

Basc: Constrained Approximation by Semidefinite Programming

Simon Foucart* and Vladlena Powers — Texas A&M University and Columbia University†

Abstract

This article details the theoretical grounds for a semidefinite-programming-based method that computes best approximants by splines under some general constraints and relative to several function norms, notably the max-norm. The method has been implemented as a MATLAB package called Basc, which relies on the two external packages Chebfun and CVX.

Key words: constrained approximation, simultaneous approximation, semidefinite programming, Riesz–Fejér theorem, nonnegative polynomials.

AMS classification: 41A10, 41A15, 41A29, 41A50, 65D15, 90C22.

1 Introduction

In this article, we embark on the task of computing a best approximant to a target function F by polynomials or splines subject to some convex constraints. In other words, with \mathcal{S} representing the linear space of approximants, we aim at solving the optimization problem

$$(1) \quad \underset{P}{\text{minimize}} \quad \|F - P\| \quad \text{subject to} \quad P \in \mathcal{S}$$

and to some prescribed constraints.

There is a well-established literature on the subject, see e.g. the survey [8] reporting mostly on existence, uniqueness, and characterization results for approximations with one-sided constraints, positivity constraints, and interpolatory constraints. The article [5] contains more comprehensive characterization results. Although both [8] and [5] mention computational procedures relative to the max-norm, the algorithms are ad hoc or not available: [8] points out the Remez-type algorithm

*S. F. partially supported by NSF grant number DMS-1120622

†This work was performed when both authors were at the University of Georgia

from [11] for approximation with restricted range and it describes a related linear program for convex approximation, while [5] alludes to a general Remez-type algorithm but the supporting manuscript seems not to have been published (see the reference numbered 9 in [5]). In a similar way, the even more general Remez-type algorithm of [2] has not been implemented (to the best of our knowledge). We propose to fill this gap by developing a novel and unified framework for constrained approximation based on semidefinite programming. The theoretical grounds presented here are accompanied by a MATLAB package called **Basc** and downloadable from the first author's webpage. The acronym **Basc** stands for **B**est **A**pproximations by **S**plines under **C**onstraints. **Basc** heavily relies on two external MATLAB packages: **Chebfun** [12] for the convenience to deal with Chebyshev expansions and **CVX** [3] for the convenience to specify and solve convex programs. **Basc** handles a number of instances of the main problem (1), precisely:

- The space \mathcal{S} can consist of polynomials of degree at most n or of splines (i.e., piecewise polynomials) of degree at most n obeying various smoothness conditions (the breakpoints being fixed);
- By default, the norm $\|\cdot\|$ is the max-norm on $[-1, 1]$, but it can also be the weighted max-norm

$$\|G\|_W := \max_{x \in [-1, 1]} \frac{|G(x)|}{W(x)} \quad \text{for some weight function } W \geq 0,$$

or a q -norm

$$\|G\|_q := \left[\int_{-1}^1 |G(x)|^q dx \right]^{1/q} \quad \text{for some even integer } q \geq 2.$$

In fact, $\|\cdot\|$ can even be a quasinorm of the form

$$\|G\| = \left[\sum_{\kappa=0}^{\infty} \left\{ \gamma_{\kappa} \|G^{(\kappa)}\| \right\}^r \right]^{1/r}$$

where $\|\cdot\|$ is one of the norms mentioned above, so that special choices of $\gamma_{\kappa} \geq 0$ and $r \geq 1$ incorporates, for instance, simultaneous approximation for which $\|G\| = \max_h \|G^{(\kappa_h)}\|_{\infty}$;

- The constraints can impose some interpolatory conditions, i.e., the approximant must satisfy

$$P^{(\kappa_h)}(x_i^{\{h\}}) = y_i^{\{h\}} \quad \text{for some prescribed } \kappa_h \text{ and } (x_1^{\{h\}}, y_1^{\{h\}}), \dots, (x_{\nu^{\{h\}}}^{\{h\}}, y_{\nu^{\{h\}}}^{\{h\}});$$

- The constraints can force the approximant to have a certain parity — it could be required to be an even or an odd function;
- The constraints can restrict the ranges of the approximant, i.e., its derivatives must satisfy

$$L^{\{h\}} \leq P^{(\kappa_h)} \quad \text{and/or} \quad P^{(\kappa_h)} \leq U^{\{h\}} \quad \text{on } [-1, 1]$$

for some prescribed indices κ_h and prescribed lower and upper functions¹ $L^{\{\kappa_h\}}$ and $U^{\{\kappa_h\}}$.

This includes one-sided approximation if $L^{\{0\}}$ or $U^{\{0\}}$ is chosen to be the target function F ;

¹it is not assumed that $F^{(\kappa_h)}$ lies between $L^{\{h\}}$ and $U^{\{h\}}$, so a best approximant by polynomial need not be unique — think of $F(x) = |x|$ to be approximated by quadratics with $U^{\{0\}} = 0$ and $L^{\{0\}} = -2$.

