# Sum of squares generalizations for conic sets

## Numerical example

Lea Kapelevich, Chris Coey, Juan Pablo Vielma

December 17, 2021

For each cone ($K_{\text{SOSPSD}}, K_{\text{SOS}\,\ell_2}, K_{\text{SOS}\,\ell_1}$) we compare the computational time to solve a simple example with its SOS formulation. We use an example analogous to the polynomial envelope problem from [4, Section 7.2], but replace the nonnegativity constraint by a conic inequality. Let $q_{i \in [\![2..m]\!]}(\mathbf{x})$ be randomly generated polynomials in $\mathbb{R}_{n,2d_r}[\mathbf{x}]$. We seek a polynomial that gives the tightest approximation to the $\ell_1$ or $\ell_2$ norm of $(q_2(\mathbf{x}), \dots, q_m(\mathbf{x}))$ for all $\mathbf{x} \in [-1,1]^n$:

$$\min_{q_1(\mathbf{x}) \in \mathbb{R}_{n,2d}[\mathbf{x}]} \int_{[-1,1]^n} q_1(\mathbf{x}) d\mathbf{x} : \tag{1a}$$

$$q_1(\mathbf{x}) \geq ||(q_2(\mathbf{x}), \dots, q_m(\mathbf{x}))||_p \qquad \forall \mathbf{x} \in [-1,1]^n, \tag{1b}$$

with $p \in \{1, 2\}$ in Equation (1b).

To restrict Equation (1b) over $[-1,1]^n$, we use *weighted sum of squares* (WSOS) formulations. A polynomial $q(\mathbf{x})$ is WSOS with respect to weights $g_{i \in [\![1..K]\!]}(\mathbf{x})$ if it can be expressed in the form of $q(\mathbf{x}) = \sum_{i \in [\![1..K]\!]} g_i(\mathbf{x}) p_i(\mathbf{x})$, where $p_{i \in [\![1..K]\!]}(\mathbf{x})$ are SOS. Papp and Yildiz [4, Section 6] show that the dual WSOS cone (we will write $K^*_{\text{WSOS}}$) may be represented by an intersection of $K^*_{\text{SOS}}$ cones. We represent the dual *weighted* cones $K^*_{\text{WSOSPSD}}$, $K^*_{\text{WSOS}\,\ell_2}$ and $K^*_{\text{WSOS}\,\ell_1}$ analogously using intersections of $K^*_{\text{SOSPSD}}$, $K^*_{\text{SOS}\,\ell_2}$ and $K^*_{\text{SOS}\,\ell_1}$ respectively.

Let $\mathbf{f}_{i \in [\![1..m]\!]}$ denote the coefficients of $q_{i \in [\![1..m]\!]}(\mathbf{x})$ and let $\mathbf{w} \in \mathbb{R}^U$ be a vector of quadrature weights on $[-1,1]^n$. A low dimensional representation of Equation (1) may be written as:

$$\min_{\mathbf{f}_1 \in \mathbb{R}^U} \mathbf{w}^\top \mathbf{f}_1 : \quad (\mathbf{f}_1, \dots, \mathbf{f}_m) \in K, \tag{2}$$

where $K$ is $K_{\text{WSOS}\,\ell_2}$ or $K_{\text{WSOS}\,\ell_1}$. If $p = 2$, we compare the $K_{\text{WSOS}\,\ell_2}$ formulation with two alternative formulations involving $K_{\text{Arw SOSPSD}}$. We use either $K_{\text{WSOSPSD}}$ to model $K_{\text{Arw SOSPSD}}$, or $K_{\text{WSOS}}$. For $p = 1$, we build an SOS formulation by replacing (2) with:

$$\min_{\mathbf{f}_1, \mathbf{g}_2, \dots, \mathbf{g}_m, \mathbf{h}_2, \dots, \mathbf{h}_m \in \mathbb{R}^U} \mathbf{w}^\top \mathbf{f}_1 : \tag{3a}$$

$$\mathbf{f}_1 - \textstyle\sum_{i \in [\![2..m]\!]} (\mathbf{g}_i + \mathbf{h}_i) \in K_{\text{WSOS}}, \tag{3b}$$

$$\mathbf{f}_i - \mathbf{g}_i + \mathbf{h}_i = 0, \quad \mathbf{g}_i, \mathbf{h}_i \in K_{\text{WSOS}} \qquad \forall i \in [\![2..m]\!]. \tag{3c}$$

