A Extensions

We explore additional extensions of GloptiNets that further enhance its appeal. We first describe a block diagonal structure for the model for faster evaluation, a theoretical splitting scheme for optimization, and finally a warm start scheme.

A.1 Block diagonal structure for efficient computation

Without any further assumption, we see that a model from definition 1 can be evaluated in $O(rmd)$ time; its Fourier coefficient given by lemma 1 in $O(m^2 d)$; the bound on the RKHS norm is computed in $O(m^3 d)$ time thanks to lemma 2; all that enables to compute a certificate, as stated in theorem 2, in $O(N(m^2 d) + m^3)$ time, where N is the number of frequencies sampled. If the function f to be minimized has big $\mathcal{H}_\mathbf{s}$ norm, we might need a large model size m to have $f - f_\star \approx g$. Hence, we introduce specific structure on G which makes it *block-diagonal* and *better conditioned*, so that the complexity on m and the positivity constraint are alleviated.

Proposition 2 (Block-diagonal PSD model)**.** *Let g be a PSD model as in definition 1, with $m = bs$ anchors. Split them into b groups, denoting them $\mathbf{z_{ij}}$, $i \in [b]$ and $j \in [s]$. Compute the Cholesky factorization of each kernel matrix $T_i^\top T_i = K_{\mathbf{z}_i} \in \mathbb{R}^{s\times s}$. Then, define G as a block-diagonal matrix, with b blocks defined as $G_i = \tilde{R}_i \tilde{R}_i^\top$, $\tilde{R}_i = T^{-1} R_i$, and $R_i \in \mathbb{R}^{r\times s}$. Equivalently,*

$$G = \begin{pmatrix} \tilde{R}_1 \tilde{R}_1^\top & & \\ & \ddots & \\ & & \tilde{R}_b \tilde{R}_b^\top \end{pmatrix}, \quad \textit{s.t.} \;\; g(x) = \sum_{i=1}^{b} \left\| R_i^\top K_{\mathbf{z_i}}(x) \right\|^2, \;\; K_{\mathbf{z_i}}(x) = K(\mathbf{z}_{ij}, x)_{1\le j\le s}. \tag{21}$$

Then g can be evaluated in $O(rbs^3 d)$ time, $\widehat{g}_\omega$ in $O(bs^2(r+d+s))$ time, and $\|g\|^2_{\mathcal{S}(\mathcal{H}_s)}$ in $O(b^2(rs^2 + r^2 s) + bs^3)$ time. The model has $(r+d)bs$ real parameters.

Proof. Having G defined as such, it is psd, of rank at most $rb \le sb = m$. Written $g(x) = \sum_{i=1}^{b} \|\tilde{R}_i^\top K_{\mathbf{z_i}}(x)\|^2$, we can compute the Fourier coefficient by applying lemma 1 to each of the b component. Adding the cost of computing $G_i = \tilde{R}_i \tilde{R}_i^\top$ results in complexity of $O(bs^2(r+d+s))$. Finally, note that $\|g\|^2_{\mathcal{S}(\mathcal{H}_s)} = \|A\|^2_{\mathcal{S}(\mathcal{H}_s)}$ where

$$A = ((\varphi(\mathbf{z}_{1j}))_{j\in[s]}, \ldots, (\varphi(\mathbf{z}_{bj}))_{j\in[s]})(\operatorname{Diag} G_i)_{i\in[b]}((\varphi(\mathbf{z}_{1j}))_{j\in[s]}, \ldots, (\varphi(\mathbf{z}_{bj}))_{j\in[s]})^*.$$