- The constraints can determine the shape of the approximant, i.e., they concern the sign pattern of the derivatives of P . This includes monotone approximation (when $P' \geq 0$ or $P' \leq 0$), convex approximation (when $P'' \geq 0$), and coconvex approximation (when P'' has the same sign as F'');

and any combination of the above scenarios.

There is a restriction on the target function F , though, namely that it is itself a (piecewise) polynomial. However, this is not a severe restriction in practice. Indeed, `Chebfun` works on the underlying assumption that a usual function F is well represented by its polynomial interpolant at Chebyshev points, hence `Chefun` manipulates a polynomial \bar{F} that agrees with F up to machine precision ϵ . Therefore, if P and \bar{P} denote the best max-norm approximants to F and \bar{F} , respectively, then the computed minimum $\|\bar{F} - \bar{P}\|_\infty$ is at most ϵ away from the sought minimum $\|F - P\|_\infty$, since

$$\begin{aligned} \|\bar{F} - \bar{P}\|_\infty &\leq \|\bar{F} - P\|_\infty \leq \|F - P\|_\infty + \|F - \bar{F}\|_\infty \leq \|F - P\|_\infty + \epsilon, \\ \|F - P\|_\infty &\leq \|F - \bar{P}\|_\infty \leq \|\bar{F} - \bar{P}\|_\infty + \|F - \bar{F}\|_\infty \leq \|\bar{F} - \bar{P}\|_\infty + \epsilon. \end{aligned}$$

These inequalities also imply that P and \bar{P} are at most ϵ/γ away in the max-norm for some constant $\gamma > 0$ a priori dependent on F , according to some results on strong uniqueness surveyed in [7].

The remainder of this article is organized as follows: we start in Section 2 by introducing the guiding principle in the simpler case of unconstrained approximation by trigonometric polynomials; in Section 3, we transfer the general strategy to the case of algebraic polynomials and piecewise algebraic polynomials and we show in details how to deal with different norms in the objective function and with several possible constraints; and finally Section 4 isolates a handful of examples to illustrate how the package `Basc` is used.

Before engaging into the heart of the matter, here are a few words on the notation chosen for this article. Capital letters such as F or S denote functions and the addition of a subscript as in $F_{m,\mu}$ denotes the restriction of F to some interval indexed by m and μ . Traditionally, $F^{(\kappa)}$ refers to the κ -th derivative of F , while $S^{\{h\}}$ refers to the element of a family of functions indexed by h , and likewise for $S^{\{h,m\}}$, $S^{\{h,m,\mu\}}$, \dots . The same convention applies to families of vectors or of matrices. Matrices are represented by bold uppercase letters and vectors by bold lowercase letters. Their entries are indicated by a subscript after appending a square bracket or by a subscript after dropping the bold face, such as $Q_{i,j}$ or $[\mathbf{p}^{\{m\}}]_k = p_k^{\{m\}}$. The fact that a matrix \mathbf{Q} is positive semidefinite is denoted by $\mathbf{Q} \succcurlyeq \mathbf{0}$. The entrywise product of two vectors \mathbf{u} and \mathbf{v} is written $\mathbf{u} \odot \mathbf{v}$, so that $[\mathbf{u} \odot \mathbf{v}]_i = u_i \times v_i$. Beware that the indexing of vectors and matrices may start at zero or at one, depending on the context. Thus, we may have $i \in \llbracket 0 : n \rrbracket$ or $i \in \llbracket 1 : n \rrbracket$ in the line above. The notation $\llbracket a : b \rrbracket$ stands for the set $\{a, a + 1, \dots, b - 1, b\}$ of integers at least equal to the integer a and at most equal to the integer b .

2 Prelude: Trigonometric Approximation without Constraints

Although the focus of `Basc` is on algebraic polynomials rather than trigonometric polynomials, this section serves as a simplified illustration of the optimization strategy used in the constrained approximation problem — keep in mind, however, that it is better to apply Remez algorithm for the unconstrained approximation problem discussed in this section. All the theoretical considerations presented next are consequences of Riesz–Fejér theorem and its generalizations. We briefly recall that by an algebraic polynomial (or simply polynomial) of degree at most n , we mean a function of the form $P(x) = \sum_{k=0}^n p_k x^k$. By a trigonometric polynomial of degree at most n , we mean a function of one of the equivalent forms:

- $C(\theta) = \sum_{k=-n}^n c_k e^{-ik\theta}$ with $c_{-k} = \overline{c_k}$ for all $k \in \llbracket 0 : n \rrbracket$;
- $C(\theta) = \frac{a_0}{2} + \sum_{k=1}^n [a_k \cos(k\theta) + b_k \sin(k\theta)]$ with $a_k, b_k \in \mathbb{R}$ for all $k \in \llbracket 0 : n \rrbracket$.