We select interpolation points using a heuristic adapted from [4, 5]. We uniformly sample $N$ interpolation points, where $N \gg U$. We form a Vandermonde matrix of the same structure as the matrix $\mathbf{P}$ used to construct the lifting operator, but using the $N$ sampled points for rows. We perform a QR factorization and use the first $U$ indices from the permutation vector of the factorization to select $U$ out of $N$ rows to keep.

All experiments are performed on hardware with an AMD Ryzen 9 3950X 16-Core Processor (32 threads) and 128GB of RAM, running Ubuntu 20.10, and Julia 1.8 [1]. Optimization models are built using JuMP [3] and solved with Hypatia 0.5.3 [2] using our specialized, predefined cones. Scripts we use to run our experiments and raw results are available in the Hypatia repository.[1] We use default settings in Hypatia and set relative optimality and feasibility tolerances to $10^{-7}$.

In Tables 1 and 2, we show Hypatia's termination status, number of iterations, and solve times for $n \in \{1, 4\}$ and varying values of $d_r$ and $m$. The termination status (*st*) columns of Tables 1 and 2 use the following codes to classify solve runs:

**co** the solver claims the primal-dual certificate returned is optimal given its numerical tolerances,

**tl** a limit of 1800 seconds is reached,

**rl** a limit of approximately 120GB of RAM is reached,

**sp** the solver terminates due to slow progress during iterations,

**er** the solver reports a different numerical error,

**sk** we skip the instance because the solver reached a time or RAM limit on a smaller instance.

If $p = 1$, we let $d = d_r$, where the maximum degree of $q_1(\mathbf{x})$ is $2d$. If $p = 2$, we vary $d \in \{d_r, 2d_r\}$ and add an additional column *obj* in Table 1 to show the ratio of the objective value under the $K_{\mathrm{WSOS}}$ (or equivalently $K_{\mathrm{WSOSPSD}}$) formulation divided by the objective value under the $K_{\mathrm{WSOS}\,\ell_2}$ formulation. Note that in our setup, the dimension of $K_{\mathrm{WSOS}\,\ell_2}$ only depends on $d$. A more flexible implementation could allow polynomial components to have different degrees in $K_{\mathrm{WSOS}\,\ell_2}$ for the $d = 2d_r$ case.

For $p = 2$ and $d = 2d_r$, the difference in objective values between $K_{\mathrm{WSOS}\,\ell_2}$ and alternative formulations is less than 1% across all converged instances. For $p = 2$ and $d = d_r$, the difference in the objective values is around 10–43% across converged instances. However, the solve times for $K_{\mathrm{WSOS}\,\ell_2}$ with $d = 2d_r$ are sometimes faster than the solve times of alternative formulations with $d = d_r$ and equal values of $n$, $m$, and $d_r$. This suggests that it may be beneficial to use $K_{\mathrm{WSOS}\,\ell_2}$ in place of SOS formulations, but with higher maximum degree in the $K_{\mathrm{WSOS}\,\ell_2}$ cone. The solve times using $K_{\mathrm{WSOSPSD}}$ are slightly faster than the solve times using $K_{\mathrm{WSOS}}$. For the case where $p = 1$, the $K_{\mathrm{WSOS}\,\ell_1}$ formulation is faster than the $K_{\mathrm{WSOS}}$ formulation, particularly for larger values of $m$. We also observe that the number of iterations the algorithm takes for $K_{\mathrm{WSOS}\,\ell_2}$ compared to alternative formulations varies, but larger for $K_{\mathrm{WSOS}\,\ell_1}$ compared to the alternative SOS formulation.