Then, defining Q the matrix of $b \times b$ blocks of size $s \times s$ s.t. for $j, k \in [b]$, $Q_{jk} = K(\mathbf{z}_j, \mathbf{z}_k) \in \mathbb{R}^{s\times s}$, we have

$$\|A\|^2_{\mathcal{S}(\mathcal{H}_s)} = \operatorname{Tr} Q(\operatorname{Diag} G_i)_{i\in[b]} Q(\operatorname{Diag} G_i)_{i\in[b]} = \sum_{j,k=1}^{b} \operatorname{Tr} G_j Q_{jk} G_k Q_{kj}, \tag{22}$$

and each term in the sum can be written $\operatorname{Tr}(\tilde{R}_j^\top Q_{jk} \tilde{R}_k)(\tilde{R}_k^\top Q_{kj} \tilde{R}_j^\top) = \|\tilde{R}_j^\top Q_{jk} \tilde{R}_k\|^2_{HS}$, which is computed in $O(s^2 r + sr^2)$ time, plus $O(bs^3)$ to compute the Cholesky factor. □

Denoting $\varphi_{\mathbf{z}_i} = (\varphi(\mathbf{z}_{ij}))_{1\le j\le s}$, note that

$$\varphi_{\mathbf{z}_i} G_i \varphi^*_{\mathbf{z}_i} = \varphi_{\mathbf{z}_i} T_i^{-1} R_i R_i^\top (\varphi_{\mathbf{z}_i} T_i^{-1})^* = E_i R_i R_i^\top E_i^*, \tag{23}$$

with $E_i = \varphi_{\mathbf{z}_i} T_i^{-1}$ an orthonormal basis of $\operatorname{Span}(\varphi_{\mathbf{z}_{ij}})_{1\le j\le s}$ as $E_i^* E_i = \mathbf{I}_s$. Thus, each model's coefficient is defined on an orthonormal basis, which makes the optimization easier.

Remark 3 (Relation to Term Sparsity in POP)**.** *The successful application of polynomial hierarchies to problems with thousands of variables rely on making the moment matrix M having a block structure [11, 12]. If the monomial basis has size m, the constraint $M \succeq 0$ is replaced with $M = (\operatorname{Diag} M_i)_{i\in[b]}$ and $M_i \succeq 0$. This enables to solve b SDP of size at most s instead of one of size m. Our model in proposition 2 follows a similar route for having a lower computational budget.*

A.2 Global optimization with splitting scheme

While GloptiNets can provide certificates for functions, it falls behind local solvers in terms of competitiveness. The challenge lies in the fact that finding a certificate is considerably more difficult than locating a local minimum, as it necessitates the uniform approximation of the entire function. However, we present a novel algorithmic framework that has the potential to enhance the competitiveness of GloptiNets with local solvers while simultaneously delivering certificates. Our approach involves partitioning the search domain into multiple regions and computing lower bounds for each partition. By discarding portions of the domain where the lower bound exceeds a certain threshold, the algorithm progressively simplifies the optimization problem and removes areas from consideration. Moreover, such an approach is naturally well suited to parallel computation.

The algorithm relies on a divide-and-conquer mechanism. First, we split the hypercube $(-1, 1)^d$ in N regions, where N is the number of core available. We compute an upper bound with a local solver. For each region, we run GloptiNets *in parallel*, computing a certificate at regular interval. As soon as the certificate is bigger than the upper bound, we stop the process: we know that the global minimum is not in the associated region. We can then reallocate the freed computing power by splitting the biggest current region, which yields an easier problem. We stop as soon as the region considered are small enough. This is summarized in alg. 2, where Ⓟ indicates the loop run in parallel.

Note that minimizing f on a hypercube of center μ and size σ amounts to minimizing $x \mapsto f((x - \mu)/\sigma)$ on $[-1, 1]^d$, which is another Chebychev polynomial whose coefficients can be evaluated efficiently thanks to the order-2 relation every orthonormal polynomial satisfy. For Chebychev polynomials, that is $H_{\omega+1}(x) = 2xH_\omega(x) - H_{\omega-1}(x)$.