One passes from one representation to the other (the equality $b_0 = 0$ is implicit) using the relations

$$(2) \quad a_k = 2 \operatorname{Re}(c_k), \quad b_k = 2 \operatorname{Im}(c_k), \quad c_k = \frac{a_k + ib_k}{2}, \quad k \in \llbracket 0 : n \rrbracket.$$

Riesz–Fejér theorem states that a trigonometric polynomial C of degree at most n is nonnegative if and only if it can be written as $C(\theta) = |Q(e^{-i\theta})|^2$ for some algebraic polynomial Q of degree at most n . This results extends to nonnegativity on subintervals of $[-\pi, \pi]$ as follows (see Section 3.1 of [9] and the references cited there or the book [4] for more statements in this direction).

Theorem 1. Let C be a trigonometric polynomial of degree at most n . Then

$$C(\theta) \geq 0 \quad \text{for all } \theta \in [\alpha - \beta, \alpha + \beta]$$

if and only if

$$C(\theta) = |Q(e^{-i\theta})|^2 + [\cos(\theta - \alpha) - \cos(\beta)] |R(e^{-i\theta})|^2$$

for some algebraic polynomials Q and R of degree at most n and $n - 1$, respectively.

From our perspective, the nonnegativity on a subinterval is better characterized by the semidefinite condition below. The argument is classical, but we include a sketch of the proof for completeness.

Corollary 2. Let $C(\theta) = \sum_{k=-n}^n c_k e^{-ik\theta}$, $c_{-k} = \overline{c_k}$, be a trigonometric polynomial of degree at most n . Then

$$C(\theta) \geq 0 \quad \text{for all } \theta \in [\alpha - \beta, \alpha + \beta]$$

if and only if there exist matrices $\mathbf{Q} \in \mathbb{C}^{(n+1) \times (n+1)}$ and $\mathbf{R} \in \mathbb{C}^{n \times n}$ such that $\mathbf{Q} \succcurlyeq \mathbf{0}$, $\mathbf{R} \succcurlyeq \mathbf{0}$, and, for any $k \in \llbracket 0 : n \rrbracket$,

$$c_k = \sum_{i-j=k} Q_{i,j} + \frac{e^{1\alpha}}{2} \left[\sum_{i-j=k-1} R_{i,j} \right] - \cos(\beta) \left[\sum_{i-j=k} R_{i,j} \right] + \frac{e^{-1\alpha}}{2} \left[\sum_{i-j=k+1} R_{i,j} \right].$$

Remark. For the whole interval $[-\pi, \pi]$, one simply discard the matrix \mathbf{R} .

Proof. \Rightarrow Use Theorem 1 to write

$$C(\theta) = \left[\sum_{i=0}^n q_i e^{-i\theta} \right] \left[\sum_{j=0}^n \bar{q}_j e^{j\theta} \right] + \left[\frac{e^{1\alpha} e^{-i\theta}}{2} - \cos(\beta) + \frac{e^{-1\alpha} e^{i\theta}}{2} \right] \left[\sum_{i=0}^{n-1} r_i e^{-i\theta} \right] \left[\sum_{j=0}^{n-1} \bar{r}_j e^{j\theta} \right].$$

Identifying the coefficient of $e^{-ik\theta}$ gives, for any $k \in \llbracket -n : n \rrbracket$,

$$c_k = \left[\sum_{i-j=k} q_i \bar{q}_j \right] + \frac{e^{1\alpha}}{2} \left[\sum_{i-j=k-1} r_i \bar{r}_j \right] - \cos(\beta) \left[\sum_{i-j=k} r_i \bar{r}_j \right] + \frac{e^{-1\alpha}}{2} \left[\sum_{i-j=k+1} r_i \bar{r}_j \right].$$

This yields the desired result with $\mathbf{Q} := \mathbf{q}\mathbf{q}^*$ and $\mathbf{R} := \mathbf{r}\mathbf{r}^*$ (note that the conditions on c_k for $k \in \llbracket -n : 0 \rrbracket$ are redundant with the conditions on c_k for $k \in \llbracket 0 : n \rrbracket$).

\Leftarrow Setting $\mathbf{u}(\theta) := [1, e^{-i\theta}, \dots, e^{-in\theta}]^\top$ and $\mathbf{v}(t) := [1, e^{-i\theta}, \dots, e^{-i(n-1)\theta}]^\top$, the expression for c_k implies that

$$C(\theta) = \langle \mathbf{Q}\mathbf{u}(\theta), \mathbf{u}(\theta) \rangle + [\cos(\theta - \alpha) - \cos(\beta)] \langle \mathbf{R}\mathbf{v}(\theta), \mathbf{v}(\theta) \rangle.$$

Since all three terms are nonnegative on $[\alpha - \beta, \alpha + \beta]$, then so is $C(\theta)$. \square

With Corollary 2 in place, we can now introduce our semidefinite minimization strategy. Let $F(\theta) = \sum_{k=-N}^N f_k e^{-ik\theta}$ be a trigonometric polynomial of degree N that we wish to approximate by trigonometric polynomials C of degree at most n , $n < N$. The first step is to recast the program

$$\underset{C}{\text{minimize}} \|F - C\|_\infty$$

by introducing a slack variable d to arrive at the equivalent program

$$\underset{d, C}{\text{minimize}} d \quad \text{subject to} \quad \|F - C\|_\infty \leq d.$$