[1] Instructions and scripts for reproducing our experiments are available at `https://github.com/chriscoey/Hypatia.jl/tree/master/benchmarks/natvsext`.

| $n$ | $d_r$ | $m$ | $d$ | $K_{\mathrm{SOS}\,\ell_2}$ st | iter | time | $K_{\mathrm{SOS}}$ st | iter | time | $K_{\mathrm{SOSPSD}}$ st | iter | time | obj |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 20 | 4 | 20 | co | 13 | 0.1 | co | 17 | 0.4 | co | 13 | 0.2 | 0.89 |
| | | | 40 | co | 16 | 0.2 | co | 19 | 1.8 | co | 15 | 1.1 | 0.99 |
| | | 8 | 20 | co | 13 | 0.1 | co | 17 | 2.9 | co | 14 | 2.1 | 0.85 |
| | | | 40 | co | 19 | 0.7 | co | 21 | 18.0 | co | 16 | 10.0 | 1.00 |
| | | 16 | 20 | co | 14 | 0.4 | co | 19 | 48.0 | co | 14 | 27.0 | 0.80 |
| | | | 40 | co | 21 | 2.4 | co | 20 | 264.0 | co | 17 | 188.0 | 1.00 |
| | | 32 | 20 | co | 15 | 1.6 | co | 22 | 1189.0 | co | 17 | 843.0 | 0.78 |
| | | | 40 | co | 23 | 13.0 | tl | 3 | 2033.0 | tl | 7 | 2075.0 | 0.03 |
| | | 64 | 20 | co | 17 | 8.5 | rl | ∗ | ∗ | rl | ∗ | ∗ | ∗ |
| | | | 40 | co | 20 | 59.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| | 40 | 4 | 40 | co | 14 | 0.2 | co | 17 | 1.4 | co | 14 | 1.0 | 0.89 |
| | | | 80 | co | 19 | 1.0 | co | 19 | 7.7 | co | 17 | 6.2 | 0.99 |
| | | 8 | 40 | co | 16 | 0.6 | co | 19 | 15.0 | co | 15 | 9.1 | 0.82 |
| | | | 80 | co | 21 | 3.1 | co | 21 | 93.0 | co | 17 | 62.0 | 1.00 |
| | | 16 | 40 | co | 17 | 2.0 | co | 20 | 246.0 | co | 16 | 152.0 | 0.79 |
| | | | 80 | co | 27 | 13.0 | co | 21 | 1737.0 | co | 18 | 1206.0 | 1.00 |
| | | 32 | 40 | co | 18 | 7.6 | tl | 3 | 2031.0 | tl | 8 | 1803.0 | 0.02 |
| | | | 80 | co | 27 | 53.0 | rl | ∗ | ∗ | rl | ∗ | ∗ | ∗ |
| | | 64 | 40 | co | 19 | 36.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| | | | 80 | co | 26 | 226.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| 4 | 2 | 4 | 2 | co | 13 | 0.2 | co | 18 | 0.9 | co | 15 | 0.6 | 0.75 |
| | | | 4 | co | 21 | 33.0 | co | 43 | 133.0 | co | 37 | 97.0 | 1.00 |
| | | 8 | 2 | co | 13 | 0.4 | co | 21 | 11.0 | co | 18 | 7.7 | 0.64 |
| | | | 4 | co | 21 | 102.0 | tl | 49 | 1816.0 | tl | 60 | 1811.0 | 1.00 |
| | | 16 | 2 | co | 15 | 2.3 | co | 30 | 242.0 | co | 25 | 203.0 | 0.59 |
| | | | 4 | co | 21 | 437.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| | | 32 | 2 | co | 15 | 10.0 | tl | 6 | 1848.0 | tl | 10 | 1972.0 | 15.00 |
| | | | 4 | co | 22 | 1707.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| | | 64 | 2 | co | 15 | 46.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| | | | 4 | tl | 10 | 1935.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| | 4 | 4 | 4 | co | 17 | 11.0 | co | 30 | 114.0 | co | 27 | 93.0 | 0.69 |
| | | | 8 | tl | 10 | 1840.0 | rl | ∗ | ∗ | tl | ∗ | ∗ | ∗ |
| | | 8 | 4 | co | 18 | 42.0 | co | 34 | 1494.0 | co | 29 | 1111.0 | 0.58 |
| | | 16 | 4 | co | 18 | 174.0 | rl | ∗ | ∗ | tl | ∗ | ∗ | ∗ |
| | | 32 | 4 | co | 16 | 580.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |
| | | 64 | 4 | tl | 10 | 1853.0 | sk | ∗ | ∗ | sk | ∗ | ∗ | ∗ |

