

Conic Optimization

Mikkel Paltorp (mpasc@dtu.dk), 4 January 2026

Department of Applied Mathematics and Computer Science, Technical University of Denmark

1. Introduction

This note introduces some central ideas behind the implementation of efficient interior-point methods for solving conic programs. A large portion of the theory section is based on the papers for the conic solvers Clarabel¹ [1] and CVXOPT [2]. What this note adds is some essential implementation details that were left out of the original two papers (e.g. iterative refinement, matrix equilibration, and handling of the structure of the scaling matrices).

2. Primal and dual problems

We will in this text consider optimization problems where the objective function is at most quadratic and the constraints are conic (including the **zero cone** in order to model equality constraints). The general form of such (primal) problems is the following

$$\begin{aligned} & \text{minimize} && \frac{1}{2}x^\top Px + q^\top x \\ & \text{subject to} && Ax + s = b \\ & && s \in \mathcal{K}, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the decision variable, $s \in \mathbb{R}^m$ is the slack variable, and $P = P^\top \succeq 0 \in \mathbb{R}^{n \times n}$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Furthermore, $\mathcal{K}$ is a composite i.e. the Cartesian product of K cones ($\mathcal{K} = \mathcal{K}_1 \times \dots \times \mathcal{K}_K$) with each of the cones $\mathcal{K}_k$ of dimension l_k being of the following types

- The **zero cone**: $\mathcal{Z}_{l_k} := \{0\}^{l_k}$ (used to model equality constraints).
- The **nonnegative cone**: $\mathbb{R}_+^{l_k} := \{s \in \mathbb{R}^{l_k} \mid s \geq 0\}$.
- The **second-order cone**: $\mathcal{Q}_{l_k} := \{(s_1, s_2) \in \mathbb{R} \times \mathbb{R}^{l_k-1} \mid \|s_2\|_2 \leq s_1\}$.
- The **exponential cone**: $\mathcal{K}_{\text{exp}} := \{(s_1, s_2, s_3) \in \mathbb{R}^3 \mid s_2 \exp\left(\frac{s_1}{s_2}\right) \leq s_3, s_2 > 0\} \cup \{(x, 0, z) \mid x \leq 0, z \geq 0\}$.
- The **power cone**: $\mathcal{K}_{\text{pow}} := \{(s_1, s_2, s_3) \in \mathbb{R}^3 \mid s_1^\alpha s_2^{1-\alpha} \geq |s_3|, s_1, s_2 \geq 0\}$ with $\alpha \in (0, 1)$.
- The **positive semidefinite cone**: $\mathcal{S}_{l_k} := \{s \in \mathbb{R}^{\frac{l_k(l_k+1)}{2}} \mid \text{smat}(s) \succeq 0\}$.

where $\text{smat}(x)$ creates a symmetric matrix from the vector x . This can be done in a plethora of ways, but we will use the definition from Clarabel, which fills the lower triangle row by row

$$\text{smat}(x) = \begin{bmatrix} x_1 & x_2/\sqrt{2} & \dots & x_{n(n-1)/2+1}/\sqrt{2} \\ x_2/\sqrt{2} & x_3 & \dots & x_{n(n-1)/2+2}/\sqrt{2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n(n-1)/2+1}/\sqrt{2} & x_{n(n-1)/2+2}/\sqrt{2} & \dots & x_{n(n+1)/2} \end{bmatrix}. \tag{2}$$

where the scaling of $\sqrt{2}$ have been introduced on the off-diagonal elements in order to preserve the inner product, i.e. $x^\top y = \text{tr}(\text{smat}(x) \text{smat}(y))$. Furthermore, we

Contents

1. Introduction
 2. Primal and dual problems
 - 2.1. The dual problem
 - 2.2. Homogeneous embedding
 3. Barrier functions
 - 3.1. The logarithmic barrier function for the three symmetric cones
 - 3.2. Gradients of the barrier function for the three symmetric cones
 - 3.3. Hessians of the barrier function for the three symmetric cones
 - 3.4. The Jordan product for the three symmetric cones
 - 3.5. Logarithmically homogeneous self-concordant barrier
 4. The central path
 - 4.1. Scaling matrices
 5. Computing step directions
 - 5.1. The affine and centering directions
 - 5.2. Solver initialization
 - 5.3. Termination criteria
 6. Implementation details
 - 6.1. Solving the KKT system
 - 6.2. Regularized iterative refinement
 - 6.3. Matrix equilibration
 7. Sparse solvers
 - 7.1. Sparsity exploitation
 - 7.2. Second-order cones
 - 7.3. Rotated second-order cones
 - 7.4. Power cones
 - 7.5. Generalized power cones
 - 7.6. Power mean cone
 - 7.7. Relative entropy cone
 - 7.8. Semidefinite cones
- Bibliography

¹ You can play around with a few examples using a Wasm-compiled Clarabel here.

denote the inverse operation by $\text{svec}(X)$, i.e. an operation that creates a vector from a symmetric matrix X . The nonnegative, second-order, and positive semidefinite cones are examples of **symmetric cones** which are a special class of cones that have a lot of nice properties. The exponential and power cones are examples of **nonsymmetric cones** which do not have the same nice properties as symmetric cones, but are still very useful in practice. The zero cone is a special case that is used to model equality constraints.

2.1. The dual problem

The primal optimization problem of interest can equivalently be written as

$$\text{Primal Problem : } \begin{cases} \text{minimize} & \frac{1}{2}x^\top Px + q^\top x, \\ \text{subject to} & s = b - Ax \in \mathcal{K}. \end{cases} \quad (3)$$

where the conic constraint $b - Ax \in \mathcal{K}$ can be understood as a **generalized inequality**, i.e.

$$0 \preceq_{\mathcal{K}} s = b - Ax. \quad (4)$$

However, the standard form (similar to LPs) requires that the inequality is turned the other way. As such we negate the expression, resulting in

$$Ax - b \preceq_{\mathcal{K}} 0. \quad (5)$$

We now write up the Lagrangian as

$$L(x, z) = \frac{1}{2}x^\top Px + q^\top x + z^\top (Ax - b) \quad (6)$$

Since $-(Ax - b) \in \mathcal{K}$ we must have that z is in the dual cone $\mathcal{K}^*$ in order for the Lagrangian to be a lower bound for any feasible x . This comes directly from the definition of the dual cone, i.e.

$$\mathcal{K}^* = \{z \in \mathbb{R}^m \mid z^\top s \geq 0, \quad \forall s \in \mathcal{K}\}, \quad (7)$$

so that, in particular, $z^\top (Ax - b) \leq 0$ for every x with $b - Ax \in \mathcal{K}$.

Now since we want to find a stationary point we compute the gradient of the Lagrangian and set it to zero

$$\nabla_x L(x, z) = Px + q + A^\top z = 0. \quad (8)$$

Now notice that the Lagrangian can be written as

$$x^\top \left(\frac{1}{2}Px + q + A^\top z \right) - b^\top z \quad (9)$$

The thing inside the parentheses, is **almost** equal to the zero constraint on the gradient. By performing the mathematician's favorite magic trick of adding zero followed by rearranging the expressions we end up with

$$\begin{aligned} & x^\top \left(\frac{1}{2}Px + q + A^\top z \right) + \underbrace{\left(\frac{1}{2}x^\top Px - \frac{1}{2}x^\top Px \right)}_{=0} - b^\top z \\ = & \underbrace{x^\top (Px + q + A^\top z)}_{\nabla_x L(x, z) = 0} - b^\top z - \frac{1}{2}x^\top Px. \end{aligned} \quad (10)$$

As such the dual formulation of the problem becomes

$$\text{Dual Problem: } \begin{cases} \text{maximize} & -\frac{1}{2}x^\top Px - b^\top z, \\ \text{subject to} & Px + q + A^\top z = 0, \\ & z \in \mathcal{K}^* \end{cases} \quad (11)$$

2.2. Homogeneous embedding

The standard KKT conditions for the primal problem are

$$\begin{aligned} Ax + s &= b \\ Px + A^\top z &= -q \\ s^\top z &= 0 \\ (s, z) &\in \mathcal{K} \times \mathcal{K}^*. \end{aligned} \quad (12)$$

However, we can take a different path in order to solve the same problem. First we define the duality gap γ as the difference between the primal and dual objectives

$$\begin{aligned} \gamma &:= \left(\frac{1}{2}x^\top Px + q^\top x \right) - \left(-\frac{1}{2}x^\top Px - b^\top z \right) \\ &= x^\top Px + q^\top x + b^\top z = s^\top z \end{aligned} \quad (13)$$

where we used that $q = -Px - A^\top z$ (from the gradient constraint) and $s = b - Ax$ (from the definition of the equality constraints). Because $(s, z) \in (\mathcal{K}, \mathcal{K}^*)$ the duality gap is always nonnegative for any feasible point, and is zero at the optimal point (x^*, s^*, z^*) when strong duality holds. We can therefore come up with a different optimization problem by combining the constraints of the primal and dual problem and with an objective to minimize the duality gap

$$\begin{aligned} &\underset{x, s, z}{\text{minimize}} && s^\top z, \\ &\text{subject to} && Px + A^\top z + q = 0, \\ & && Ax + s - b = 0, \\ & && x^\top Px + q^\top x + b^\top z = 0, \\ & && (s, z) \in \mathcal{K} \times \mathcal{K}^*. \end{aligned} \quad (14)$$

An issue with the above is that the duality gap is explicitly set to zero in the third equality constraint. This condition can be relaxed after a change of variables $x \rightarrow \frac{x}{\tau}$, $z \rightarrow \frac{z}{\tau}$, and $s \rightarrow \frac{s}{\tau}$ and instead solve

$$\text{Homogeneous embedding: } \begin{cases} \underset{x, s, z, \tau, \kappa}{\text{minimize}} & s^\top z + \tau \kappa \\ \text{subject to} & Px + A^\top z + q\tau = 0, \\ & Ax + s - b\tau = 0, \\ & \tau^{-1}x^\top Px + q^\top x + b^\top z + \kappa = 0, \\ & (s, z, \tau, \kappa) \in \mathcal{K} \times \mathcal{K}^* \times \mathbb{R}_+ \times \mathbb{R}_+. \end{cases} \quad (15)$$

In [1] they go on to show the existence of solutions and infeasibility detection. This will be skipped here.