The constraint $\|F - C\|_\infty \leq d$ is equivalent to $-d \leq F(\theta) - C(\theta) \leq d$ for all $\theta \in [-\pi, \pi]$, i.e., to the two nonnegativity constraints $d + F(\theta) - C(\theta) \geq 0$ for all $\theta \in [-\pi, \pi]$ and $d - F(\theta) + C(\theta) \geq 0$ for all $\theta \in [-\pi, \pi]$. According to Corollary 2 (with \mathbf{R} being discarded), these constraints can be

expressed as semidefinite conditions. So we end up with the semidefinite program

$$\begin{aligned}
 & \underset{\substack{d \in \mathbb{R}, \mathbf{c} \in \mathbb{R}^{N+1} \\ \mathbf{Q}^-, \mathbf{Q}^+ \in \mathbb{C}^{(N+1) \times (N+1)}}}{\text{minimize}} & d & \text{subject to} & \mathbf{Q}^- \succcurlyeq \mathbf{0}, \mathbf{Q}^+ \succcurlyeq \mathbf{0}, \\
 & & & \text{to} & \left\{ \begin{array}{l} f_k - c_k, \quad k \in \llbracket 1 : N \rrbracket, \\ d + f_0 - c_0, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} Q_{i,j}^-, \\
 & & & \text{and to} & \left\{ \begin{array}{l} -f_k + c_k, \quad k \in \llbracket 1 : N \rrbracket, \\ d - f_0 + c_0, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} Q_{i,j}^+,
 \end{aligned}$$

with the implicit understanding that $c_k := 0$ for $n < k \leq N$. This optimization program can be passed directly in the above form to the `sdp` utility of `CVX`. `Basc` relies on `CVX` in the same way to solve semidefinite programs.

3 Center Piece: Approximation by Splines under Constraints

This section describes in details all the steps involved in the computation of best constrained approximations by splines obeying some prescribed smoothness conditions. We first transpose the considerations from Section 2 to the algebraic case, before focusing on handling the norm underlying the approximation, and finishing with the treatment of the possible constraints.

3.1 The main ingredient

Our approach relies on a reduction to the trigonometric case once the target function has been expanded on the basis of Chebyshev polynomials of the first kind. Precisely, if

$$(3) \quad A(x) = \sum_{k=0}^N a_k T_k(x)$$

is an algebraic polynomial of degree at most N expanded on the Chebyshev basis, then substituting $x = \cos(\theta)$ yields a trigonometric polynomial of degree at most N defined by

$$\widehat{A}(\theta) := A(\cos(\theta)) = \sum_{k=0}^N a_k \cos(k\theta).$$

The nonnegativity of A on a subinterval of $[-1, 1]$ translates into the nonnegativity of \widehat{A} on a subinterval of $[-\pi, \pi]$ (in fact, of $[0, \pi]$). Thus, according to Corollary 2 and the relations (2), we readily deduce the following theorem.²

² Different semidefinite characterizations can be deduced from other representations of nonnegative polynomials, e.g. [10, Theorem 1.21.1]. We chose this characterization based on Chebyshev expansion to take advantage of `Chebfun`.

Theorem 3. Let $A(x) = \sum_{k=0}^N a_k T_k(x)$ be a polynomial of degree at most N . Then

$$A(x) \geq 0 \quad \text{for all } x \in [u, v]$$

if and only if there exist matrices $\mathbf{Q} \in \mathbb{C}^{(N+1) \times (N+1)}$ and $\mathbf{R} \in \mathbb{C}^{N \times N}$ such that $\mathbf{Q} \succcurlyeq \mathbf{0}$, $\mathbf{R} \succcurlyeq \mathbf{0}$, and, for any $k \in \llbracket 0 : N \rrbracket$,

$$\begin{aligned} \left\{ \begin{array}{l} \frac{a_k}{2}, \quad k \in \llbracket 1 : N \rrbracket, \\ a_0, \quad k = 0. \end{array} \right\} &= \sum_{i-j=k} Q_{i,j} \\ &+ \frac{e^{i\alpha}}{2} \left[\sum_{i-j=k-1} R_{i,j} \right] - \cos(\beta) \left[\sum_{i-j=k} R_{i,j} \right] + \frac{e^{-i\alpha}}{2} \left[\sum_{i-j=k+1} R_{i,j} \right]. \end{aligned}$$

Here, $\alpha := (\arccos(u) + \arccos(v))/2$ and $\beta := (\arccos(u) - \arccos(v))/2$.

Remark. For the whole interval $[-1, 1]$, one simply discard the matrix \mathbf{R} .

We emphasize that the task of producing the expansion (3) is performed very efficiently by `Chebfun`, which makes computing with functions possible thanks precisely to such faithful expansions for usual functions. `Basc` also relies on `Chebfun` in several other instances pointed out later.

