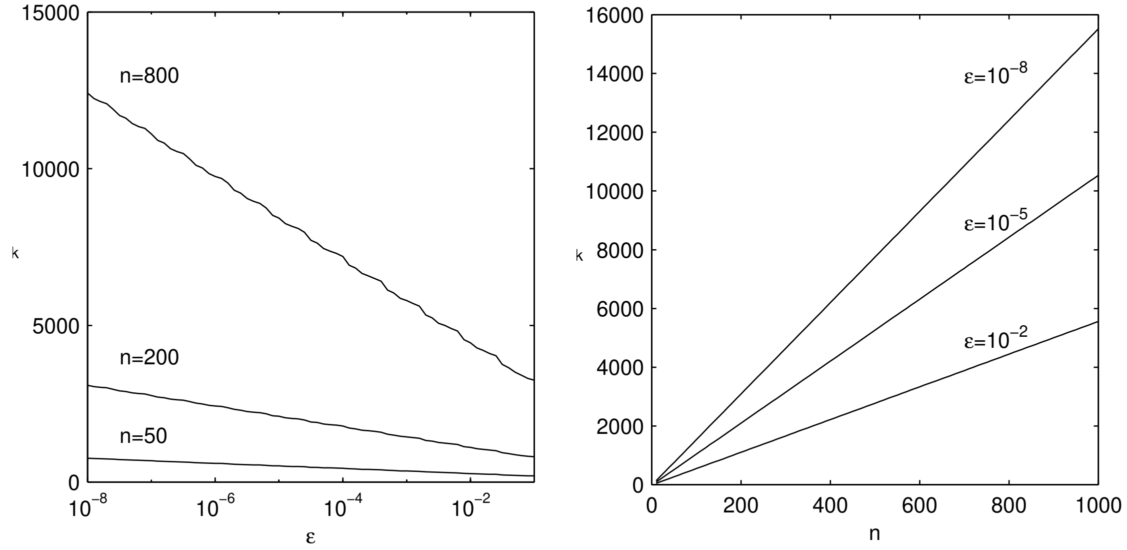

Figure 5: Size of the optimal approximation versus accuracy (left) and dimension (right).

Finally, in order to give an idea of the efficiency of our improved rounding procedure and dynamic programming resolution, we provide in Table 2 the value of $\sigma_{n,\epsilon}$ for different strategies, using a target accuracy $\epsilon = 10^{-8}$.

Table 2: Different approaches to optimize the size of a 10^{-8}-approximation of $\mathbb{L}^n$

n	Theory	All equal (rounded)	Theory (rounded)	Dynamic programming
10	171	139	141	139
100	1881	1584	1552	1537
1000	18981	15987	15688	15522
10000	189981	160140	157063	155392

The first column represents the theoretical value $\sigma_{n,\epsilon} = (n-1)\lceil 5.3 - \log_4 \log(1+\epsilon)\rceil$, the second column describes the choice of all u_k's, equal to each other, albeit using the improved rounding procedure presented above, the third column reports the choice of the theoretical value for u_k, this time with the improved rounding procedure, and the last column gives the true optimal value via our dynamic programming approach. We observe that our iterative rounding procedure improves the theoretical value of $\sigma_{n,\epsilon}$ in a noticeable way, lowering it by approximately 15%. The differences between the last three columns are less important, the dynamic programming approach giving a few additional percents of decrease in the size of the approximation.

3.2 Truss-topology design

We first tested our linearizing scheme for second-order cone optimization on a series of truss topology design problems. A *truss* is a structure composed of elastic bars connecting a set of nodes, like a railroad bridge or the Eiffel tower. The task consists in determining the size (i.e. the cross sectional areas) of the bars that lead to the stiffest truss when submitted to a set of forces, subject to a total weight limit. The problem we want to solve here is a multi-load truss topology design problem, which means we are simultaneously considering a set of k loading scenarios. This problem can be formulated as follows (see [BTN94]):

$$\min \sum_{i=1}^{n} \sigma_i \quad \text{s.t.} \quad \|(q_{i1}, \ldots, q_{ik})\| \le \sigma_i \ \forall 1 \le i \le n \text{ and } B \begin{pmatrix} q_{1j} \\ q_{2j} \\ \vdots \\ q_{nj} \end{pmatrix} = f_j \ \forall 1 \le j \le k \ , \quad \text{(TTD)}$$

where n is the number of bars, k is the number of loading, vector $\sigma \in \mathbb{R}^n$ and matrix $Q \in \mathbb{R}^{n\times k}$ are the design variables, $B \in \mathbb{R}^{m\times n}$ is a matrix describing the physical configuration of the truss and $f_j \in \mathbb{R}^m, 1 \le j \le k$ are vectors of forces describing the loadings scenarios.