3. Barrier functions

Barrier functions are functions that are defined only inside of the feasible region (i.e. when $s = b - Ax \in \mathcal{K}$) and tend towards infinity when we move towards the boundary of the feasible set. They are useful as they can be used to approximate the constrained problem by an unconstrained problem

$$\begin{cases} \text{minimize} & \frac{1}{2}x^\top Px + q^\top x \\ \text{subject to} & Ax + s = b \\ & s \in \mathcal{K}, \end{cases} \approx \text{minimize} \quad \frac{1}{2}x^\top Px + q^\top x + \mu f(b - Ax). \quad (16)$$

As $\mu \rightarrow 0$ the solution to the unconstrained problem tends towards the solution of the constrained problem. A naive approach would simply be to pick μ to be a small value in order to get a good approximation of the original problem. However, this is not a good idea as the resulting optimization problem is very ill-conditioned and hard to solve. Instead, interior-point methods start with a large value of μ and then track the solution of the unconstrained problem as μ decreases until we are reasonably close to the optimal solution of the original problem. Methods that rely on this strategy of solving a series of simpler problems are known as **homotopy methods**. As an example we will now describe the properties of the logarithmic barrier functions for the three symmetric cones², but later we will also discuss the barrier functions for the exponential and power cones.

² That is the nonnegative, second-order, and positive semidefinite cone.

3.1. The logarithmic barrier function for the three symmetric cones

For a composite cone $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \dots \times \mathcal{K}_K$ where each component is one of the three symmetric cones we have that the logarithmic barrier function is given by

$$f(s) = \sum_{k=1}^K f_k(s_k), \quad f_k(s) = \begin{cases} -\sum_{j=1}^n \log(s_j) & \mathcal{K}_k = \mathbb{R}_+^{l_k} \\ -\frac{1}{2} \log(s^\top J s) & \mathcal{K}_k = \mathcal{Q}_{l_k} \\ -\log \det \text{smat}(s) & \mathcal{K}_k = \mathcal{S}_{l_k} \end{cases} \quad (17)$$

where $s = [s_1 \ s_2 \ \dots \ s_K]^\top$ with $s_k \in \mathcal{K}_k$ and

$$J = \begin{bmatrix} 1 & 0 \\ 0 & -I_{l_k-1} \end{bmatrix}, \quad (18)$$

meaning that $s_k^\top J s_k = s_1^2 - \|s_2\|^2 \geq 0$ ³ for all $s_k \in \mathcal{Q}_{l_k}$. Furthermore, the logarithmic barrier function satisfies the relation $f(ts) = f(s) - m \log(t)$ for $t > 0$ where

³ The notation here is a little sloppy as we use that $s_k = (s_1, s_2) \in \mathbb{R} \times \mathbb{R}^{l_k-1}$.

$$m = m_1 + \dots + m_K, \quad m_k = \begin{cases} l_k & \mathcal{K}_k = \mathbb{R}_+^{l_k} \\ 1 & \mathcal{K}_k = \mathcal{Q}_{l_k} \\ l_k & \mathcal{K}_k = \mathcal{S}_{l_k} \end{cases}. \quad (19)$$

We refer to m_k as the degree of the k th cone, so that m is the degree of the composite cone $\mathcal{K}$.

3.2. Gradients of the barrier function for the three symmetric cones

By linearity the gradient of the logarithmic barrier function is given by the sum of gradients of the individual barrier functions for each symmetric cone given by

$$\nabla f_k(s_k) = \begin{cases} -\text{diag}(s_k)^{-1} 1_{l_k} & \mathcal{K}_k = \mathbb{R}_+^{l_k} \\ -(s_k^\top J s_k)^{-1} J s_k & \mathcal{K}_k = \mathcal{Q}_{l_k} \\ -\text{svec}(\text{smat}(s_k)^{-1}) & \mathcal{K}_k = \mathcal{S}_{l_k} \end{cases} \quad (20)$$

where 1_l is a l -vector of ones. It can be shown that $-\nabla f(s) \in \mathcal{K}$ and that $s^\top \nabla f(s) = -m$ for $s \in \mathcal{K}$.

3.3. Hessians of the barrier function for the three symmetric cones

We denote the Hessian of f and f_k as $H(s) = \nabla^2 f(s)$ and $H_{k(s_k)} = \nabla^2 f_{k(s_k)}$:

$$H_k(s_k) = \begin{cases} \text{diag}(s_k)^{-2} & \mathcal{K}_k = \mathbb{R}_+^l \\ (s_k^\top J s_k)^{-2} (2J s_k s_k^\top J - (s_k^\top J s_k)J) & \mathcal{K}_k = \mathcal{Q}_l \\ \text{smat}(s_k)^{-1} \otimes_s \text{smat}(s_k)^{-1} & \mathcal{K}_k = \mathcal{S}_l \end{cases} \quad (21)$$

where $\otimes_s$ is the symmetric Kronecker product, defined in the section on semidefinite cones at the end of the post. For the scaling matrices below, we also need $R_k = L_1 V \Lambda_k^{-1/2} = L_2^\top U \Lambda_k^{1/2}$, which depends on both s_k and z_k , with

$$\text{smat}(s_k) = L_1 L_1^\top, \quad \text{smat}(z_k) = L_2 L_2^\top, \quad L_2^\top L_1 = U \Lambda_k V^\top. \quad (22)$$

In addition, we have that

$$H_k(s_k)^{-1} = \begin{cases} \text{diag}(s_k)^2 & \mathcal{K}_k = \mathbb{R}_+^l \\ 2s_k s_k^\top - (s_k^\top J s_k)J & \mathcal{K}_k = \mathcal{Q}_l \\ \text{smat}(s_k) \otimes_s \text{smat}(s_k) & \mathcal{K}_k = \mathcal{S}_l \end{cases} \quad (23)$$

Further properties of the Hessian of the logarithmic barrier function are

$$\begin{aligned} H(s)^{-1} &= H(s^{-1}), \quad H(s)^{1/2} = H(s^{1/2}) \\ H(s)s &= s^{-1}, \quad H(s^{1/2})s = e, \quad (H(s)v)^{-1} = H(s)^{-1}v^{-1}. \end{aligned} \quad (24)$$

3.4. The Jordan product for the three symmetric cones

The Jordan products for the three symmetric cones are given by

$$s \circ z = \begin{cases} [s_1 z_1 \ s_2 z_2 \ \dots \ s_l z_l]^\top & \mathcal{K}_k = \mathbb{R}_+^{l_k} \\ [s^\top z \ s_1 z_2^\top + s_2^\top z_1]^\top & \mathcal{K}_k = \mathcal{Q}_{l_k} \\ \frac{1}{2} \text{svec}(\text{smat}(s) \text{smat}(z) + \text{smat}(z) \text{smat}(s)) & \mathcal{K}_k = \mathcal{S}_{l_k} \end{cases} \quad (25)$$

where e is the idempotent for the cones given by

$$e_k = \begin{cases} [1 \ 1 \ \dots \ 1]^\top & \mathcal{K}_k = \mathbb{R}_+^{l_k} \\ [1 \ 0 \ \dots \ 0]^\top & \mathcal{K}_k = \mathcal{Q}_{l_k} \\ \text{svec}(I_{l_k}) & \mathcal{K}_k = \mathcal{S}_{l_k} \end{cases} \quad (26)$$

Note that $e^\top(z \circ s) = z^\top s$ and $e^\top e = m$. In addition, the inverse, square, and square root are defined by the relations $s^{-1} \circ s = e$, $s^2 = s \circ s$, and $s^{\frac{1}{2}} \circ s^{\frac{1}{2}} = s$.