Algorithm 2: Splitting scheme with GloptiNets

Data: A Chebychev polynomial f with a unique global optimum, a probability δ, a number of cores N and a volume $\rho < 1/N$.
Result: A certificate on f: $f_\star \geq C_\delta(f)$ with proba. $1 - \delta_\star$.
`/* Initialization: upper bound and partition */`
$\Pi = \mathsf{partition}([-1, 1]^d, N)$, $\delta_\star = 0.$;
Ⓟ $\mathsf{ub} = \min_{\pi\in\Pi} \{\mathsf{localsolver}_{x\in\pi} f(x)\}$;
`/* Iterate over the partition */`
Ⓟ **for** $\pi \in \Pi$, **While** $\mathsf{length}(\Pi) > 1$ **do**
    while $C_\delta(f_\pi) < \mathsf{ub}$ **do**
        Continue optimization;
    Split biggest part: $\pi_0 = \arg\max_{\pi\in\Pi} \mathrm{Vol}(\pi)$; $(\pi_1, \pi_2) = \mathsf{partition}(\pi_0, 2)$;
    If $\mathrm{Vol}(\pi_{1,2}) < \rho$: end this process ;
    Update upper bound: $\mathsf{ub} = \min\left\{\mathsf{ub}, \mathsf{localsolver}_{x\in\pi_{1,2}} f(x)\right\}$;
    Update search space and $\delta_\star$: $\Pi = \Pi \setminus \{\pi, \pi_0\} \cup \{\pi_1, \pi_2\}$, $\delta_\star = 1 - (1 - \delta_\star)(1 - \delta)$;
`/* A single region in Π remains */`
Returns $\Pi = \{\pi\}$, $C_\delta(f_\pi)$, $\delta_\star$;

A.3 Warm restarts

Our model distinguishes itself by leveraging the analytical properties of the objective function, rather than relying solely on algebraic characteristics. This approach offers a notable advantage, as closely related functions can naturally benefit from a warm restart. For example, if we already have a certificate for a function f using a PSD model g, and we seek to compute a certificate for a similar function $\tilde{f} \approx f$, we can readily employ GloptiNets by initializing the PSD model with g. In contrast, P-SoS methods, which rely on SDP programs, cannot directly adapt to new problems without significant effort. For instance, if a new component is introduced, an entirely new SDP must be solved. Our model's ability to accommodate related yet distinct problems could prove highly valuable in domains with a frequent need to certify different but closely related problems. In the industry, the Optimal Power Flow (OPF) problem requires periodic solves every 5 minutes [21]. With GloptiNets, once the initial challenging solve is performed, subsequent solves become easier assuming minimal changes in supply and demand conditions.

B Kernel defined on the Chebychev basis

In this section we describe the approach we take to model functions written in the Chebychev basis. For f such a polynomial, a naive approach would simply model $\tilde{f} = f \circ \cos(2\pi\cdot)$ as a trigonometric polynomial. However, note that the decomposition of $\tilde{f}$ only has cosine terms. Thus, approximating $f - f_\star$ efficiently requires a PSD model which has only cosine terms in its Fourier decomposition. This is achieved by using a kernel written in the Chebychev basis, as introduce in proposition 1, for which we now provide a proof.

Proof of proposition 1. Let $x, y \in [-1, 1]$ and $u, v \in [0, 1/2]$ s.t. $x, y = \cos(2\pi u), \cos(2\pi v)$, by bijectivity of the cosine function on $[0, \pi]$. From the definition of K in eq. (19) and the definition of q in eq. (5), we have that

$$
\begin{aligned}
K(x, y) &= \frac{1}{2} \sum_{\omega \in \mathbb{Z}} \widehat{q}_\omega \left(e^{2\pi i\omega(u+v)} + e^{2\pi i\omega(u-v)} \right) \\
&= \sum_{\omega \in \mathbb{Z}} \widehat{q}_\omega e^{2\pi i\omega u} \cos(2\pi\omega v) \\
&= \widehat{q}_0 + 2 \sum_{\omega \in \mathbb{N}} \widehat{q}_\omega \cos(2\pi\omega u) \cos(2\pi\omega v) \\
&= \widehat{q}_0 + 2 \sum_{\omega \in \mathbb{N}} \widehat{q}_\omega H_\omega(u) H_\omega(v).
\end{aligned}
$$

Since q has positive Fourier transform, this makes the feature map of K explicit with $K(x, y) = \varphi(u) \cdot \varphi(v)$, $\varphi(u)_\omega = \sqrt{(1 + \mathbf{1}_{\omega \neq 0})\widehat{q}_\omega} H_\omega(u)$, for $\omega \in \mathbb{N}$. Hence the kernel is a reproducing kernel. $\square$