It is easily cast as a second-order cone problem in the form (1.3), since the norm constraints can be modelled as $(\sigma_i, q_{i1}, \ldots, q_{ik}) \in \mathbb{L}^k$ for all $1 \le i \le n$. Indeed, we have that the variables $x \in \mathbb{R}^{k(n+1)}$, the objective $c \in \mathbb{R}^{k(n+1)}$ and the equality constraints $Ax = b$ with

$A \in \mathbb{R}^{km \times k(n+1)}$ and $b \in \mathbb{R}^{km}$ are given by

$$x = \begin{pmatrix} \sigma \\ q_{11} \\ q_{21} \\ \vdots \\ q_{nk} \end{pmatrix}, \; c = \begin{pmatrix} 1^{n\times 1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \; A = \begin{pmatrix} 0^{m\times n} & B & & \\ 0^{m\times n} & & B & \\ \vdots & & & \ddots & \\ 0^{m\times n} & & & & B \end{pmatrix} \text{ and } b = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{pmatrix}$$

to give the following second-order cone optimization problem equivalent to (TTD):

$$\min \begin{pmatrix} 1^{n\times 1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}^T \begin{pmatrix} \sigma \\ q_{11} \\ q_{21} \\ \vdots \\ q_{nk} \end{pmatrix} \quad \text{s.t.} \quad \left\{ \begin{array}{l} \begin{pmatrix} 0^{m\times n} & B & & \\ 0^{m\times n} & & B & \\ \vdots & & & \ddots & \\ 0^{m\times n} & & & & B \end{pmatrix} \begin{pmatrix} \sigma \\ q_{11} \\ q_{21} \\ \vdots \\ q_{nk} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{pmatrix} \\ (\sigma_i, q_{i1}, \dots, q_{ik}) \in \mathbb{L}^k \text{ for all } 1 \le i \le n \end{array} \right. . \tag{TP}$$

This allows us to write a dual problem in the form (1.4) in a straightforward manner:

$$\max \sum_{j=1}^{k} f_j^T y_j \quad \text{s.t.} \quad \left\{ \begin{array}{l} \begin{pmatrix} 0^{m\times n} & 0^{m\times n} & \cdots & 0^{m\times n} \\ B^T & & & \\ & B^T & & \\ & & \ddots & \\ & & & B^T \end{pmatrix} \begin{pmatrix} y_\sigma \\ y_1 \\ \vdots \\ y_k \end{pmatrix} + \begin{pmatrix} \sigma^* \\ q^*_{11} \\ q^*_{21} \\ \vdots \\ q^*_{nk} \end{pmatrix} = \begin{pmatrix} 1^{n\times 1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ (\sigma^*_i, q^*_{i1}, \dots, q^*_{ik}) \in \mathbb{L}^k \text{ for all } 1 \le i \le n \end{array} \right. \tag{TD}$$

where $\sigma^* \in \mathbb{R}^n$, $Q^* \in \mathbb{R}^{n\times k}$, $y_\sigma \in \mathbb{R}^m$ and $y_j \in \mathbb{R}^m$ for all $1 \le j \le k$ are the dual variables. Variables σ^* and Q^* can be pivoted out of the formulation using the linear constraints

$$\sigma^* = 1^{n\times 1} \text{ and } q^*_{ij} = -b_i^T y_j \; \forall 1 \le i \le n, \; 1 \le j \le k$$

(where $b_i \in \mathbb{R}^m$ is the i^{th} column of B), which gives then

$$\max \sum_{j=1}^{k} f_j^T y_j \quad \text{s.t.} \quad (1, -b_i^T y_1, \dots, -b_i^T y_k) \in \mathbb{L}^k \text{ for all } 1 \le i \le n$$

and finally

$$\max \sum_{j=1}^{k} f_j^T y_j \quad \text{s.t.} \quad \sum_{j=1}^{k} (b_i^T y_j)^2 \le 1 \text{ for all } 1 \le i \le n \; , \tag{TQC}$$

which is a convex quadratically constrained problem with a linear objective. We can thus solve a truss-topology design problem in at least three different manners: either solving the second-order cone optimization problems (TP) or (TD) or solving the quadratically constrained problem (TQC).