3.5. Logarithmically homogeneous self-concordant barrier

The ideas of the logarithmic barrier functions can be extended to the nonsymmetric cones as well. We often refer to the general class of **logarithmically homogeneous self-concordant barrier** functions, which are a special class of barrier functions that have nice properties that can be exploited in the design of interior-point methods as follows: A function $f : \mathcal{K} \rightarrow \mathbb{R}$ is called a logarithmically homogeneous self-concordant barrier with degree $m \geq 1$ (m -LHSCB)⁴ for the convex cone $\mathcal{K}$ if it satisfies the following properties

$$\begin{aligned} |\nabla^3 f(x)[r, r, r]| &\leq 2(\nabla^2 f(x)[r, r])^{3/2} \quad \forall x \in \text{int}(\mathcal{K}), r \in \mathbb{R}^d, \\ f(tx) &= f(x) - m \log(t), \quad \forall x \in \mathcal{K}, t > 0. \end{aligned} \quad (27)$$

The **conjugate barrier function** $f_* : \mathcal{K}^* \rightarrow \mathbb{R}$ is defined as⁵

⁴ The logarithmic barrier functions from the previous section are all m -LHSCB functions.

⁵ This is not to be confused with the **convex conjugate** of a function, which is denoted by $f^*(y) = \sup \{y^\top x - f(x), x \in \text{int}(\mathcal{K})\}$. The relation between the two conjugates are $f_*(y) = f^*(-y)$ and $\nabla f_*(y) = -\nabla f^*(-y)$.

$$f_*(y) := \sup_{x \in \text{int}(\mathcal{K})} \{-y^\top x - f(x)\}. \quad (28)$$

The function f_* is m -LHSCB for $\mathcal{K}^*$ if the associated f is. The primal gradient ∇f satisfies

$$x^\top \nabla f(x) = -m, \quad \forall x \in \text{int}(\mathcal{K}). \quad (29)$$

In cases where the conjugate barrier function is known only through the definition (i.e. rather than a closed-form representation) we can compute its gradient as the solution to

$$\nabla f_*(y) := - \arg \sup_{x \in \text{int}(\mathcal{K})} \{-y^\top x - f(x)\}. \quad (30)$$

4. The central path

As described previously, the idea of interior-point methods is to track the solution of an approximate unconstrained problem as μ decreases until we are reasonably close to the optimal solution of the original constrained problem. The central path denotes this path. In order to describe the central path mathematically we define the function G representing the equality constraints of the homogeneous embedding

$$G(x, z, s, \tau, \kappa) := \begin{bmatrix} 0 \\ s \\ \kappa \end{bmatrix} - \begin{bmatrix} P & A^\top & q \\ -A & 0 & b \\ -q^\top & -b^\top & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ \tau \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tau^{-1} x^\top P x \end{bmatrix}. \quad (31)$$

Now assume that $f : \mathcal{K} \rightarrow \mathbb{R}$ is an m -LHSCB function on $\mathcal{K}$ with conjugate barrier f_* for some $m \geq 1$. Given any initial v^0 , we define the **central path** of $v^*(\mu)$ as the unique solution to

$$\begin{aligned} G(v) &= \mu G(v^0) \\ s &= -\mu \nabla f_*(z), \quad z = -\mu \nabla f(s), \quad \tau \kappa = \mu \end{aligned} \quad (32)$$

Now inserting the definition of z while using that $s^\top \nabla f(s) = -m$ it follows that

$$s^\top \frac{z}{m} = s^\top \frac{-\mu \nabla f(s)}{m} = \mu \frac{m}{m} = \mu. \quad (33)$$

If the cone $\mathcal{K}$ is symmetric, then we can replace the condition with a simpler condition defined in terms of the Jordan product on $\mathcal{K}$, i.e.

$$s \circ z = \mu e, \quad (34)$$

where e is the idempotent for each of the symmetric cones.

Interactive figure 1: A central path in two dimensions

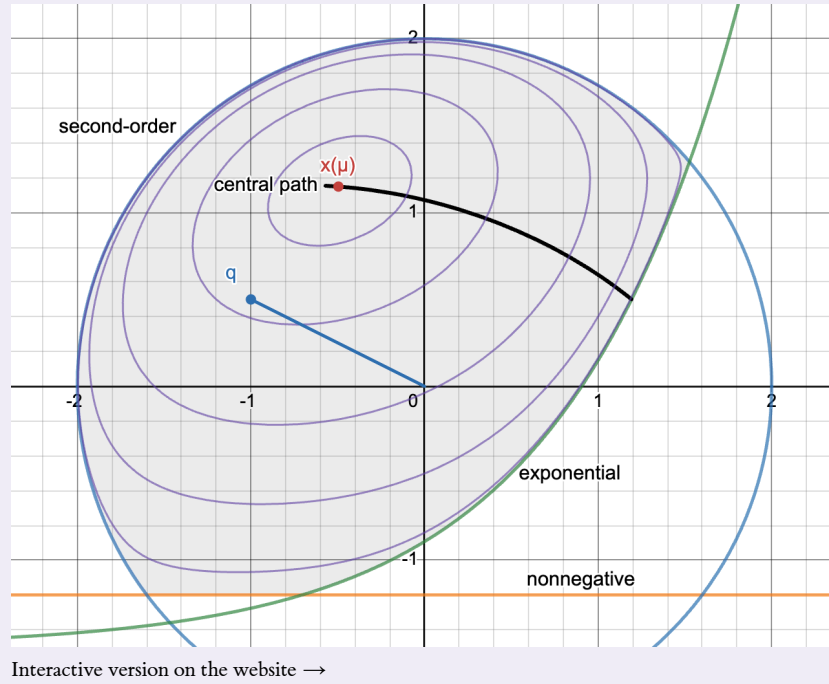
Interactive figure 1 (A central path in two dimensions) Consider $x \in \mathbb{R}^2$, the objective $q^\top x$, and the composite cone $\mathcal{K} = \mathbb{R}_+ \times \mathcal{Q}_3 \times \mathcal{K}_{\text{exp}}$ with

$$s = b - Ax = \begin{bmatrix} 1.2 \\ 2 \\ 0 \\ 0 \\ -0.5 \\ 1 \\ 1.5 \end{bmatrix} - \begin{bmatrix} 0 & -1 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{x_2 + 1.2}{2} \\ x_1 \\ x_2 \\ x_1 - 0.5 \\ 1 \\ x_2 + 1.5 \end{bmatrix} \in \mathcal{K}. \quad (35)$$

The three cones give, respectively, the half-plane $x_2 \geq -1.2$, the disk $\|x\|_2 \leq 2$, and the region $x_1 \leq 0.5 + \log(x_2 + 1.5)$, and the barrier is

$$f(s) = -\log(s_1) - \log(s_2^2 - s_3^2 - s_4^2) - \log(s_6 \log(s_7/s_6) - s_5) - \log(s_6) - \log(s_7), \quad (36)$$

of degree $m = 1 + 2 + 3 = 6$ (here the second-order cone uses $-\log(s^\top J s)$, twice the barrier of the previous section, so it contributes degree 2). The central path is the curve of minimizers $x(\mu)$ of $q^\top x + \mu f(b - Ax)$ and along it $z = -\mu \nabla f(s)$. For large μ the barrier dominates, and $x(\mu)$ lies close to the analytic center, well inside the feasible region. As μ decreases the linear term takes over, and the level sets are pushed against the boundary, where they follow its shape.



4.1. Scaling matrices

We now introduce the idea of a scaling matrix H . The choice of scaling matrix H depends on which way we linearize the central path. For symmetric cones the most common choice is the Nesterov-Todd scaling. For nonsymmetric cones the central path is defined by the set of points satisfying Equation 32. For the zero cone we set $H = 0$.

4.1.1. Symmetric cones

For symmetric cones the central path described with Equation 34 is linearized. The Nesterov–Todd scaling exploits the self-scaled property of symmetric cones to compute a unique scaling point $w \in \mathcal{K}$ satisfying

$$H(w)s = z. \quad (37)$$

We can factorize $H(w)$ as $H^{-1}(w) = W^\top W$.

4.1.2. Nonsymmetric cones

As nonsymmetric cones are not self-scaled, we cannot use Equation 34 to describe the central path and must instead linearize Equation 32. A general primal-dual scaling strategy suitable for nonsymmetric cones was proposed in [3], and used later in [4], which relies on the satisfaction of the two secant equations

$$Hz = s, \quad H\nabla f(s) = \nabla f_*(z). \quad (38)$$

We now define **shadow iterates** as

$$\tilde{z} := -\nabla f(s), \quad \tilde{s} := -\nabla f_*(z), \quad (39)$$

with

$$\tilde{\mu} = \tilde{s}^\top \tilde{z} / m \quad (40)$$

A scaling matrix H can be obtained from the rank-4 Broyden–Fletcher–Goldfarb–Shanno (BFGS) update, which is commonly used in quasi-Newton methods,

$$H := H_{\text{BFGS}} := S(S^\top Z)^{-1}S^\top + H_a - H_a Z(Z^\top H_a Z)^{-1}Z^\top H_a, \quad (41)$$

where $Z := [z, \tilde{z}]$, $S := [s, \tilde{s}]$, and $H_a \succ 0$ is an approximation of the Hessian. In our implementation we choose $H_a = \mu \nabla^2 f_*(z)$ following [4]. Putting things together means that the computation of H_{BFGS} reduces to a rank-3-update as

$$\begin{aligned} H_{\text{BFGS}} = & \mu \nabla^2 f_*(z) + (2\mu m)^{-1} \delta_s (s + \mu \tilde{s} + (\mu \tilde{\mu} - 1)^{-1} \delta_s)^\top \\ & + (2\mu m)^{-1} (s + \mu \tilde{s} + (\mu \tilde{\mu} - 1)^{-1} \delta_s) \delta_s^\top \\ & - \mu \frac{(\nabla^2 f_*(z) \tilde{z} - \tilde{\mu} \tilde{s})(\nabla^2 f_*(z) \tilde{z} - \tilde{\mu} \tilde{s})^\top}{\langle \tilde{z}, \nabla^2 f_*(z) \tilde{z} \rangle - m \tilde{\mu}^2}, \end{aligned} \quad (42)$$

where $\mu = s^\top z / m$ and $\delta_s := s - \mu \tilde{s}$. A more detailed description can be found in [4].