3.2 Dealing with the objective function

From now on, we assume that the target function F is a polynomial or a piecewise polynomial³. As emphasized earlier, this restriction is not severe and `Chebfun` does indeed represent functions in this way. The arguments below show that in this case best constrained spline approximants can be computed exactly as solutions of semidefinite programs. In its generic form, the optimization problem at hand reads

$$(4) \quad \underset{P \in \mathcal{S}}{\text{minimize}} \left[\sum_{h=1}^H \left\{ \gamma_h \|(F - P)^{(\kappa_h)}\| \right\}^r \right]^{1/r} \quad \text{subject to some prescribed constraints,}$$

where $H = H^{\text{obj}}$, $\gamma = \gamma^{\text{obj}}$, and $\kappa = \kappa^{\text{obj}}$ are parameters specific to the given objective function. The set \mathcal{S} is a fixed space of splines with knots $-1 = t_0 < t_1 < \dots < t_{M-1} < t_M = 1$, with degree n_m on each subinterval $I_m := [t_{m-1}, t_m]$, $m \in \llbracket 1 : M \rrbracket$, and with smoothness \mathcal{C}^{s_m} at each breakpoint t_m , $m \in \llbracket 1 : M - 1 \rrbracket$. In the same spirit as Section 2, we introduce slack variables d_h , $h \in \llbracket 1 : H \rrbracket$, and we look at the equivalent optimization problem

$$\begin{aligned} \underset{\mathbf{d} \in \mathbb{R}^H, P \in \mathcal{S}}{\text{minimize}} \quad & \|\gamma \odot \mathbf{d}\|_r & \text{subject to} & \quad \|(F - P)^{(\kappa_h)}\| \leq d_h, \quad h \in \llbracket 1 : H \rrbracket, \\ & & \text{and to} & \quad \text{some originally prescribed constraints.} \end{aligned}$$

³similar arguments would also hold if F was a piecewise rational function.

We now distinguish several cases depending on the norm $\|\cdot\|$ under consideration.

Max-norm: The condition $\|(F - P)^{(\kappa_h)}\|_\infty \leq d_h$ transforms into the condition

$$(5) \quad -d_h \leq F^{(\kappa_h)}(x) - P^{(\kappa_h)}(x) \leq d_h \quad \text{for all } x \in [-1, 1],$$

and in turn into the M conditions given for each $m \in \llbracket 1 : M \rrbracket$ by

$$(6) \quad -d_h \leq F_m^{(\kappa_h)}(t) - P_m^{(\kappa_h)}(t) \leq d_h \quad \text{for all } t \in I_m,$$

where F_m and P_m denote the restrictions to I_m of F and P , respectively. We transpose F_m and P_m from the subinterval I_m to the interval $[-1, 1]$ by setting, for any $x \in [-1, 1]$,

$$(7) \quad \tilde{P}_m(x) := P_m \left(\frac{t_m + t_{m-1}}{2} + x \frac{t_m - t_{m-1}}{2} \right), \quad \tilde{F}_m(x) := F_m \left(\frac{t_m + t_{m-1}}{2} + x \frac{t_m - t_{m-1}}{2} \right).$$

Then, with $\lambda_m := (t_m - t_{m-1})/2$ denoting the half-length of the subinterval I_m , the conditions (6) become

$$(8) \quad -\lambda_m^{\kappa_h} d_h \leq \tilde{F}_m^{(\kappa_h)}(x) - \tilde{P}_m^{(\kappa_h)}(x) \leq \lambda_m^{\kappa_h} d_h \quad \text{for all } x \in [-1, 1].$$

We expand the polynomial \tilde{P}_m in the Chebyshev basis as

$$\tilde{P}_m = \sum_{k=0}^{n_m} p_k^{\{m\}} T_k.$$

We do the same for \tilde{F}_m , assuming first that it is a single polynomial — in general, it may consist of several polynomial pieces. Thus, we have

$$\tilde{F}_m = \sum_{k=0}^{n'_m} f_k^{\{m\}} T_k.$$

The difference $\tilde{F}_m^{(\kappa_h)} - \tilde{P}_m^{(\kappa_h)}$ has degree at most $N := \max_{\ell \in \llbracket 1 : M \rrbracket} \max\{n_\ell, n'_\ell\}$, hence according to Theorem 3 (with \mathbf{R} being discarded), the two nonnegativity conditions of (8) are equivalent to the existence of positive semidefinite matrices $\mathbf{Q}^{-, \{h, m\}}, \mathbf{Q}^{+, \{h, m\}} \in \mathbb{C}^{(N+1) \times (N+1)}$ such that

$$(9) \quad \left\{ \begin{array}{l} \frac{[\mathbf{D}^{\{\kappa_h\}}(\mathbf{f}^{\{m\}} - \mathbf{p}^{\{m\}})]_k}{2}, \quad k \in \llbracket 1, N \rrbracket, \\ \lambda_m^{\kappa_h} d_h + [\mathbf{D}^{\{\kappa_h\}}(\mathbf{f}^{\{m\}} - \mathbf{p}^{\{m\}})]_0, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} Q_{i,j}^{-, \{h, m\}},$$

$$\left\{ \begin{array}{l} \frac{[\mathbf{D}^{\{\kappa_h\}}(-\mathbf{f}^{\{m\}} + \mathbf{p}^{\{m\}})]_k}{2}, \quad k \in \llbracket 1, N \rrbracket, \\ \lambda_m^{\kappa_h} d_h + [\mathbf{D}^{\{\kappa_h\}}(-\mathbf{f}^{\{m\}} + \mathbf{p}^{\{m\}})]_0, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} Q_{i,j}^{+, \{h, m\}}.$$