The problems we used for our computational experiments we randomly created using a generator developed by A. Nemirovski. Given three integers p, q and k, it produced the matrix B and vectors f_j corresponding to k loading scenarios for a truss using a 2-dimensional $p \times q$

Table 3: Dimensions of the truss-topology design problems.

Problem description	Formulation	Primal	Dual
2×2 grid with 2 loads	5 cones $\mathbb{L}^2$	15×8	23×15
2×2 grid with 4 loads	5 cones $\mathbb{L}^4$	25×16	41×25
2×2 grid with 8 loads	5 cones $\mathbb{L}^8$	45×32	77×45
2×2 grid with 16 loads	5 cones $\mathbb{L}^{16}$	85×64	149×85
2×2 grid with 32 loads	5 cones $\mathbb{L}^{32}$	165×128	293×165
4×4 grid with 2 loads	114 cones $\mathbb{L}^2$	342×48	390×342
4×4 grid with 4 loads	114 cones $\mathbb{L}^4$	570×96	666×570
4×4 grid with 6 loads	114 cones $\mathbb{L}^6$	798×144	942×798
6×6 grid with 2 loads	615 cones $\mathbb{L}^2$	1845×120	1965×1845
6×6 grid with 4 loads	615 cones $\mathbb{L}^4$	3075×240	3315×3075
8×8 grid with 2 loads	1988 cones $\mathbb{L}^2$	5964×224	6188×5964

nodal grid, with $n \approx \frac{1}{2}p^2q^2$ and $m \approx 2pq$. We tested 11 combinations of parameters p, q and k. The dimensions of the corresponding problems are reported in Table 3 (the last two columns report the number of variables $\times$ the number of constraints). We see that the last problems involve a fairly large number of small second-order cones.

Polyhedral approximations of these problems were computed for three different accuracies, namely $\epsilon = 10^{-2}, 10^{-5}$ and 10^{-8}. The dimensions of the resulting linear optimization problems are reported in Table 4. It is interesting to note that problems with accuracy 10^{-8} are only approximately three times larger than problems with accuracy 10^{-2} and 50% larger than problems with accuracy 10^{-5} (with several dozens of thousands of variables for the largest among them).

Before we present computing times, we have to mention a special feature of this class of problems. Contrary to the general assertion that is stated in Section 2.9, it is possible to give here an estimation of the quality of the optimum objective value of the approximated problem. Indeed, let us call t^* the optimal objective value of problem (TTD) and t^*_ϵ the optimal objective value of the approximated problem with accuracy ϵ. Since our approximation is a relaxation, we obviously have $t^*_\epsilon \leq t^*$, and the optimal solution of the approximated problem $(Q^*_\epsilon, \sigma^*_\epsilon)$ is not necessarily feasible for the original problem. However, Definition 2.1 of a ϵ-approximation of a second-order cone implies in our case that

$$\left\|(q^*_{\epsilon,i1}, \ldots, q^*_{\epsilon,ik})\right\| \leq (1+\epsilon)\sigma^*_{\epsilon,i} \ \forall 1 \leq i \leq n \ ,$$

which means that $(Q^*_\epsilon, (1+\epsilon)\sigma^*_\epsilon)$ is feasible for the original problem, with an objective value equal to $(1+\epsilon)t^*_\epsilon$. Since we must have then $t^* \leq (1+\epsilon)t^*_\epsilon$, we conclude that

$$\frac{t^*}{1+\epsilon} \leq t^*_\epsilon \leq t^* \Leftrightarrow 0 \leq \frac{t^* - t^*_\epsilon}{t^*} \leq \frac{\epsilon}{1+\epsilon} \ ,$$

i.e. we have a bound on the relative accuracy of our approximated optimum objective value.

We generated three random problems for each of the 11 combinations of parameters (p, q, k) presented in Table 3, and report in Table 5 the average computing time in seconds. Each column in this table corresponds to a different way to solve the truss-topology design problem:

Table 4: Dimensions of the approximated problems (primal above, dual below).