We now use this kernel with the Bessel function $x \mapsto e^{s(\cos(2\pi x)-1)}$, *i.e.* we define the kernel K on $[-1, 1]$ to satisfy

$$
\forall u, v \in (0, 1/2), \quad K(\cos(2\pi u), \cos(2\pi v)) = \frac{1}{2} \left(e^{s(\cos(2\pi(u+v))} + e^{s(\cos(2\pi(u-v))} \right). \tag{24}
$$

As it was the case for the torus, this kernel enables an easy characterization of a RKHS in which an associated PSD model g lives.

Lemma 3 (Chebychev coefficient of the Bessel kernel)**.** *Let g be a PSD model as in definition 1, with the kernel K of eq. (24). Then, the Chebychev coefficient $\omega \in \mathbb{N}^d$ of g can be computed in $O(m^2 d)$ time with*

$$
\begin{aligned}
g_\omega = \sum_{i,j=1}^{m} A_{ij} \prod_{\ell=1}^{d} (1 + \mathbf{1}_{\omega \neq 0}) \frac{e^{-2\mathbf{s}_\ell}}{2} \Big[& \; I_{\omega_\ell}(2\mathbf{s}_\ell \sigma_{-\ell ij}) H_{\omega_\ell}(\sigma_{+\ell ij}) \\
& + I_{\omega_\ell}(2\mathbf{s}_\ell \sigma_{+\ell ij}) H_{\omega_\ell}(\sigma_{-\ell ij}) \Big]
\end{aligned} \tag{25}
$$

where

$$
\sigma_{\pm \ell ij} = \cos(2\pi m_{\pm \ell ij}), \quad m_{\pm \ell ij} = (\mathbf{u}_{\ell ij} \pm \mathbf{u}_{\ell ij})/2, \quad \textit{and} \quad \cos 2\pi \mathbf{u}_{\ell ij} = \mathbf{z}_{\ell ij}.
$$

Proof.

Expanding g and definition of Chebychev coefficient. From the definition of g in eq. (4), we have

$$
g(\mathbf{x}) = \sum_{i,j=1}^{m} A_{ij} \prod_{\ell=1}^{d} K_{\mathbf{s}_\ell}(\mathbf{x}_\ell, \mathbf{z}_{\ell i}) K_{\mathbf{s}_\ell}(\mathbf{x}_\ell, \mathbf{z}_{\ell j}). \tag{26}
$$

We consider $x, y, z \in (-1, 1)$ and $s > 0$. We denote $u, v, w \in (0, 1/2)$ s.t.

$$
x, y, z = \cos 2\pi u, \cos 2\pi v, \cos 2\pi w
$$

with the bijectivity of $x \mapsto \cos(2\pi x)$ on $(0, 1/2)$. We now compute the Chebychev coefficient of $x \mapsto K_s(x, y)K_s(x, z)$. Denoted h_ω, this is

$$\forall \omega \in \mathbb{N}, \;\; h_\omega = \frac{1 + \mathbf{1}_{\omega \neq 0}}{\pi} \int_{-1}^{1} K_s(x, y)K_s(x, z)T_\omega(x) \frac{\mathrm{d}x}{\sqrt{1 - x^2}},$$

or equivalently

$$\forall \omega \in \mathbb{N}, \;\; h_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \int_{0}^{1} K_s(\cos 2\pi u, \cos 2\pi v)K_s(\cos 2\pi u, \cos 2\pi w) \cos(2\pi \omega u)\mathrm{d}u. \quad (27)$$

Chebychev coefficient of kernel product. With the definition of the kernel in proposition 1, eq. (19), we have

$$\begin{aligned} K_s(x, y)K_s(x, z) &= \frac{1}{4} \left(h(u + v) + h(u - v)\right) \times \left(h(u + w) + h(u - w)\right) \\ &= \frac{e^{-2s}}{4} \left(e^{s \cos 2\pi(u+v)} + e^{s \cos 2\pi(u-v)}\right) \times \left(e^{s \cos 2\pi(u+w)} + e^{s \cos 2\pi(u-w)}\right) \end{aligned}$$