Here it is implicitly understood that $p_k^{\{m\}} := 0$ for $n_m < k \leq N$ and $f_k^{\{m\}} := 0$ for $n'_m < k \leq N$ and $\mathbf{D}^{\{\kappa\}}$ represents the $(N+1) \times (N+1)$ matrix transforming the Chebyshev coefficients of

a polynomial into the Chebyshev coefficients of its κ th derivative — its construction is greatly facilitated by Chebfun.

Let us now assume that \tilde{F}_m consists of several polynomial pieces $\tilde{F}_{m,\mu}$, $\mu \in \llbracket 1 : \mu_m \rrbracket$, each associated with subintervals $[\tau_{m,\mu-1}, \tau_{m,\mu}]$ of $[-1, 1]$. Condition (8) splits into μ_m conditions given for each $\mu \in \llbracket 1 : \mu_m \rrbracket$ by

$$(10) \quad -\lambda_m^{\kappa_h} d_h \leq \tilde{F}_{m,\mu}^{(\kappa_h)}(x) - \tilde{P}_m^{(\kappa_h)}(x) \leq \lambda_m^{\kappa_h} d_h \quad \text{for all } x \in [\tau_{m,\mu-1}, \tau_{m,\mu}].$$

Let us expand each $\tilde{F}_{m,\mu}$ in the Chebyshev basis as

$$\tilde{F}_{m,\mu} = \sum_{k=0}^{n'_{m,\mu}} f_k^{\{m,\mu\}} T_k$$

and let us set $N := \max_{\ell \in \llbracket 1 : M \rrbracket} \max\{n_\ell, \max_{\nu \in \llbracket 1 : \mu_\ell \rrbracket} n'_{\ell,\nu}\}$. Theorem 3 enables us to rewrite the two nonnegativity conditions implicit in (10) as the existence of positive semidefinite matrices $\mathbf{Q}^{-,\{h,m,\mu\}}, \mathbf{Q}^{+,\{h,m,\mu\}} \in \mathbb{C}^{(N+1) \times (N+1)}$ and $\mathbf{R}^{-,\{h,m,\mu\}}, \mathbf{R}^{+,\{h,m,\mu\}} \in \mathbb{C}^{N \times N}$ such that

$$(11) \quad \left\{ \begin{array}{l} \frac{[\mathbf{D}^{\{\kappa_h\}}(\mathbf{f}^{\{m,\mu\}} - \mathbf{p}^{\{m\}})]_k}{\lambda_m^{\kappa_h} d_h + [\mathbf{D}^{\{\kappa_h\}}(\mathbf{f}^{\{m,\mu\}} - \mathbf{p}^{\{m\}})]_0}, \quad k \in \llbracket 1, N \rrbracket, \\ \lambda_m^{\kappa_h} d_h + [\mathbf{D}^{\{\kappa_h\}}(\mathbf{f}^{\{m,\mu\}} - \mathbf{p}^{\{m\}})]_0, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} \mathbf{Q}_{i,j}^{-,\{h,m,\mu\}} \\ + \frac{e^{1\alpha_{m,\mu}}}{2} \left[\sum_{i-j=k-1} R_{i,j}^{-,\{h,m,\mu\}} \right] - \cos(\beta_{m,\mu}) \left[\sum_{i-j=k} R_{i,j}^{-,\{h,m,\mu\}} \right] + \frac{e^{-1\alpha_{m,\mu}}}{2} \left[\sum_{i-j=k+1} R_{i,j}^{-,\{h,m,\mu\}} \right], \\ \left\{ \begin{array}{l} \frac{[\mathbf{D}^{\{\kappa_h\}}(-\mathbf{f}^{\{m,\mu\}} + \mathbf{p}^{\{m\}})]_k}{\lambda_m^{\kappa_h} d_h + [\mathbf{D}^{\{\kappa_h\}}(-\mathbf{f}^{\{m,\mu\}} + \mathbf{p}^{\{m\}})]_0}, \quad k \in \llbracket 1, N \rrbracket, \\ \lambda_m^{\kappa_h} d_h + [\mathbf{D}^{\{\kappa_h\}}(-\mathbf{f}^{\{m,\mu\}} + \mathbf{p}^{\{m\}})]_0, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} \mathbf{Q}_{i,j}^{+,\{h,m,\mu\}} \\ + \frac{e^{1\alpha_{m,\mu}}}{2} \left[\sum_{i-j=k-1} R_{i,j}^{+,\{h,m,\mu\}} \right] - \cos(\beta_{m,\mu}) \left[\sum_{i-j=k} R_{i,j}^{+,\{h,m,\mu\}} \right] + \frac{e^{-1\alpha_{m,\mu}}}{2} \left[\sum_{i-j=k+1} R_{i,j}^{+,\{h,m,\mu\}} \right],$$

where $\alpha_{m,\mu} := (\arccos(\tau_{m,\mu-1}) + \arccos(\tau_{m,\mu}))/2$ and $\beta_{m,\mu} := (\arccos(\tau_{m,\mu-1}) - \arccos(\tau_{m,\mu}))/2$.