The above is defined in terms of the conjugate barrier function $f_{*(z)}$. This is in contrast to [4], which defines it in terms of the standard barrier function.

5. Computing step directions

The overall aim is to find the optimal solution to the homogeneous embedding. This amounts to solving the nonlinear system of equations $G(v) = 0$ with the additional constraint $\kappa\tau = \mu$. The aim of most interior-point methods is to track the central path for $\mu \rightarrow 0$ until reasonably close to the optimum. In order to solve the nonlinear system Newton’s method is applied. This requires a linearization of the central path which can be seen below,

$$\begin{bmatrix} 0 \\ \Delta s \\ \Delta \kappa \end{bmatrix} - \begin{bmatrix} P & A^\top & q \\ -A & 0 & b \\ -(q + 2P\xi)^\top & -b^\top & \xi^\top P\xi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \\ \Delta \tau \end{bmatrix} = - \begin{bmatrix} d_x \\ d_z \\ d_\tau \end{bmatrix} \quad (43)$$

$$H\Delta z + \Delta s = -d_s, \quad \kappa\Delta\tau + \tau\Delta\kappa = -d_\kappa,$$

where $d = (d_x, d_z, d_\tau, d_s, d_\kappa)$ is the vector of residuals, while $\xi = \frac{x}{\tau}$ and $H \in \mathbb{R}^{m \times m}$ is a positive semidefinite block-diagonal matrix (zero on zero-cone blocks) with blocks equal to the scaling matrices⁶. Note that in order to allow for nonsymmetric cones we use H , which is in contrast to [2] that uses $\lambda \circ (W\Delta z + W^{-\top}\Delta s) = d_s$ with $H = W^\top W$.

⁶ Ignoring the sparse form of the scaling matrices that are usually required in practice. This will be explained in a later section.

We now eliminate $(\Delta s, \Delta \kappa)$ from the system resulting in

$$- \begin{bmatrix} P & A^\top & q \\ -A & H & b \\ -(q + 2P\xi)^\top & -b^\top & \xi^\top P\xi + \kappa\tau^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \\ \Delta \tau \end{bmatrix} = - \begin{bmatrix} d_x \\ d_z - d_s \\ d_\tau - d_\kappa\tau^{-1} \end{bmatrix} \quad (44)$$

$$\Delta s = -d_s - H\Delta z, \quad \Delta \kappa = -(d_\kappa + \kappa\Delta\tau)\tau^{-1}.$$

The above system can be solved by a pair of linear systems with a common coefficient matrix, meaning that it only has to be factorized once

$$\begin{bmatrix} P & A^\top \\ A & -H \end{bmatrix} \begin{bmatrix} \Delta x_1 & \Delta x_2 \\ \Delta z_1 & \Delta z_2 \end{bmatrix} = \begin{bmatrix} d_x & -q \\ -(d_z - d_s) & b \end{bmatrix} \quad (45)$$

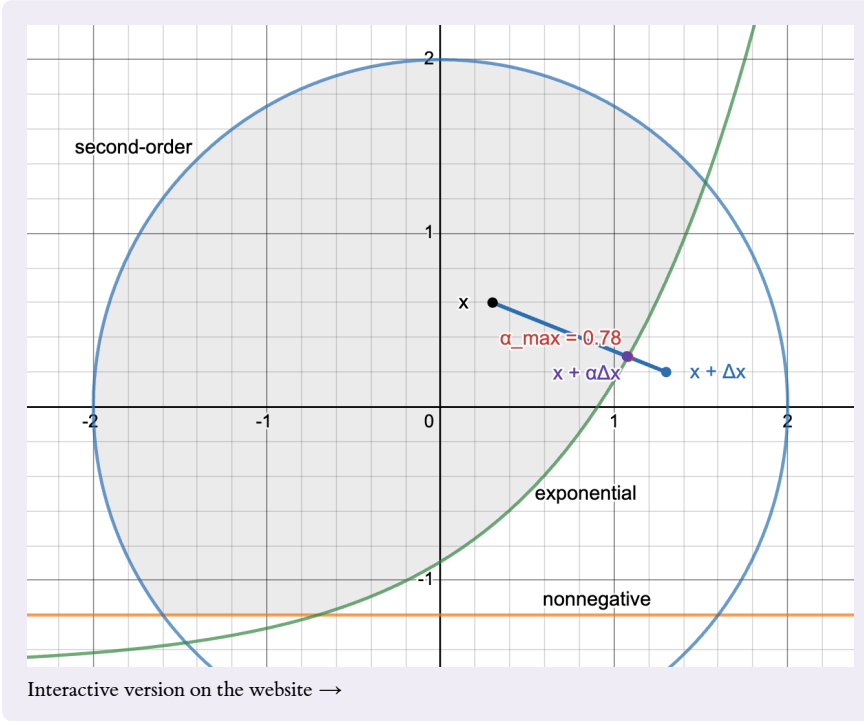
From the above solution we can recover the search direction $(\Delta x, \Delta z, \Delta s, \Delta \tau, \Delta \kappa)$ as follows

$$\begin{aligned} \Delta x &= \Delta x_1 + \Delta \tau \cdot \Delta x_2, \\ \Delta z &= \Delta z_1 + \Delta \tau \cdot \Delta z_2 \\ \Delta s &= -d_s - H\Delta z, \\ \Delta \tau &= \frac{d_\tau - d_\kappa\tau^{-1} + (2P\xi + q)^\top \Delta x_1 + b^\top \Delta z_1}{\kappa\tau^{-1} + \xi^\top P\xi - (2P\xi + q)^\top \Delta x_2 - b^\top \Delta z_2} \\ &= \frac{d_\tau - d_\kappa\tau^{-1} + q^\top \Delta x_1 + b^\top \Delta z_1 + 2\xi^\top P\Delta x_1}{\kappa\tau^{-1} + \|\Delta x_2 - \xi\|_P^2 - \|\Delta x_2\|_P^2 - q^\top \Delta x_2 - b^\top \Delta z_2} \\ \Delta \kappa &= -(d_\kappa + \kappa\Delta\tau)\tau^{-1}. \end{aligned} \quad (46)$$

The actual step is then obtained by computing the maximal step size α that ensures that the new update is still in the interior of the conic constraints.

Interactive figure 2: The largest step inside the cones

Interactive figure 2 (The largest step inside the cones) The feasible region of the central path example, with an iterate x and a search direction Δx , both of which can be dragged. Following Δx from x , the first cone boundary is reached at the step $\alpha_{\max}$ (dashed). Clarabel moves only a fraction of the way, $x + \alpha\Delta x$ with $\alpha = 0.99 \min(1, \alpha_{\max})$, so that the iterate stays strictly inside the cones.



5.1. The affine and centering directions

At every interior-point iteration we solve the KKT system for two sets of right-hand sides each, corresponding to the so-called affine and centering directions. In short this means solving Equation 45 twice, however since the system matrix stays the same and the right-hand side $(-q, b)$ stays the same the computation requires only a single numerical factorization⁷ and three solves.