$p \times q \times k$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-8}$
$2 \times 2 \times 2$	35×58	60×108	85×158
$2 \times 2 \times 4$	85×146	160×296	235×446
$2 \times 2 \times 8$	200×352	370×692	545×1042
$2 \times 2 \times 16$	420×744	795×1494	1165×2234
$2 \times 2 \times 32$	860×1528	1635×3078	2410×4628
$4 \times 4 \times 2$	798×1188	1368×2328	1938×3468
$4 \times 4 \times 4$	1938×3060	3648×6480	5358×9900
$4 \times 4 \times 6$	3306×5388	6156×11088	9006×16788
$6 \times 6 \times 2$	4305×6270	7380×12420	10455×18570
$6 \times 6 \times 4$	10455×16230	19680×34680	28905×53130
$8 \times 8 \times 2$	13916×20104	23856×39984	33796×59864

$p \times q \times k$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-8}$
$2 \times 2 \times 2$	43×65	68×115	93×165
$2 \times 2 \times 4$	101×155	176×305	251×455
$2 \times 2 \times 8$	232×365	402×705	577×1055
$2 \times 2 \times 16$	484×765	859×1515	1229×2255
$2 \times 2 \times 32$	988×1565	1763×3115	2538×4665
$4 \times 4 \times 2$	846×1482	1416×2622	1986×3762
$4 \times 4 \times 4$	2034×3534	3744×6954	5454×10374
$4 \times 4 \times 6$	3450×6042	6300×11742	9150×17442
$6 \times 6 \times 2$	4425×7995	7500×14145	10575×20295
$6 \times 6 \times 4$	10695×19065	19920×37515	29145×55965
$8 \times 8 \times 2$	14140×25844	24080×45724	34020×65604

Table 5: Computing times to solve truss-topology problems using different approaches.

	QCO	SOCO			10^{-2}		10^{-5}		10^{-8}	
$p \times q \times k$	(D)	(P)	(D)	(D')	(P)	(D)	(P)	(D)	(P)	(D)
$2 \times 2 \times 2$	0.00	0.00	0.00	0.58	0.01	0.00	0.01	0.02	0.04	0.03
$2 \times 2 \times 4$	0.01	0.01	0.00	0.10	0.02	0.02	0.05	0.05	0.11	0.12
$2 \times 2 \times 8$	0.01	0.01	0.01	0.12	0.05	0.05	0.14	0.14	0.33	0.34
$2 \times 2 \times 16$	0.01	0.02	0.01	0.19	0.10	0.11	0.37	0.37	0.85	0.83
$2 \times 2 \times 32$	0.02	0.03	0.03	0.35	0.26	0.27	0.87	0.90	1.96	1.86
$4 \times 4 \times 2$	0.09	0.06	0.29	0.60	0.30	0.21	0.73	0.69	1.54	1.52
$4 \times 4 \times 4$	0.11	0.13	0.74	1.74	1.52	0.92	3.19	2.86	9.30	5.61
$4 \times 4 \times 6$	0.20	0.26	2.22	5.03	3.86	2.09	7.98	5.50	13.50	10.84
$6 \times 6 \times 2$	0.47	0.61	1.96	30.13	2.54	1.88	21.59	5.16	34.95	10.45
$6 \times 6 \times 4$	1.28	1.32	6.48	475.73	19.08	7.82	40.80	23.89	396.04	43.43
$8 \times 8 \times 2$	4.30	2.76	11.81	339.66	12.08	8.89	53.29	24.55	127.68	48.42

a. the first column reports computing times using MOSEK on the quadratically constrained formulation (TQC),

b. the following three columns report computing times using MOSEK on the primal and the dual second-order cone formulations (TP)–(TD) in columns (P) and (D), as well as the results of SeDuMi on the dual formulation in column (D').

c. the last six columns report computing times using the interior-point code in MOSEK to solve the polyhedral approximations of the primal and dual second-order cone problems (TP)–(TD) with three different accuracies.

Our first constatation is that solving the quadratically constrained formulation (TQC) and the primal second-order cone formulation (TP) directly are the two fastest methods (with similar computing times). The quadratically constrained formulation has less variables and constraints, but this advantage seems to be counterbalanced by a more efficient second-order cone solver.

Solving the dual second-order cone formulation (TP) directly is also very fast with a 2×2 nodal grid but noticeably slower on the larger problems (3 to 8 times slower). This is most probably due to the greater dimensions of the problem. The SeDuMi solver is much less efficient to solve these dual problems, and is really slow on the three largest problems.

Let us now look at the approximated problems. First of all, we checked whether the accuracy of the optimum approximated objective was below the theoretical bound $\frac{\epsilon}{1+\epsilon}$, since rounding errors in the computations and handling of irrational coefficients could affect this result. Unsurprisingly, the accuracy was below the theoretical threshold for all experiments.