In summary, the problem (4) for the max-norm has been recast as the semidefinite program

$$\begin{array}{ll} \text{minimize} & \|\gamma \odot \mathbf{d}\|_r \quad \text{subject to} \quad \mathbf{Q}^{-,\{h,m,\mu\}} \succcurlyeq \mathbf{0}, \mathbf{Q}^{+,\{h,m,\mu\}} \succcurlyeq \mathbf{0}, \\ \mathbf{d} \in \mathbb{R}^H, \mathbf{p}^{\{m\}} \in \mathbb{R}^{N+1} & \\ \mathbf{Q}^{-,\{h,m,\mu\}}, \mathbf{Q}^{+,\{h,m,\mu\}} \in \mathbb{C}^{(N+1) \times (N+1)} & \\ \mathbf{R}^{-,\{h,m,\mu\}}, \mathbf{R}^{+,\{h,m,\mu\}} \in \mathbb{C}^{N \times N} & \\ & \text{to } \mathbf{R}^{-,\{h,m,\mu\}} \succcurlyeq \mathbf{0}, \mathbf{R}^{+,\{h,m,\mu\}} \succcurlyeq \mathbf{0}, \\ & \text{to the constraints (11) (or just (9)),} \\ & \text{to inherent smoothness constraints,} \\ & \text{and to the originally prescribed constraints.} \end{array}$$

Weighted max-norm: This setting requires only an appropriate modification of the previous argument. We start from the pendant of (10), namely

$$-\lambda_m^{\kappa_h} d_h \leq \frac{\widetilde{F}_{m,\mu}^{(\kappa_h)}(x) - \widetilde{P}_m^{(\kappa_h)}(x)}{\widetilde{W}_m(x)} \leq \lambda_m^{\kappa_h} d_h \quad \text{for all } x \in [\tau_{m,\mu-1}, \tau_{m,\mu}],$$

where \widetilde{W}_m denotes the restriction to I_m of W transposed to the interval $[-1, 1]$, as in (7). We further distinguish between two cases:

- If W , hence also \widetilde{W}_m , is a polynomial, then the constraints are equivalent to

$$\lambda_m^{\kappa_h} d_h \widetilde{W}_m + \widetilde{F}_{m,\mu}^{(\kappa_h)} - \widetilde{P}_m^{(\kappa_h)} \geq 0, \quad \lambda_m^{\kappa_h} d_h \widetilde{W}_m - \widetilde{F}_{m,\mu}^{(\kappa_h)} + \widetilde{P}_m^{(\kappa_h)} \geq 0, \quad \text{on } [\tau_{m,\mu-1}, \tau_{m,\mu}].$$

Theorem 3 enables us to express these nonnegativity conditions as semidefinite conditions similar to (11), with $[\lambda_m^{\kappa_h} d_h \mathbf{w}^{\{m\}}]_k$ taking the place of $[\lambda_m^{\kappa_h} d_h]_k$, i.e., of 0 for $k \geq 1$ and of $\lambda_m^{\kappa_h} d_h$ for $k = 0$. The size of the positive \mathbf{Q} involved in the semidefinite program is $N_{\text{obj}} := \max\{N, \deg(W)\} + 1$, where $N = \max_{\ell \in \llbracket 1:M \rrbracket} \max\{n_\ell, \max_{\nu \in \llbracket 1:\mu_\ell \rrbracket} n'_{\ell,\nu}\}$ as before.

- If $1/W$, hence also $1/\widetilde{W}_m$, is a polynomial, then writing $\overline{F}^{\{\kappa_h, m, \mu\}}$ and $\overline{P}^{\{\kappa_h, m, \mu\}}$ for the polynomials $\widetilde{F}_{m,\mu}^{(\kappa_h)}/\widetilde{W}_m$ and $\widetilde{P}_m^{(\kappa_h)}/\widetilde{W}_m$, respectively, the constraint are equivalent to

$$\lambda_m^{\kappa_h} d_h + \overline{F}^{\{\kappa_h, m, \mu\}} - \overline{P}^{\{\kappa_h, m, \mu\}} \geq 0, \quad \lambda_m^{\kappa_h} d_h - \overline{F}^{\{\kappa_h, m, \mu\}} + \overline{P}^{\{\kappa_h, m, \mu\}} \geq 0, \quad \text{on } [\tau_{m,\mu-1}, \tau_{m,\mu}].$$

Theorem 3 enables us to express these nonnegativity conditions as semidefinite conditions similar to (11), where $\mathbf{D}^{\{\kappa\}}$ is replaced by $\mathbf{M}_{\widetilde{W}_m} \mathbf{D}^{\{\kappa\}}$, $\mathbf{M}_{\widetilde{W}_m}$ being the matrix transforming the Chebyshev coefficients of a polynomial into the Chebyshev coefficients of this polynomial multiplied by $1/\widetilde{W}_m$. `Chebfun` again comes in handy for its construction. The size of the matrices \mathbf{Q} involved in the semidefinite program is $N_{\text{obj}} := N + \deg(1/W) + 1$.