The two sets of right-hand sides correspond to a **predictor** step and a **corrector** step. The steps are also known as the **affine** step and the **centering** step. In our case the affine step has the right-hand side of

$$d_x = r_x, \quad d_z = r_z, \quad d_\tau = r_\tau, \quad d_s = s, \quad d_\kappa = \kappa\tau, \quad (47)$$

where (r_x, r_z, r_τ) are the three components of the residual $G(v)$ at the current iterate v , while the corrector step has the right-hand side of

$$\begin{aligned} d_x &= (1 - \sigma)r_x, \quad d_z = (1 - \sigma)r_z, \quad d_\tau = (1 - \sigma)r_\tau, \\ d_\kappa &= \kappa\tau + \Delta\kappa\Delta\tau - \sigma\mu, \\ d_s &= \begin{cases} W^\top(\lambda \diamond (\lambda \circ \lambda - \sigma\mu e + \eta)) & \text{(symmetric)} \\ s + \sigma\mu\nabla f_*(z) + \eta & \text{(nonsymmetric),} \end{cases} \end{aligned} \quad (48)$$

where $\diamond$ denotes the inverse operator of $\circ$. Computation of a higher-order correction term η is a heuristic technique that is known to accelerate the convergence of IPMs significantly. The choice of this term varies depending on the choice of the scaling matrix H and whether a given constraint is symmetric or not. For symmetric cones we use the Mehrotra correction $\eta = (W^{-\top} \Delta s) \circ (W \Delta z)$, while for nonsymmetric cones we compute η using the 3rd-order correction from [4], i.e.

$$\eta = -\frac{1}{2} \nabla^3 f_*(z) [\Delta z, \nabla^2 f_*(z)^{-1} \Delta s]. \quad (49)$$

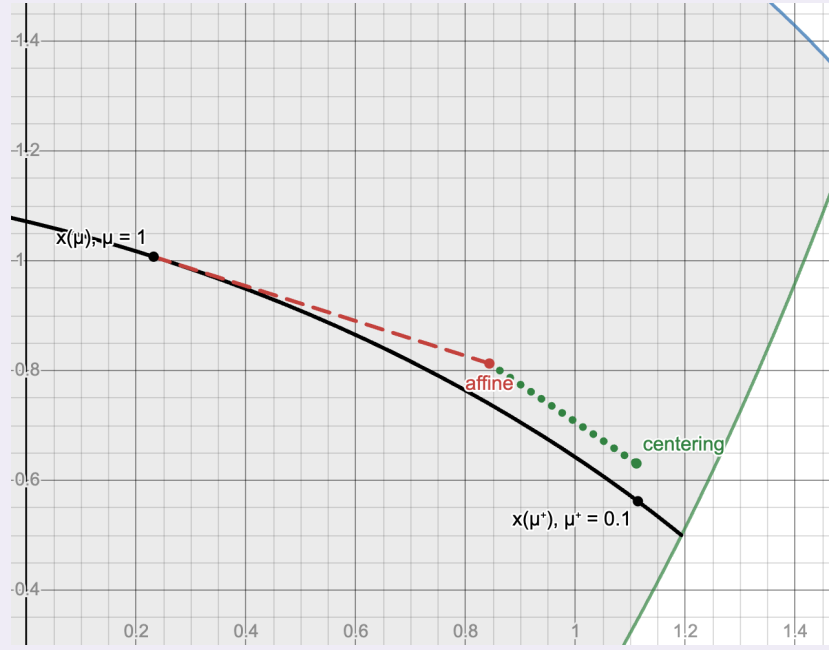
⁷ As the sparsity pattern does not change the symbolic factorization can be reused

We set $\sigma = (1 - \alpha_{\text{aff}})^3$ where α_{aff} is the step size of the affine step.

Interactive figure 3: Affine and centering directions

Interactive figure 3 (Affine and centering directions) A primal-only version on the central path example, with the dual variable tied to the iterate by $z = -\mu \nabla f(s)$, so that the step towards the central path for a target μ is the Newton step $\Delta x = -\nabla^2 F(x)^{-1}(q/\mu + \nabla F(x))$, where $F(x) = f(b - Ax)$. From $x(\mu)$, the affine step (dashed, target $\mu = 0$) is taken a fraction $\alpha/\alpha_{\text{max}}$ of the way to the boundary, which leaves the central path and reduces μ to $\mu^+ = (1 - \alpha)\mu$. The centering step (dotted) is one damped Newton step for μ^+ .⁸ Only the number μ^+ enters this step; $x(\mu^+)$ is unknown to the method and drawn for reference only. This is the classical predictor–corrector pair; Clarabel combines the two steps into one through σ .

⁸ The next iteration starts where the centering step ends, near but not on the central path.



Interactive version on the website →

5.2. Solver initialization

An essential consideration is how to choose the initial point v^0 . The initialization strategies depend on whether the objective is linear or quadratic and whether the cones are symmetric or not. However, in all cases the initial scalar values are set as $\tau^0 = \kappa^0 = 1$.

5.2.1. Symmetric cones

The initialization strategies below can be thought of as projections of the equality constraints onto cones in the direction of the idempotent e for each symmetric cone.

Quadratic objective. If $P \neq 0$ the initialization happens by solving the following linear system

$$\begin{bmatrix} P & A^\top \\ A & -I_E \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} \quad (50)$$

with $I_E = I - E$ with E being a matrix with ones on the diagonal corresponding to zero cones (i.e. equality constraints) but otherwise zero. The above linear system corresponds to solving the following optimization problem

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} x^\top P x + q^\top x + \frac{1}{2} \|Ax - b\|_2^2 \\ & \text{subject to} \quad EAx = Eb, \end{aligned} \quad (51)$$

From this solution we project z and s back to the interior of the cone

$$\begin{aligned} x^0 &= x, \\ s^0 &= \begin{cases} -z, & \alpha_p < -\epsilon \\ -z + (\epsilon + \alpha_p)e, & \text{otherwise} \end{cases}, \\ z^0 &= \begin{cases} z, & \alpha_d < -\epsilon \\ z + (\epsilon + \alpha_d)e, & \text{otherwise} \end{cases}. \end{aligned} \quad (52)$$

where $\alpha_p = \inf\{\alpha \mid -z + \alpha e \in \mathcal{K}\}$ and $\alpha_d = \inf\{\alpha \mid -s + \alpha e \in \mathcal{K}\}$.

Linear objective. In the case of $P = 0$ we instead solve two linear systems

$$\begin{bmatrix} 0 & A^\top \\ A & -I_E \end{bmatrix} \begin{bmatrix} x & t \\ s & z \end{bmatrix} = \begin{bmatrix} 0 & -q \\ b & 0 \end{bmatrix}. \quad (53)$$

From this we set

$$\begin{aligned} x^0 &= x, \\ s^0 &= \begin{cases} -s, & \alpha_p < -\epsilon \\ -s + (\epsilon + \alpha_p)e, & \text{otherwise} \end{cases}, \\ z^0 &= \begin{cases} z, & \alpha_d < -\epsilon \\ z + (\epsilon + \alpha_d)e, & \text{otherwise} \end{cases}, \end{aligned} \quad (54)$$

where $\alpha_p = \inf\{\alpha \mid -s + \alpha e \in \mathcal{K}\}$ and $\alpha_d = \inf\{\alpha \mid z + \alpha e \in \mathcal{K}\}$.

5.2.2. Nonsymmetric cones

When the composite cone $\mathcal{K}$ contains any nonsymmetric cone both primal and dual variables are set as $z = s = -\nabla f_{*(z)}$ (corresponding to $\mu^0 = 1$). This is equivalent to solving the unconstrained optimization problem

$$\text{minimize}_z \quad \frac{1}{2} \|z\|_2^2 + f_*(z), \quad (55)$$

which is strictly convex and has a unique solution. It yields $s_{\text{sym}}^0 = z_{\text{sym}}^0 = e$ for symmetric cones, and

$$s_{\text{exp}}^0 = z_{\text{exp}}^0 \approx (-1.051383, 0.556409, 1.258967), \quad (56)$$

for exponential cones, and

$$s_{\text{pow}}^0 = z_{\text{pow}}^0 \approx (\sqrt{1+\alpha}, \sqrt{2-\alpha}, 0), \quad (57)$$

for power cones with parameter α .

5.3. Termination criteria

All termination criteria are based on unscaled problem data and iterates, i.e. after the Ruiz scaling has been reverted.

5.3.1. Feasibility checks

For checks of primal and dual feasibility we introduce the normalized variables $\hat{x} = x/\tau$, $\hat{s} = s/\tau$, $\hat{z} = z/\tau$ and define the primal and dual residual as follows

$$\begin{aligned} r_p &:= A\hat{x} + \hat{s} - b & (\text{primal residual}) \\ r_d &:= P\hat{x} + A^\top \hat{z} + q & (\text{dual residual}) . \end{aligned} \quad (58)$$

Likewise, we define the primal and dual objectives as

$$\begin{aligned} g_p &:= \frac{1}{2} \hat{x}^\top P \hat{x} + q^\top \hat{x} & (\text{primal objective}) \\ g_d &:= -\frac{1}{2} \hat{x}^\top P \hat{x} - b^\top \hat{z} & (\text{dual objective}) . \end{aligned} \quad (59)$$

Using the above definitions we can declare convergence if all three of the following hold:

$$\begin{aligned} \|r_p\| &< \epsilon_f \cdot \max \{1, \|b\|_\infty + \|\hat{x}\| + \|\hat{s}\|\} & (\text{primal convergence}) \\ \|r_d\| &< \epsilon_f \cdot \max \{1, \|q\|_\infty + \|\hat{x}\| + \|\hat{z}\|\} & (\text{dual convergence}) \\ |g_p - g_d| &< \epsilon_f \cdot \max \{1, \min \{|g_p|, |g_d|\}\} & (\text{gap convergence}) \end{aligned} \quad (60)$$

We specify a default value of $\epsilon_f = 10^{-8}$ as well as a weaker threshold of $\epsilon_f = 10^{-5}$ when testing for “near optimality” in cases of early termination (e.g. lack of progress, timeout, iterations limit, etc.).