Computing times are worse than for the direct approaches, even with the low accuracy 10^{-2}. The difference grows up to one or two orders of magnitude for the larger problems. We also observe that despite slightly greater dimensions, solving the approximated dual problem is more efficient than solving the primal problem, especially with the largest problems. The reasons for this behaviour, which is opposite to the situation for direct resolutions, are unclear to us, but could be related to sparsity issues.

Finally, let us mention that we also tried to solve the linear approximations using the simplex algorithm instead of an interior-point method. This lead to surprisingly bad computing times: for example, solving problem $4 \times 4 \times 4$ with accuracy 10^{-2} using the MOSEK[7] simplex code took 21.57 seconds, instead of 1.52 seconds with the interior-point algorithm. We believe this disastrous behaviour of the simplex algorithm is due to the presence of an exponential number of vertices in the approximation, which leads to very slow progress.

3.3 Quadratic optimization

Second-order cone formulations of truss-topology design problems feature a relatively large number of small cones. Since our approximation procedure has not proven to be more efficient than direct methods on these problems, we would like to turn our attention to the opposite configuration, i.e. a small number of large cones. We are going to show that convex quadratic optimization can be formulated such as to meet this requirement.

More specifically, we are going to consider linearly constrained convex quadric optimization problems. Such problems can be formulated as

$$\min \ \frac{1}{2}x^T Q x + c^T x + c_0 \quad \text{s.t.} \quad l_c \le Ax \le u_c \text{ and } l_x \le x \le u_x \ , \tag{QO}$$

where $x \in \mathbb{R}^n$ denotes the vector of design variable. The objective is defined by a matrix $Q \in \mathbb{R}^{n \times n}$, required to be positive semidefinite to ensure convexity of the problem, a vector $c \in \mathbb{R}^n$ and a scalar $c_0 \in \mathbb{R}$. Variables are bounded by two vectors $l_x \in \mathbb{R}^n$ and $u_x \in \mathbb{R}^n$ (note that some components of l_x or u_x can be equal to $-\infty$ or $+\infty$ if a variable has no lower or upper bound). Finally, the linear constraints are described by a matrix $A \in \mathbb{R}^{m \times n}$ and two vectors $l_c \in \mathbb{R}^m$ and $u_c \in \mathbb{R}^m$ (with the same remark holding about possible infinite values for some components of l_c and u_c).

In order to model problem (QO) with a second-order cone formulation, we first write the Cholevsky factorization of matrix Q. Indeed, we have $Q = L^T L$ with $L \in \mathbb{R}^{k \times n}$, where $k \le n$ is the rank of Q. Introducing a vector of auxiliary variables $z \in \mathbb{R}^k$ such that $z = Lx$, we have that $x^T Q x = x^T L^T L x = (Lx)^T Lx = z^T z$, which allows us to write the following problem:

$$\min \frac{r+v}{2} + c^T x + c_0 \quad \text{s.t.} \quad (r, v, z) \in \mathbb{L}^{k+1}, \ r - v = 1, \ l_c \le Ax \le u_c \text{ and } l_x \le x \le u_x \ . \tag{QO'}$$

It is indeed equivalent to (QO), since the conic constraint $(r, v, z) \in \mathbb{L}^{k+1}$ combined with the equality $r - v = 1$ leads to

$$v^2 + \sum_{i=1}^{k} z_i^2 \le r^2 \Leftrightarrow \sum_{i=1}^{k} z_i^2 \le r^2 - v^2 \Leftrightarrow z^T z \le (r-v)(r+v) \Leftrightarrow z^T z \le r + v \ ,$$

which is why the quadratic term $\frac{1}{2}x^T Q x$ in the objective of (QO) could be replaced by $\frac{r+v}{2}$ in (QO').

The problems we tested come from the convex quadratic optimization library QPDATA, collected by Maros and Mészáros [MM99]. As for our tests with truss-topology design problems, we decided to formulate approximations with three different accuracies $10^{-2}, 10^{-5}$ and 10^{-8}. Table 6 lists for each problem its original size (variables $\times$ constraints), the number of

[7] In order to make sure that this behaviour was not caused by a flaw in the MOSEK simplex solver, we performed a similar comparison with the CPLEX solver, which lead to the same conclusion.

nonzero elements in the constraint matrix A, the number of nonzero elements in the upper triangular part of Q and the size (variables $\times$ constraints) of each of the three polyhedral approximations.

Table 7 reports computing times (in seconds) needed to solve these convex quadratic optimization problems in three different ways:

a. the first column reports computing times using MOSEK directly on the original quadratic formulation (QO),

b. the following two columns report computing times needed to solve the second-order cone formulation (QO') of these problems with MOSEK and SeDuMi (in columns labelled (SOCO) and (SOCO') respectively),

c. the last three columns report computing times using MOSEK to solve the polyhedral approximations of the second-order cone problem (QO') with three different accuracies.