Now use the sum-to-product formula with the cosines to obtain

$$K_s(x, y)K_s(x, z) = \frac{e^{-2s}}{4} \Big(e^{2s \cos 2\pi(\frac{v-w}{2}) \cos 2\pi(u + \frac{v+w}{2})} + e^{2s \cos 2\pi(\frac{v-w}{2}) \cos 2\pi(u - \frac{v+w}{2})} + e^{2s \cos 2\pi(\frac{v+w}{2}) \cos 2\pi(u + \frac{v-w}{2})} + e^{2s \cos 2\pi(\frac{v+w}{2}) \cos 2\pi(u - \frac{v-w}{2})} \Big), \quad (28)$$

We simplify this expression by introducing

$$m_\pm = \frac{1}{2}(v \pm w) \;\text{ and }\; \sigma_\pm = \cos 2\pi m_\pm. \quad (29)$$

Then, eq. (28) becomes

$$K_s(x, y)K_s(x, z) = \frac{e^{-2s}}{4} \Big(e^{2s\sigma_- \cos 2\pi(u+m_+)} + e^{2s\sigma_- \cos 2\pi(u-m_+)} + e^{2s\sigma_+ \cos 2\pi(u+m_-)} + e^{2s\sigma_+ \cos 2\pi(u-m_-)} \Big). \quad (30)$$

We recognize the definition of the kernel (which is not a surprise as we chose the kernel to be stable by product). However, we need variables in $(0, 1/2)$ to retrieve the proper definition of the kernel. Instead, we use lemma 4 on eq. (30) combined with eq. (27), to obtain

$$h_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \frac{e^{-2s}}{4} \Big(\cos(2\pi\omega m_+) I_\omega(2s\sigma_-) + \cos(2\pi\omega m_+) I_\omega(2s\sigma_-) + \cos(2\pi\omega m_-) I_\omega(2s\sigma_+) + \cos(2\pi\omega m_-) I_\omega(2s\sigma_+) \Big),$$

which gives

$$h_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \frac{e^{-2s}}{2} (\cos(2\pi\omega m_+) I_\omega(2s\sigma_-) + \cos(2\pi\omega m_-) I_\omega(2s\sigma_+)). \quad (31)$$

Equation (31) contains the Chebychev coefficient of the product of two kernel function as defined in eq. (27). Plugging this result into the definition of g in eq. (26), and noting that $\cos(2\pi\omega m_\pm) = H_\omega(\cos 2\pi m_\pm) = H_\omega(\sigma_\pm)$, we obtain the result. □

Thanks to lemma 3, we see that a model g defined as in definition 1 with the Bessel kernel $K_\mathbf{s}$ of eq. (24) as its Chebychev coefficients decaying in $O(I_\omega(2s))$. Hence, it belongs to $\mathcal{H}_{2\mathbf{s}}$, the RKHS associated to $K_{2\mathbf{s}}$.

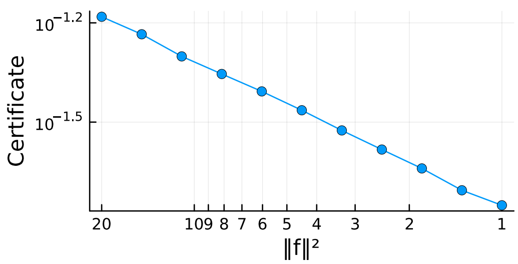

Figure 3: Certificate *vs.* RKHS norm of f, for a given model g with a fixed number of parameters. f has 1146 coefficients and g has 22528 parameters. Best certificate is kept among a set of optimization hyperparameters. As the norm of f decreases, fitting $f - f_\star$ with g is easier and the certificate becomes tighter.

C Additional details on the experiments

Regularization. Regularization is performed by approximating the HS norm with a proxy which is faster to compute. We use $\|R_j^\top \tilde{R}_k\|_{HS}^2$ instead of $\|\tilde{R}_j^\top Q_{jk} \tilde{R}_k\|_{HS}^2$ in eq. (22).