5.3.2. Infeasibility checks

When testing for infeasibility we do not normalize iterates, but rather work directly with the unscaled variables since infeasibility corresponds to the case where $\tau \rightarrow 0$. We declare primal infeasibility if the following holds

$$\begin{aligned} \|A^\top z\| &< -\epsilon_{i,r} \cdot \max \{1, \|x\| + \|z\|\} \cdot (b^\top z) & (\text{primal infeasibility}) \\ b^\top z &< -\epsilon_{i,a} & (\text{primal infeasibility}) . \end{aligned} \quad (61)$$

Similarly, we define dual infeasibility if the following hold

$$\begin{aligned} \|Px\| &< -\epsilon_{i,r} \cdot \max \{1, \|x\|\} \cdot (q^\top x) & (\text{dual infeasibility}) \\ \|Ax + s\| &< -\epsilon_{i,r} \cdot \max \{1, \|x\| + \|s\|\} \cdot (q^\top x) & (\text{dual infeasibility}) \\ q^\top x &< -\epsilon_{i,a} & (\text{dual infeasibility}) \end{aligned} \quad (62)$$

We set the relative and absolute tolerances as $\epsilon_{i,r} = \epsilon_{i,a} = 10^{-8}$ and allow for weaker thresholds to declare “near infeasibility” certificates in cases of early termination.

6. Implementation details

6.1. Solving the KKT system

The main computational effort in the interior-point method is to solve the Karush-Kuhn-Tucker (KKT) system of the form

$$\underbrace{\begin{bmatrix} P & A^\top \\ A & -H \end{bmatrix}}_K \underbrace{\begin{bmatrix} x \\ z \end{bmatrix}}_y = \underbrace{\begin{bmatrix} b_x \\ b_z \end{bmatrix}}_b, \quad (63)$$

The scaling matrix, however, has the unfortunate property that it becomes ill-conditioned when the iterates are close to the boundary. As such one instead solves the regularized problem

$$\underbrace{\begin{bmatrix} P + \epsilon I & A^\top \\ A & -(H + \epsilon I) \end{bmatrix}}_{K+D} \underbrace{\begin{bmatrix} x \\ z \end{bmatrix}}_y = \underbrace{\begin{bmatrix} b_x \\ b_z \end{bmatrix}}_b, \quad (64)$$

where D denotes the added diagonal part. Note that while solving Equation 64 is more numerically stable, it is not the system that we actually aim to solve i.e. Equation 63. The regularization ϵ is computed as

$$\epsilon = \epsilon_{\text{constant}} + \epsilon_{\text{proportion}} \parallel \text{diag}(K) \parallel_\infty. \quad (65)$$

6.2. Regularized iterative refinement

As described in [5], the regularization term can be removed by solving a sequence of systems of the form

$$(K + D)\Delta x_m = b - Kx_m, \quad x_{m+1} = x_m + \Delta x_m. \quad (66)$$

The sequence $\{x_m\}$ converges to the true solution of $Kx = b$, provided that it exists. To show this first notice that we can rewrite Equation 66 as

$$(K + D)x_{m+1} - Dx_m = b. \quad (67)$$

Now assume that x^0 is the solution to the equation $Kx^0 = b$ (which is the equation we aim to solve). Now we perform the mathematician's favorite trick of adding 0 (in this case $Dx^0 - Dx^0$), it follows that

$$(K + D)x^0 - Dx^0 = b. \quad (68)$$

We now subtract Equation 68 from Equation 67

$$(K + D)(x_{m+1} - x^0) - D(x_m - x^0) = 0. \quad (69)$$

Rewriting the above gives us

$$x_{m+1} - x^0 = (K + D)^{-1}D(x_m - x^0). \quad (70)$$

The final step is to take norms on both sides resulting in

$$\parallel x_{m+1} - x^0 \parallel \leq \parallel (K + D)^{-1}D \parallel \parallel x_m - x^0 \parallel. \quad (71)$$

Thus, if $\parallel (K + D)^{-1}D \parallel < 1$ we have that $x_m \rightarrow x^0$ for $m \rightarrow \infty$ (more precisely, it suffices that the spectral radius of $(K + D)^{-1}D$ is less than one).

6.3. Matrix equilibration

The equilibration in Clarabel is a modified version of Ruiz scaling [6], [7], similar as to what is done in the OSQP package [5]. The equilibration is based on the scaling of the constant part of the KKT system (i.e. the part excluding the primal-dual scaling matrices) as shown below

$$M = \begin{bmatrix} P & A^\top \\ A & 0 \end{bmatrix}. \quad (72)$$

The aim is to find a diagonal scaling matrix S such that

$$SMS = \begin{bmatrix} D & 0 \\ 0 & E \end{bmatrix} \begin{bmatrix} P & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & E \end{bmatrix} = \begin{bmatrix} DPD & DA^\top E \\ EAD & 0 \end{bmatrix}, \quad (73)$$

is better conditioned. The above can be seen as a transformation of input data to a conic optimization problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \bar{x}^\top \bar{P} \bar{x} + \bar{q}^\top \bar{x} \\ & \text{subject to} && \bar{A} \bar{x} + \bar{s} = \bar{b} \\ & && \bar{s} \in \mathcal{K}, \end{aligned} \quad (74)$$

where $\bar{x} = D^{-1}x$, $\bar{s} = Es$, $\bar{P} = cDPD$, $\bar{q} = cDq$, $\bar{A} = EAD$, $\bar{b} = Eb$, with $c > 0$ a cost scaling and E chosen such that $E\mathcal{K} = \mathcal{K}$.

The main idea of the equilibration procedure is to scale the rows (and columns) for the matrix M so that they have equal ℓ_p -norm. An efficient approximate solution to this problem is the so-called Modified Ruiz scaling described in [5] based on the Ruiz scaling described in [6].

7. Sparse solvers

More often than not the main computational time in an interior-point method is solving Equation 64. Luckily, the linear system is often sparse leading to fast factorization algorithms. As such it is important to use an efficient implementation of a direct sparse solver if one wants the interior-point solver to be computationally efficient. In Clarabel it is possible to utilize the following sparse solvers

- (Rust) faer-rs. Multithreaded. Wasm compatible.
- (Rust/Julia) Quasi-Definite LDL (QDLDL) based on [8]. Single threaded.
- (Rust/Julia) PanuaPardiso (Requires license). Multithreaded.
- (Rust/Julia) oneMKL-Pardiso (Only Intel CPUs). Multithreaded.
- (Julia) cuDSS (CUDA Direct Sparse Solver). GPU-based.
- (Julia) MA57 (Requires license). Multithreaded.
- (Julia) CHOLMOD (Standard Julia Solver). Multithreaded.

7.1. Sparsity exploitation

7.1.1. Augmented sparsity

In numerical computations one often meets a matrix of the form sparse-plus-low-rank. The general form of this is the following

$$H := S - \sum_{i=1}^{n_V} v_i v_i^\top + \sum_{i=1}^{n_U} u_i u_i^\top = S - VV^\top + UU^\top, \quad (75)$$

where S is sparse and $V = [v_1 \ \dots \ v_{n_V}]$ and $U = [u_1 \ \dots \ u_{n_U}]$ with $n_U, n_V \ll n$. The problem with the above matrix is that forming the outer products (i.e. assembling the plus-low-rank part) will ruin the sparsity of S and H in turn will be a dense matrix. Given that our aim is to solve a system of the form $Hx = b$ lets look at the structure of Hx by performing the product

$$Hx = Sx - V \begin{pmatrix} \overbrace{V^\top x} \\ -t_V \end{pmatrix} + U \begin{pmatrix} \overbrace{U^\top x} \\ t_U \end{pmatrix} = Sx + Vt_V + Ut_U. \quad (76)$$

The above together with the constraints $V^\top x + t_V = 0$ and $U^\top x - t_U = 0$ we can solve $Hx = b$ by solving the following equivalent system

$$\underbrace{\begin{bmatrix} S & V & U \\ V^\top & I_{n_V} & 0 \\ U^\top & 0 & -I_{n_U} \end{bmatrix}}_{H_{\text{aug}}} \begin{bmatrix} x \\ t_V \\ t_U \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}. \quad (77)$$

Note that while H_{aug} is sparse it is larger. An important property of the augmented system is that if $S - VV^\top > 0$ (and hence $S \succ 0$) then H_{aug} is Quasi-Definite and therefore strongly factorizable, meaning that for every permutation P one can find L and D so that $PH_{\text{aug}}P^\top = LDL^\top$ [9].

7.2. Second-order cones

7.2.1. The Clarabel approach

Clarabel uses the same approach as in ECOS [10]. In short, they utilize that the structure of $W_k^\top W_k$ for a second-order cone is augmented-sparse, meaning that it can be split as follows

$$W_k^\top W_k = S_k - v_k v_k^\top + u_k u_k^\top, \quad (78)$$

where $S_k - v_k v_k^\top \succ 0$. We can generalize the above for all second-order cones by defining

$$\begin{aligned} S &= \text{blkdiag}(S_1, S_2, \dots, S_n), \\ V &= \text{blkdiag}(v_1, v_2, \dots, v_n), \\ U &= \text{blkdiag}(u_1, u_2, \dots, u_n), \end{aligned} \quad (79)$$

meaning that

$$W^\top W = S - VV^\top + UU^\top. \quad (80)$$

As such the KKT of the form

$$\begin{bmatrix} P & A^\top \\ A & -W^\top W \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} b_x \\ b_z \end{bmatrix} \quad (81)$$

can equivalently be solved by solving

$$\begin{bmatrix} P & A^\top & 0 & 0 \\ A & -S & -V & -U \\ 0 & -V^\top & -I & 0 \\ 0 & -U^\top & 0 & I \end{bmatrix} \begin{bmatrix} x \\ z \\ t_V \\ t_U \end{bmatrix} = \begin{bmatrix} b_x \\ b_z \\ 0 \\ 0 \end{bmatrix}, \quad (82)$$

The approach is simple as it leaves A intact and does not scale z . However, one has to deal with computing V and U , which is not unique. In fact Clarabel and ECOS apply different approaches when calculating V and U . A detail here is that one wants to compute V and U in a way for which $S - VV^\top \succ 0$ so that the system remains quasi-definite (and therefore strongly factorizable).