Once again, the direct approach, i.e. solving (QO), is the most efficient method. Solving these problems with a second-order cone formulation is slower, especially on larger problems. Using SeDuMi instead of MOSEK degrades further the computing times.

It is also manifest that the linear approximations take more time than the direct approach to provide a solution, even with the lowest accuracy 10^{-2} (however, we note that this low accuracy approximation is faster than the SeDuMi resolution on a few problems).

Although we only tested small-scale and medium-scale problems, it is pretty clear from the trend present for the last problems that large-scale problems would also be most efficiently solved directly as convex quadratic optimization problems.

We pointed out in Section 2.9 that our bound on the accuracy of the polyhedral approximation did not imply anything on the quality of the optimum of the approximated problems. Indeed, Table 8 reports the relative accuracy for a few representative approximated problems. Some problems (`GENHS28`, `DUALC5`) behave very well, with a relative accuracy well below the target accuracy. Other problems (`GOULDQP2`, `MOSARQP1`) have higher relative accuracies, but still decreasing when the target accuracy is decreased. Problem `CVXQP3S` shows a worse behaviour, with virtually no improvement between the second and the third approximation. Finally, `HS21` is a toy problem with the surprising property that its approximation is exact for any accuracy.

We were able to compute these relative accuracies because the true optimal objective values were known by other means. In a real-world situation where such a piece information would not be available, it would be still possible to estimate roughly this accuracy. Indeed, since our approximation is a relaxation, we have that the approximated optimal objective p^*_ϵ is lower than the true optimum p^*. On the other hand, the optimal solution x^*_ϵ of the approximation must be feasible, since it satisfies the linear constraints. Computing the objective function corresponding to this solution, i.e. letting ${p'}^*_\epsilon = \frac{1}{2}x^{*T}_\epsilon Q x^*_\epsilon + c^T x^*_\epsilon + c_0$, we have finally that $p^*_\epsilon \leq p^* \leq {p'}^*_\epsilon$, which allows us to estimate *a posteriori* the true optimum objective value.

However, in the special case where the objective is purely quadratic, i.e. when $c = 0$ and $c_0 = 0$, it is possible to slightly modify the formulation so that we have a bound on the accuracy of the objective[8]. Indeed, letting again $z = Lx$, we add this time the conic constraint

[8] A similar improvement can be made in the case when c belongs to the column space of L^T, the Cholevsky factor of Q. However, we have been unable to generalize this construction to the case where c does not belong to this column space, e.g. for an objective equal to $x_1^2 + x_2$.

Table 6: Statistics for the convex quadratic optimization problems.

Name	Size	ANZ	QNZ	Size 10^{-2}	Size 10^{-5}	Size 10^{-8}
TAME	2×1	2	3	9×13	14×23	19×33
HS21	2×1	2	2	14×22	24×42	34×62
ZECEVIC2	2×2	4	1	9×14	14×24	19×34
HS35	3×1	3	5	20×31	35×61	50×91
HS35MOD	3×1	3	5	20×31	35×61	50×91
HS52	5×3	7	7	29×46	49×86	69×126
HS76	4×3	10	6	28×46	48×86	68×126
HS51	5×3	7	7	29×46	49×86	69×126
HS53	5×3	7	7	29×46	49×86	69×126
S268	5×5	25	15	34×57	59×107	84×157
HS268	5×5	25	15	34×57	59×107	84×157
GENHS28	10×8	24	19	61×100	106×190	151×280
LOTSCHD	12×7	54	6	47×70	76×128	106×188
HS118	15×17	39	15	99×169	174×319	248×467
QPCBLEND	83×74	491	83	545×914	959×1742	1373×2570
CVXQP2S	100×25	74	386	647×1020	1135×1996	1624×2974
CVXQP1S	100×50	148	386	647×1045	1135×2021	1624×2999
CVXQP3S	100×75	222	386	647×1070	1135×2046	1624×3024
QPCBOEI2	143×166	1196	143	939×1614	1652×3040	2366×4468
DUALC5	8×278	2224	36	54×61	94×441	134×521
PRIMALC1	230×9	2070	229	1505×2329	2646×4611	3789×6897
PRIMALC5	287×8	2296	286	1878×2903	3304×5755	4732×8611
DUAL4	75×1	75	2799	493×761	867×1509	1241×2257
GOULDQP2	699×349	1047	697	2635×3872	4370×7342	6107×10816
DUAL1	85×1	85	3558	558×861	982×1709	1406×2557
PRIMALC8	520×8	4160	519	3410×5268	5996×10440	8586×15620
GOULDQP3	699×349	1047	1395	4584×7420	8062×14376	11546×21344
DUAL2	96×1	96	4508	632×976	1110×1932	1589×2890
MOSARQP2	900×600	2390	945	5911×9721	10395×18689	14887×27673