Hardware. GloptiNets was used with NVIDIA V100 GPUs for the interpolation part, and Intel Xeon CPU E5-2698 v4 @ 2.20GHz for computing the certificate. TSSOS was ran on a Apple M1 chip with Mosek solver.

Configuration of TSSOS. We use the lowest possible relaxation order d (*i.e.* $\lceil \deg f/2 \rceil$), along with Chordal sparsity. We use the first relaxation step of the hierarchy. In these settings, TSSOS is not guaranteed to converge to $f_\star$ but will executes the fastest.

Certificate vs. number of parameter for a given function. In fig. 2, the target function is a random polynomial of norm 1. The models forming the blue line are defined as in proposition 2, with rank, block size and number of blocks equal to $(1, bs, 1)$ respectively, with bs the block size we vary. The number of frequencies sampled to compute the certificate is 160000, and accounts for the fact that the bound on the variance becomes larger than the MOM estimator for large models.

Certificate vs. problem difficulty for a given model. We have 3 related parameters: the quality of the optimization (given by the certificate), the expressivity of the model (given by its number of parameters), and the difficulty of the optimization (given by the norm of the function). In fig. 2, we fix the latter and plot the relation between the first two. Here, we fix the model with parameters $(8, 16, 128)$, and we optimize a polynomial in $3d$ of degree 12, with RKHS norm ranging from 1 to 20. The certificates obtained are given in fig. 3. The resulting plot exhibits a clear polynomial relation between the certificate and the norm of the function, with a slope of -0.88. This suggest that the certificate behaves as $O(\|f\|_{\mathcal{H}_{2s}}^{-1})$.

Comparison with TSSOS on the Fourier basis. In table 1, the polynomials f all have a RKHS norm of 1. The small model is defined as in proposition 2, with rank, block size and number of blocks equal to $4, 8, 16$ respectively. For the big models, those values are $8, 16, 32$. The certificate is the maximum of the Chebychev bound of theorem 2 and the MoM bound of theorem 3. The number of frequencies sampled is 160000.

Comparison with TSSOS on the Chebychev basis. We compare GloptiNets with TSSOS on random Chebychev polynomials in table 2, similarly to the comparison with trigonometric polynomials in table 1. Minimizing polynomials defined on the canonical basis is easier: contrary to trigonometric polynomials, there is no need to account for the imaginary part of the variable. If d is the dimension, complex polynomials are encoded in a variable of dimension $2d$ in TSSOS, following the definition of Hermitian Sum-of-Squares iontroduced in [30]. Hence, the random polynomials we consider are characterized by the dimension d and their number of coefficients n; instead of bounding the degree, we use all the basis elements $H_\omega(\mathbf{x}) = \prod_{\ell=1}^d H_{\omega_\ell}(\mathbf{x}_\ell)$ for which $\|\omega\|_\infty \leq p$. The maximum degree

Table 2: GloptiNets and TSSOS on random Chebychev polynomials. The same conclusion as in table 1 applies. While TSSOS is very efficient on small problems, its memory requirements grow exponentially with the problem size. GloptiNets has less accuracy, but a computational burden which does not increase with the problem size.

d	p	n	TSSOS		GN-small		GN-big	
			Certif.	t	Certif.	t	Certif.	t
	3	255	$3.4 \cdot 10^{-7}$	6	$9.3 \cdot 10^{-2}$	54	$3.3 \cdot 10^{-2}$	264
4	4	624	$2.1 \cdot 10^{-9}$	153	$8.3 \cdot 10^{-2}$	55	$2.8 \cdot 10^{-2}$	258
	5	1295	Out of memory!	-	$1.0 \cdot 10^{-1}$	56	$3.2 \cdot 10^{-2}$	264

is then dp. The RKHS norm of f is fixed to 1. As with the comparison on Trigonometric polynomial table 1, we see that GloptiNets provides similar certificates no matter the number of coefficients in f. Even though it lags behind TSSOS for small polynomials, it handles large polynomials which are intractable to TSSOS. The "small" and "big" models have the same structure as for the trigonometric polynomials experiments.