7.2.2. The CVXOPT approach

Another approach is that of CVXOPT. Here the diagonal-plus-low-rank structure of the scaling matrix of a second-order cone (and its inverse) is used, i.e. that

$$W_k = D_k + u_k u_k^\top, \quad W_k^{-1} = E_k + v_k v_k^\top, \quad (83)$$

where the D_k and E_k are diagonal. We can again bunch all the scaling matrices together by defining

$$D = \text{blkdiag}(D_1, D_2, \dots, D_n), \quad E = \text{blkdiag}(E_1, E_2, \dots, E_n), \quad (84)$$

so that

$$W = D + UU^\top, \quad W^{-1} = E + VV^\top. \quad (85)$$

Note that U and V are defined as before, but are different since the u_k and v_k come from Equation 83 rather than Equation 78. We now show how the rank-update and the known inverse can be used explicitly in order to preserve sparsity. The KKT system of interest using Nesterov-Todd scaling is the following

$$\begin{bmatrix} P & A^\top \\ A & -W^\top W \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} b_x \\ b_z \end{bmatrix} \quad (86)$$

By scaling the z -variable and multiplying the last row with W^{-1} one finds that

$$\begin{bmatrix} P & A^\top(E + VV^\top) \\ (E + VV^\top)A & -I \end{bmatrix} \begin{bmatrix} x \\ Wz \end{bmatrix} = \begin{bmatrix} b_x \\ W^{-1}b_z \end{bmatrix}. \quad (87)$$

For simplicity, we now define $S = A^\top V$

$$\begin{bmatrix} P & A^\top E + SV^\top \\ EA + VS^\top & -I \end{bmatrix} \begin{bmatrix} x \\ Wz \end{bmatrix} = \begin{bmatrix} b_x \\ W^{-1}b_z \end{bmatrix}. \quad (88)$$

We can now introduce new variables $\alpha, \beta \in \mathbb{R}^{n_v}$ resulting in

$$\begin{bmatrix} P & A^\top E & S & 0 \\ E^\top A & -I & 0 & V \\ S^\top & 0 & 0 & -I \\ 0 & V^\top & -I & 0 \end{bmatrix} \begin{bmatrix} x \\ Wz \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} b_x \\ W^{-1}b_z \\ 0 \\ 0 \end{bmatrix}. \quad (89)$$

Where we have used that $E = E^\top$ to show why the system is also symmetric. Now the above preserves sparsity. For SOCs, $E_k = -\gamma_k^{-1}J$ for a scalar $\gamma_k > 0$ (in CVXOPT, $W_k = \gamma_k(2w_k w_k^\top - J)$ for a vector w_k), so EE is diagonal with blocks $\gamma_k^{-2}I$. If the scaling is normalized so that $\gamma_k = 1$, then $EE = I$, and we can reduce the above as

$$\begin{bmatrix} P & A^\top & S & 0 \\ A & -I & 0 & EV \\ S^\top & 0 & 0 & -I \\ 0 & V^\top E^\top & -I & 0 \end{bmatrix} \begin{bmatrix} x \\ EWz \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} b_x \\ EW^{-1}b_z \\ 0 \\ 0 \end{bmatrix}. \quad (90)$$

7.3. Rotated second-order cones

By rotating the rotated second-order cone back to the second-order cone form we can exploit the sparsity similarly to the regular second-order cone. This rotation can be done by applying the following transformation

$$R = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \dots & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \quad (91)$$

7.4. Power cones

The conjugate barrier for certain power cones has a Hessian that, similar to that of second-order cones, can be represented using augmented sparsity [11]. The intuition here is that e.g. the generalized power cone reduces to a second-order cone when $d_1 = 1$. Given that the augmented sparsity is only available for the conjugate barrier the following is only relevant in the case of dual scaling.

The generalized power cone is parametrized by $\alpha \in \mathbb{R}_{++}^{d_1}$ such that

$$\sum_{i \in [d_1]} \alpha_i = 1 \quad (92)$$

and is defined as

$$\mathcal{K}_{\text{gpow}}(\alpha, d_1, d_2) = \left\{ (u, w) \in \mathbb{R}_+^{d_1} \times \mathbb{R}^{d_2} \mid \prod_{i=1, \dots, d_1} u_i^{\alpha_i} \geq \|w\| \right\} \quad (93)$$

with dual cone

$$\mathcal{K}_{\text{gpow}}^*(\alpha, d_1, d_2) = \left\{ (u, w) \in \mathbb{R}_+^{d_1} \times \mathbb{R}^{d_2} \mid \prod_{i=1, \dots, d_1} \left(\frac{u_i}{\alpha_i} \right)^{\alpha_i} \geq \|w\| \right\} \quad (94)$$

The power mean cone is parametrized by $\alpha \in \mathbb{R}_{++}^{d_1}$ such that

$$\sum_{i \in [d_1]} \alpha_i = 1 \quad (95)$$

and is defined as

$$\mathcal{K}_{\text{powm}}(\alpha, d_1) = \left\{ (u, w) \in \mathbb{R}_+^{d_1} \times \mathbb{R} \mid \prod_{i=1, \dots, d_1} u_i^{\alpha_i} \geq w \right\} \quad (96)$$

with dual cone

$$\mathcal{K}_{\text{powm}}^*(\alpha, d_1) = \left\{ (u, w) \in \mathbb{R}_+^{d_1} \times \mathbb{R}_- \mid \prod_{i=1, \dots, d_1} \left(\frac{u_i}{\alpha_i} \right)^{\alpha_i} \geq -w \right\} \quad (97)$$

The relative entropy cone is defined as

$$\mathcal{K}_{\text{rel}}(d) = \text{cl} \left\{ (u, v, w) \in \mathbb{R} \times \mathbb{R}_{++}^d \times \mathbb{R}_{++}^d \mid u \geq \sum_{i=1, \dots, d} w_i \ln \left(\frac{w_i}{v_i} \right) \right\} \quad (98)$$

with dual cone

$$\begin{aligned} \mathcal{K}_{\text{rel}}^*(d) = \text{cl} \left\{ (u, v, w) \in \mathbb{R}_{++} \times \mathbb{R}_{++}^d \times \mathbb{R}^d \mid \right. \\ \left. w_i \geq u \left(\ln \left(\frac{u}{v_i} \right) - 1 \right) \forall i = 1, \dots, d \right\} \end{aligned} \quad (99)$$

7.5. Generalized power cones

As shown in [11] the Hessian of the conjugate barrier for the generalized power cones is augmented sparse with $n_u = 1$ and $n_v = 2$, i.e. that

$$H^*(z) = D + pp^\top - qq^\top - rr^\top. \quad (100)$$

where $z := (u, w)$ and $D - qq^\top - rr^\top \succ 0$ with

$$D = \begin{bmatrix} \ddots & & & \\ & \frac{\tau_i \phi}{u_i \zeta} + \frac{1-\alpha_i}{u_i^2} & & \\ & & \ddots & \\ & & & \frac{2}{\zeta} I_{d_2} \end{bmatrix}, p = \begin{bmatrix} p_0 \frac{\tau}{\zeta} \\ p_1 \frac{w}{\zeta} \end{bmatrix}, q = \begin{bmatrix} q_0 \frac{\tau}{\zeta} \\ 0 \end{bmatrix}, r = \begin{bmatrix} 0 \\ r_1 \frac{w}{\zeta} \end{bmatrix}, \quad (101)$$

where

$$\begin{aligned} p_0 &= \sqrt{\frac{\phi(\phi + \|w\|^2)}{2}}, \quad p_1 = -2\sqrt{\frac{2\phi}{\phi + \|w\|^2}} \\ q_0 &= \sqrt{\frac{\phi\zeta}{2}}, \quad r_1 = 2\sqrt{\frac{\zeta}{\phi + \|w\|^2}} \end{aligned} \quad (102)$$

and $\phi = \prod_{i \in [d_1]} \left(\frac{u_i}{\alpha_i} \right)^{2\alpha_i}$, $\tau_i = \frac{2\alpha_i}{u_i}$ and $\zeta = \phi - \|w\|^2$.