Table 7: Computing times to solve convex quadratic optimization problems

Name	QO	SOCO	SOCO'	10^{-2}	10^{-5}	10^{-8}
TAME	0.06	0.11	0.46	0.01	0.01	0.02
HS21	0.00	0.01	0.17	0.01	0.01	0.02
ZECEVIC2	0.01	0.00	0.101	0.00	0.00	0.03
HS35	0.00	0.00	0.1	0.01	0.01	0.02
HS35MOD	0.00	0.03	0.361	0.01	0.01	0.02
HS52	0.00	0.01	0.09	0.01	0.02	0.03
HS76	0.00	0.00	0.11	0.00	0.02	0.04
HS51	0.00	0.01	0.08	0.00	0.02	0.04
HS53	0.01	0.00	0.09	0.00	0.02	0.03
S268	0.02	0.01	0.20	0.01	0.04	0.07
HS268	0.01	0.00	0.20	0.01	0.04	0.07
GENHS28	0.00	0.01	0.08	0.01	0.03	0.07
LOTSCHD	0.02	0.01	0.16	0.02	0.03	0.08
HS118	0.00	0.01	0.25	0.05	0.11	0.17
QPCBLEND	0.03	0.12	0.83	0.41	1.43	2.75
CVXQP2S	0.02	0.23	1.34	0.58	1.91	3.58
CVXQP1S	0.03	0.14	1.55	0.57	1.88	4.47
CVXQP3S	0.05	0.21	1.86	0.60	2.32	3.93
QPCBOEI2	0.08	0.59	4.01	1.26	5.53	11.55
DUALC5	0.01	0.01	1.56	0.04	0.04	0.09
PRIMALC1	0.04	0.91	2.17	1.27	7.93	17.85
PRIMALC5	0.03	0.83	0.69	3.15	9.49	20.83
DUAL4	0.04	0.13	0.57	0.39	0.83	1.42
GOULDQP2	1.21	0.41	1.87	3.54	10.14	20.65
DUAL1	0.06	0.18	0.72	0.72	1.66	2.53
PRIMALC8	0.08	4.72	4.32	3.97	30.32	50.99
GOULDQP3	0.98	12.16	6.32	9.16	38.44	74.15
DUAL2	0.07	0.16	0.93	0.88	2.03	3.03
MOSARQP2	0.38	1.64	8.512	7.69	39.03	76.59

Table 8: Relative accuracy of the optimum of some approximated problems.

Name	10^{-2}	10^{-5}	10^{-8}
GENHS28	5.2e-4	8.8e-7	7.6e-10
HS21	3.3e-10	3.3e-10	3.3e-10
CVXQP3S	6.7e-3	7.2e-4	4.7e-4
DUALC5	3.9e-3	7.2e-7	3e-10
GOULDQP2	5.2e-1	5.6e-3	2.1e-5
MOSARQP2	8.6e-1	2.5e-4	8.5e-7

$(r, z) \in \mathbb{L}^k$, which implies to $z^T z \leq r^2 \Leftrightarrow x^T Q x \leq r^2$. We can now choose r as our objective, which is equivalent to minimizing $\sqrt{x^T Q x}$ and is obviously the same thing as minimizing the true quadratic objective $\frac{1}{2} x^T Q x$. This leads to a situation that is very similar to the case of truss-topology design problems, and one can show without difficulties that this approximated problem provides an estimation of the true optimum with a relative accuracy equal to $(\frac{\epsilon}{1+\epsilon})^2$.

4 Concluding remarks

In this paper, we presented a polyhedral approximation of the second-order cone originally developed by Ben-Tal and Nemirovski [BTN98]. Our presentation features several improvements, including smaller dimensions for the approximation, a more transparent proof, complete developments valid for any size of the second-order cone (i.e. not limited to powers of two) and explicit constants in the derivation of a theoretical bound on the size of the approximation (Theorem 2.4).