Sampling from the Bessel distribution. The function $\omega \mapsto e^{-s} I_\omega(s)$ decays rapidly. In fact, with $s = 2$, which is the value used to generate the random polynomials, it falls under machine precision as soon as $\omega > 14$. Thus, we approximate the distribution with a discrete one with weights $I_\omega(s)$ for ω s.t. the result is above the machine precision. We then extend it to multiple dimension with a tensor product. Finally, we use a hash table to store the already sampled frequency, to make the evaluation of million of frequencies much faster. For instance in dimension 5, sampling 10^6 frequencies from the Bessel distribution of parameter $s = 2$ on $\mathbb{N}^5$ yields only $\approx 10^4$ unique frequencies. This allows for tighter certificates, as it makes the r.h.s of eq. (9), in $1/N$, much smaller. Note that the time to generate this hash table is *not* reported in tables 1 and 2, and of the order of a few seconds.

D Other computation

Lemma 4. *Let f be the function defined on $(-1,1)$ with*

$$\forall u \in (0, 1/2), \quad f(\cos 2\pi u) = e^{s \cos 2\pi(u-v)}. \tag{32}$$

Then, its Chebychev coefficient are given with

$$f_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \cos(2\pi\omega v) I_\omega(s). \tag{33}$$

Proof. The $\omega \in \mathbb{N}_*$. The component ω of a function f on the Chebychev basis is given with

$$f_\omega = \frac{2}{\pi} \int_{-1}^{1} f(x) T_\omega(x) \frac{\mathrm{d}x}{\sqrt{1-x^2}},$$

which we conveniently rewrite, with the classical change of variable $x = \cos 2\pi u$,

$$f_\omega = 2 \int_{I_1} f(\cos 2\pi u) \cos(2\pi\omega u) \mathrm{d}u \tag{34}$$

which is valid for any interval $I_1 \subset \mathbb{R}$ of length 1.

Now, for $s > 0$, consider the function f defined on $(-1, 1)$ with $x \mapsto e^{s\cos(\arccos(x) - 2\pi v)}$, or equivalently

$$\forall u \in (0, 1/2), \quad f(\cos 2\pi u) = e^{s \cos 2\pi(u-v)}. \tag{35}$$

Putting eq. (35) into eq. (34), we obtain

$$\begin{aligned} f_\omega &= 2 \int_{I_1} e^{s \cos 2\pi(u-v)} \cos(2\pi\omega u) \mathrm{d}u \\ &= 2 \int_{I_1} e^{s \cos 2\pi u} \cos(2\pi\omega(u+v)) \mathrm{d}u \\ &= 2 \int_{I_1} e^{s \cos 2\pi u} \cos(2\pi\omega u)\cos(2\pi\omega v) \mathrm{d}u - 2 \int_{I_1} e^{s \cos 2\pi u} \sin(2\pi\omega u)\sin(2\pi\omega v) \mathrm{d}u. \end{aligned}$$

The last term is odd, hence integrate to 0 on an interval centered around 0. Hence,

$$f_\omega = 2\cos(2\pi\omega v)\int_{I_1} e^{s\cos 2\pi u}\cos(2\pi\omega u)\mathrm{d}u. \tag{36}$$

We recognize the definition of the modified Bessel function of the first kind, defined in eq. (14). Plugging this into eq. (36), we obtain

$$f_\omega = 2\cos(2\pi\omega v)I_\omega(s) = 2I_\omega(s)H_\omega(\cos(2\pi v)). \tag{37}$$

If $\omega = 0$, we add a factor $1/2$ into the definition in eq. (34), which yields

$$f_\omega = I_0(s). \tag{38}$$

□

References

[1] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826(1989):37, 1989.

[2] Reiner Horst and Panos M Pardalos. *Handbook of global optimization*, volume 2. Springer Science & Business Media, 2013.

[3] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] Peter JM Van Laarhoven, Emile HL Aarts, Peter JM van Laarhoven, and Emile HL Aarts. *Simulated annealing*. Springer, 1987.