7.6. Power mean cone

For the power mean cone it is shown in [11] that the Hessian of the conjugate barrier is augmented sparse with $n_u = 1$ and $n_v = 2$, i.e. that

$$H^*(z) = D + pp^\top - qq^\top - rr^\top. \quad (103)$$

where $z := (u, w)$ and $D - qq^\top - rr^\top \succ 0$ with

$$D = \begin{bmatrix} \ddots & & & \\ & \frac{\tau_i \phi}{u_i} + \frac{1-\alpha_i}{u_i^2} & & \\ & & \ddots & \\ & & & \theta \end{bmatrix}, p = \begin{bmatrix} p_0 \tau \\ p_1 \frac{1}{\zeta} \end{bmatrix}, q = \begin{bmatrix} q_0 \tau \\ 0 \end{bmatrix}, r = \begin{bmatrix} 0 \\ r_1 \frac{1}{\zeta} \end{bmatrix}, \quad (104)$$

where

$$\theta = 1 + w^{-2}, \quad p_0 = \phi, \quad p_1 = 1, \quad q_0 = \sqrt{\zeta \phi}, \quad r_1 = \zeta \quad (105)$$

and

$$\phi = \prod_{i \in [d_1]} \left(\frac{u_i}{\alpha_i} \right)^{\alpha_i}, \quad \zeta = \phi + w \text{ and } \tau_i = \frac{\alpha_i}{u_i \zeta}, i = 1, \dots, d.$$

7.7. Relative entropy cone

For the relative entropy cone it is shown in [11] that the Hessian of the conjugate barrier is sparse with the following structure

$$H^*(z) = \begin{bmatrix} \nabla_{u,u}^2 f_* & \nabla_{u,v}^2 f_* & \nabla_{u,w}^2 f_* \\ \nabla_{v,u}^2 f_* & D_{v,v} & D_{v,w} \\ \nabla_{w,u}^2 f_* & D_{w,v} & D_{w,w} \end{bmatrix} \quad (106)$$

where $z := (u, v, w)$ and $H^*(z) \succ 0$

$$\begin{aligned} D_{v,v} &= \text{diag} \left(\left[\frac{u(\gamma_1 + u)}{\gamma_1^2 v_1^2} + \frac{1}{v_1^2} \quad \dots \quad \frac{u(\gamma_d + u)}{\gamma_d^2 v_d^2} + \frac{1}{v_d^2} \right] \right) \\ D_{w,w} &= \text{diag}([\gamma_1^{-2} \quad \dots \quad \gamma_d^{-2}]) \\ D_{v,w} &= D_{w,v} = \text{diag}([u(\gamma_1^2 v_1)^{-1} \quad \dots \quad u(\gamma_d^2 v_d)^{-1}]), \end{aligned} \quad (107)$$

with $\gamma_i = w_i - u \left(\ln \left(\frac{u}{v_i} \right) - 1 \right) > 0, i = 1, \dots, d$.

7.8. Semidefinite cones

We start by introducing the vectorization of a symmetric matrix M defined below

$$\begin{aligned} & [M_{1,1}, \sqrt{2}M_{2,1}, M_{2,2}, \sqrt{2}M_{3,1}, \dots, \\ \text{svec}(M) &:= \sqrt{2}M_{n,n-2}, \sqrt{2}M_{n,n-1}, M_{n,n}]^\top. \end{aligned} \quad (108)$$

Note that the scaling of $\sqrt{2}$ is in order to preserve inner-products i.e. that $\text{tr}(L^\top M) = \text{svec}(L)^\top \text{svec}(M)$. We equivalently introduce the **matrization** of a vector (and denote it by smat) which unsurprisingly has the property $M = \text{smat}(\text{svec}(M))$ as

$$\text{smat}(m) := \begin{bmatrix} m_1 & m_2/\sqrt{2} & \dots & m_{n(n-1)/2+1}/\sqrt{2} \\ m_2/\sqrt{2} & m_3 & \dots & m_{n(n-1)/2+2}/\sqrt{2} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n(n-1)/2+1}/\sqrt{2} & m_{n(n-1)/2+2}/\sqrt{2} & \dots & m_{n(n+1)/2} \end{bmatrix}. \quad (109)$$

Note that the scaling ensures that we preserve inner-products i.e. that $u^\top v = \text{tr}(\text{smat}(u) \text{smat}(v))$.

7.8.1. Computing dense $W_k^\top W_k$

In CVXOPT the Nesterov-Todd scaling of a semidefinite cone is described by the product

$$W_k \text{svec}(V) = \text{svec}(R_k^\top V R_k), \quad (110)$$

However, stating it like this gives no insights into the structure of W_k , which is needed when assembling the KKT system (to be precise $W_k^\top W_k$ is needed for the KKT assembly). It turns out that the structure comes from the symmetric Kronecker product given by

$$(G \otimes_s H) \text{svec}(V) = \frac{1}{2} \text{svec}(HVG^\top + GVH^\top). \quad (111)$$

Using the above we can see that this means that the Nesterov-Todd scaling of the semidefinite cone is

$$(R^\top \otimes_s R^\top) \text{svec}(V) = \text{svec}(R^\top V R), \quad (112)$$

from which it becomes clear that $W_k = (R_k^\top \otimes_s R_k^\top)$. We're now almost ready to describe $W_k^\top W_k$, but first we need to introduce the following property of the symmetric Kronecker product [12]

$$(A \otimes_s B)(C \otimes_s D) = \frac{1}{2}(AC \otimes_s BD + AD \otimes_s BC). \quad (113)$$

Using the above it follows that

$$\begin{aligned} W_k^\top W_k &= (R_k \otimes_s R_k)(R_k^\top \otimes_s R_k^\top) \\ &= R_k R_k^\top \otimes_s R_k R_k^\top \end{aligned} \quad (114)$$

Now how do we write the symmetric Kronecker product as an explicit matrix? First we need to introduce a mapping Q that works as follows

$$Q \text{vec}(S) = \text{svec}(S), \quad Q^\top \text{svec}(S) = \text{vec}(S) \quad (115)$$

Using Q it is now possible to describe the symmetric Kronecker product as

$$G \otimes_s H = \frac{1}{2}Q(G \otimes H + H \otimes G)Q^\top \quad (116)$$

This definition intuitively makes sense. The $Q^\top$ on the right is applied first and lifts a svec to vec i.e. from a symmetric vector to a regular vector. We then apply the sum of regular Kronecker products (the sum makes it map symmetric matrices to symmetric matrices) and then pull the product down back to svec using Q . Note that in general $G \otimes_s H$ is not symmetric. Instead, it represents a single half of an otherwise symmetric product. In the case where $H = G$, as is the case of NT-scaling, the product is symmetric and in fact

$$W_k^\top W_k = R_k R_k^\top \otimes_s R_k R_k^\top = Q(R_k R_k^\top \otimes R_k R_k^\top)Q^\top \quad (117)$$

Bibliography

- [1] P. J. Goulart and Y. Chen, “Clarabel: An interior-point solver for conic programs with quadratic objectives.” [Online]. Available: <https://arxiv.org/abs/2405.12762>
- [2] L. Vandenberghe, “The CVXOPT linear and quadratic cone program solvers.” [Online]. Available: <https://www.seas.ucla.edu/~vandenbe/publications/coneprog.pdf>
- [3] L. Tunçel, “Generalization of Primal-Dual Interior-Point Methods to Convex Optimization Problems in Conic Form,” *Foundations of Computational Mathematics*, vol. 1, no. 3, pp. 229–254, 2001, doi: 10.1007/s002080010009.
- [4] J. Dahl and E. D. Andersen, “A primal-dual interior-point algorithm for non-symmetric exponential-cone optimization,” *Mathematical Programming*, 2022, doi: 10.1007/s10107-021-01631-4.
- [5] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, “OSQP: an operator splitting solver for quadratic programs,” *Mathematical Programming Computation*, vol. 12, no. 4, 2020, doi: 10.1007/s12532-020-00179-2.
- [6] D. Ruiz, “A scaling algorithm to equilibrate both rows and columns norms in matrices.” [Online]. Available: https://cerfacs.fr/wp-content/uploads/2017/06/14_DanielRuiz.pdf
- [7] R. Takapoui and H. Javadi, “Preconditioning via Diagonal Scaling.” [Online]. Available: <https://arxiv.org/abs/1610.03871>
- [8] T. A. Davis, “Algorithm 849: A concise sparse Cholesky factorization package,” *ACM Transactions on Mathematical Software*, vol. 31, no. 4, pp. 587–591, 2005, doi: 10.1145/1114268.1114277.
- [9] R. J. Vanderbei, “Symmetric Quasidefinite Matrices,” *SIAM Journal on Optimization*, 1995, doi: 10.1137/0805005.
- [10] A. Domahidi, E. Chu, and S. Boyd, “ECOS: An SOCP solver for embedded systems,” in *European Control Conference (ECC)*, EUCA, 2013, pp. 3071–3076. doi: 10.23919/ECC.2013.6669541.

- [11] Y. Chen and P. Goulart, "An Efficient IPM Implementation for A Class of Nonsymmetric Cones." [Online]. Available: <https://arxiv.org/abs/2305.12275>
- [12] K. Schacke, "On the Kronecker Product." [Online]. Available: <https://www.math.uwaterloo.ca/~hwoikowi/henry/reports/kronthesisschaecke04.pdf>