This scheme was implemented in MATLAB and optimized as much as possible. Indeed, we developed several approaches to reduce the size of the resulting linear problems (including pivoting out some variables and using dynamic programming to choose the best accuracies for each stage of the decomposition). Our experiments mainly showed that solving the original second-order cone problems or alternative equivalent formulations is more efficient than solving the linear approximations, even at low accuracies. On a side note, we noticed that these approximate problems are particularly difficult to solve with the simplex algorithm.

However, we would like to point out this approximating scheme can still prove very useful in certain well-defined circumstances, such as a situation where a user is equipped with a solver that is only able to solve linear optimization problems. In this case, this procedure provides him with an inexpensive and relatively straightforward way to test improved versions of his linear models that make use of second-order cones.

Moreover, we have to admit that we tested two very specific classes of second-order cone optimization problems for which either a simplified formulation or a well-understood dedicated algorithm was available. It might well be possible that this linearizing scheme becomes competitive for other types of difficult (i.e. that cannot be simplified and for which no dedicated solver is available) second-order cones optimization problems.

We would also like to insist on the fact that it is not possible to guarantee *a priori* the accuracy of a linear approximation of a general second-order optimization problem (see the example in Section 2.9). It is nevertheless possible to provide such a bound in some special cases (e.g. truss-topology design problems or convex quadratic optimization problems with a pure quadratic objective).

It is worth to point out that a straightforward modification of our polyhedral approximation of $\mathbb{L}^2$ can lead to a restriction instead of a relaxation of second-order cone optimization problems. This would then provide an upper bound instead of a lower bound on the true optimum objective value, and optimal solutions of the approximate problems would always be feasible for the original problem. However, this approach can be problematic in some cases since it might happen that the approximated problem is unfeasible, even if the original problem admits some feasible solutions.

An interesting topic for further research is the generalization of the polyhedral approximation of $\mathbb{L}^2$ or, more precisely, of the unit ball $\mathcal{B}_2(1)$, to other convex sets. Indeed, finding a similar polyhedral approximation for a set like $\{(x_1, x_2) \in \mathbb{R}^2 \mid |x_1|^p + |x_2|^p \leq 1\}$ with $p > 1$, i.e. the unit ball for the p-norm, would lead to linearizing scheme for other classes of convex problems, such as l_p-norm optimization. However, it is unclear to us at this stage whether this goal is achievable or not, since the symmetry of the standard unit ball, which is not present for other norms, seems to play a great role in the construction of the approximation.

Acknowledgements. The author wishes to thank Professor Kees Roos and the SSOR department at the Delft University of Technology for their kind hospitality, as well as Erling Andersen for making the MOSEK solver available to him.

References

[AA99] E. D. Andersen and K. D. Andersen, *The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm*, High Performance Optimization (H. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds.), Applied optimization, vol. 33, Kluwer Academic Publishers, 1999.

[ART00] E. D. Andersen, C. Roos, and T. Terlaky, *On implementing a primal-dual interior-point method for conic quadratic optimization*, in preparation, 2000.

[BTN94] A. Ben-Tal and A. Nemirovski, *Potential reduction polynomial-time method for truss topology design*, SIAM Journal of Optimization **4** (1994), 596–612.

[BTN98] A. Ben-Tal and A. Nemirovski, *On polyhedral approximations of the second-order cone*, Tech. report, Minerva Optimization Center, Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa, Israel, 1998, to appear in Mathematics of Operations Research.

[LVBL98] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, *Applications of second-order cone programming*, Linear Algebra and its Applications **284** (1998), 193–228.

[MM99] I. Maros and Cs. Mészáros, *A repository of convex quadratic programming problems*, Optimization Methods and Software **11-12** (1999), 671–681, special issue on interior-point methods (CD supplement with software), guest editors: Florian Potra, Cornelis Roos and Tamás Terlaky.

[NN94] Y. E. Nesterov and A. S. Nemirovski, *Interior-point polynomial methods in convex programming*, SIAM Studies in Applied Mathematics, SIAM Publications, Philadelphia, 1994.

[Stu99a] J. F. Sturm, *Duality results*, High Performance Optimization (H. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds.), Applied optimization, vol. 33, Kluwer Academic Publishers, 1999, pp. 21–60.

[Stu99b] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software **11-12** (1999), 625–653, special issue on interior-point methods (CD supplement with software), guest editors: Florian Potra, Cornelis Roos and Tamás Terlaky.

[SW70] J. Stoer and Ch. Witzgall, *Convexity and optimization in finite dimensions I*, Springer Verlag, Berlin, 1970.