[5] Jean Bernard Lasserre. *Moments, Positive Polynomials and Their Applications*, volume 1 of *Series on Optimization and Its Applications*. IMPERIAL COLLEGE PRESS, October 2009.

[6] Jean B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, January 2001.

[7] Blake Woodworth, Francis Bach, and Alessandro Rudi. Non-Convex Optimization with Certificates and Fast Rates Through Kernel Sums of Squares. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 4620–4642. PMLR, June 2022.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[9] Didier Henrion, Milan Korda, and Jean-Bernard Lasserre. *The Moment-SOS Hierarchy*, volume 4 of *Optimization and Its Applications*. World Scientific Publishing Europe Ltd., December 2020.

[10] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of Squares and Semidefinite Program Relaxations for Polynomial Optimization Problems with Structured Sparsity. *SIAM Journal on Optimization*, 17(1):218–242, January 2006.

[11] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity. *SIAM Journal on Optimization*, 31(1):30–58, January 2021.

[12] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Chordal-TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity with Chordal Extension. *SIAM Journal on Optimization*, 31(1):114–141, January 2021.

[13] Francis Bach and Alessandro Rudi. Exponential convergence of sum-of-squares hierarchies for trigonometric polynomials, January 2023.

[14] Monique Laurent and Lucas Slot. An effective version of schmüdgen's positivstellensatz for the hypercube. *Optimization Letters*, September 2022. Funding Information: This work is supported by the European Union's Framework Programme for Research and Innovation Horizon 2020 under the Marie Skłodowska-Curie Actions Grant Agreement No. 764759 (MINOA). Publisher Copyright: © 2022, The Author(s).

[15] Ngoc Hoang Anh Mai, J. B. Lasserre, Victor Magron, and Jie Wang. Exploiting Constant Trace Property in Large-scale Polynomial Optimization. *ACM Transactions on Mathematical Software*, 48(4):40:1–40:39, December 2022.

[16] Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Non-parametric Models for Non-negative Functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12816–12826. Curran Associates, Inc., 2020.

[17] Eloïse Berthier, Justin Carpentier, Alessandro Rudi, and Francis Bach. Infinite-Dimensional Sums-of-Squares for Optimal Control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 577–582, December 2022.

[18] Boris Muzellec, Adrien Vacher, Francis Bach, François-Xavier Vialard, and Alessandro Rudi. Near-optimal estimation of smooth transport maps with kernel sums-of-squares, December 2021.

[19] Alessandro Rudi and Carlo Ciliberto. PSD Representations for Effective Probability Models. In *Advances in Neural Information Processing Systems*, volume 34, pages 19411–19422. Curran Associates, Inc., 2021.

[20] Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding Global Minima via Kernel Approximations. *arXiv:2012.11978 [cs, math, stat]*, December 2020.

[21] Pascal Van Hentenryck. Machine Learning for Optimal Power Flows. *INFORMS Tutorials in Operations Research*, October 18.

[22] Joseph J. Hilling and Anthony Sudbery. The geometric measure of multipartite entanglement and the singular values of a hypermatrix. *Journal of Mathematical Physics*, 51(7):072102, July 2010.

[23] Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. Phase Recovery, MaxCut and Complex Semidefinite Programming, July 2013.

[24] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[25] The Basic Theorems of Fourier Analysis. In *Fourier Analysis on Groups*, chapter 1, pages 1–34. John Wiley & Sons, Ltd, 1990.

[26] Luc Devroye, Matthieu Lerasle, Gabor Lugosi, and Roberto I. Oliveira. Sub-Gaussian Mean Estimators. *The Annals of Statistics*, 44(6):2695–2725, 2016.

[27] G. N. Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press, 1922.

[28] Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 2016.

[29] Jie Wang and Victor Magron. Exploiting Sparsity in Complex Polynomial Optimization. *Journal of Optimization Theory and Applications*, 192(1):335–359, January 2022.

[30] Cédric Josz and Daniel K. Molzahn. Lasserre Hierarchy for Large Scale Polynomial Optimization in Real and Complex Variables. *SIAM Journal on Optimization*, 28(2):1017–1048, January 2018.