Table 1: Solve time in seconds and number of iterations (iter) for instances with $p = 2$.

| | | | $K_{\mathrm{SOS}\,\ell_1}$ | | | $K_{\mathrm{SOS}}$ | | |
|---|---|---|---|---|---|---|---|---|
| $n$ | $d$ | $m$ | st | iter | time | st | iter | time |
| 1 | 40 | 8 | co | 17 | 0.5 | co | 15 | 0.5 |
| | | 16 | co | 21 | 1.3 | co | 15 | 1.9 |
| | | 32 | co | 25 | 3.2 | co | 15 | 11.0 |
| | | 64 | co | 29 | 7.6 | co | 17 | 87.0 |
| | | 128 | co | 32 | 17.0 | co | 18 | 610.0 |
| | 80 | 8 | co | 21 | 2.6 | co | 18 | 2.6 |
| | | 16 | co | 24 | 5.6 | co | 17 | 13.0 |
| | | 32 | co | 27 | 13.0 | co | 18 | 89.0 |
| | | 64 | co | 31 | 31.0 | co | 18 | 600.0 |
| | | 128 | co | 38 | 83.0 | tl | $*$ | $*$ |
| 4 | 2 | 8 | co | 17 | 0.5 | co | 17 | 0.4 |
| | | 16 | co | 18 | 1.0 | co | 16 | 1.3 |
| | | 32 | co | 24 | 2.8 | co | 17 | 7.8 |
| | | 64 | co | 27 | 6.4 | co | 17 | 57.0 |
| | | 128 | co | 30 | 14.0 | co | 17 | 400.0 |
| | 4 | 8 | co | 25 | 28.0 | co | 21 | 54.0 |
| | | 16 | co | 28 | 86.0 | co | 22 | 318.0 |
| | | 32 | co | 29 | 198.0 | tl | 9 | 1823.0 |
| | | 64 | co | 31 | 423.0 | sk | $*$ | $*$ |
| | | 128 | co | 42 | 1210.0 | sk | $*$ | $*$ |

Table 2: Solve time in seconds and number of iterations (iter) for instances with $p = 1$.

## References

[1] Bezanson J, Edelman A, Karpinski S, Shah VB (2017) Julia: A fresh approach to numerical computing. SIAM review 59(1):65–98

[2] Coey C, Kapelevich L, Vielma JP (2021) Solving natural conic formulations with Hypatia.jl. 2005.01136

[3] Lubin M, Dunning I (2015) Computing in Operations Research using Julia. INFORMS Journal on Computing 27(2):238–248, DOI 10.1287/ijoc.2014.0623

[4] Papp D, Yildiz S (2019) Sum-of-squares optimization without semidefinite programming. SIAM Journal on Optimization 29(1):822–851

[5] Sommariva A, Vianello M (2009) Computing approximate Fekete points by QR factorizations of Vandermonde matrices. Computers & Mathematics with Applications 57(8):1324–1336