It is of course possible that neither W nor $1/W$ are polynomials. But since `Chebfun` represents them both as polynomials up to machine precision⁴, we choose between the two possibilities considered above by selecting the smallest size for the matrices involved after comparing $\max\{N, \deg(W)\}$ and $N + \deg(1/W)$.

q -norm for an even integer q : The condition $\|(F - P)^{(\kappa_h)}\|_q \leq d_h$ is again expressible in terms of the Chebyshev coefficients of each \widetilde{P}_m , but the resulting constraint will not be a semidefinite

⁴the case where W is a piecewise polynomial is not treated here, even though similar ideas would apply.

condition. We start by noticing that

$$\begin{aligned}
 \|(F - P)^{(\kappa_h)}\|_q^q &= \int_{-1}^1 |(F - P)^{(\kappa_h)}(t)|^q dt = \sum_{m=1}^M \int_{t_{m-1}}^{t_m} |F_m^{(\kappa_h)}(t) - P_m^{(\kappa_h)}(t)|^q dt \\
 &= \sum_{m=1}^M \lambda_m^{-(\kappa_h q + 1)} \int_{-1}^1 |\tilde{F}_m^{(\kappa_h)}(x) - \tilde{P}_m^{(\kappa_h)}(x)|^q dx \\
 &= \sum_{m=1}^M \lambda_m^{-(\kappa_h q + 1)} \sum_{\mu=1}^{\mu_m} \int_{\tau_{m,\mu-1}}^{\tau_{m,\mu}} |\tilde{F}_{m,\mu}^{(\kappa_h)}(x) - \tilde{P}_m^{(\kappa_h)}(x)|^q dx.
 \end{aligned}$$

Let us introduce slack variables $\delta_{1,1}^{\{h\}}, \dots, \delta_{1,\mu_1}^{\{h\}}, \dots, \delta_{M,1}^{\{h\}}, \dots, \delta_{M,\mu_M}^{\{h\}}$ with

$$(12) \quad \int_{\tau_{m,\mu-1}}^{\tau_{m,\mu}} |\tilde{F}_{m,\mu}^{(\kappa_h)}(x) - \tilde{P}_m^{(\kappa_h)}(x)|^q dx \leq \frac{\tau_{m,\mu} - \tau_{m,\mu-1}}{2} \left[\delta_{m,\mu}^{\{h\}} \right]^q, \quad m \in \llbracket 1 : M \rrbracket, \quad \mu \in \llbracket 1 : \mu_m \rrbracket,$$

so that the condition $\|(F - P)^{(\kappa_h)}\|_q^q \leq d_h^q$ translates into

$$\sum_{m=1}^M \sum_{\mu=1}^{\mu_m} \lambda_m^{-(\kappa_h q + 1)} \frac{\tau_{m,\mu} - \tau_{m,\mu-1}}{2} \left[\delta_{m,\mu}^{\{h\}} \right]^q \leq d_h^q.$$

Now, in view of $q =: 2q'$ being an even integer, the function $\left| \tilde{F}_{m,\mu}^{(\kappa_h)} - \tilde{P}_m^{(\kappa_h)} \right|^q = \left\{ \tilde{F}_{m,\mu}^{(\kappa_h)} - \tilde{P}_m^{(\kappa_h)} \right\}^{2q'}$ is a polynomial of degree at most $2q'N$, hence the integral in (12) is computed exactly by a Gaussian quadrature involving a shifted version of the zeros $\zeta_0, \dots, \zeta_{Nq'}$ of the Legendre polynomial of degree $Nq' + 1$. Precisely, there are explicit weights $\omega_0, \dots, \omega_{Nq'}$ such that

$$\begin{aligned}
 \int_{\tau_{m,\mu-1}}^{\tau_{m,\mu}} |\tilde{F}_{m,\mu}^{(\kappa_h)}(x) - \tilde{P}_m^{(\kappa_h)}(x)|^q dx &= \frac{\tau_{m,\mu} - \tau_{m,\mu-1}}{2} \sum_{i=0}^{Nq'} \omega_i \left\{ \tilde{F}_{m,\mu}^{(\kappa_h)}(\zeta_i^{\{m,\mu\}}) - \tilde{P}_m^{(\kappa_h)}(\zeta_i^{\{m,\mu\}}) \right\}^q \\
 &= \frac{\tau_{m,\mu} - \tau_{m,\mu-1}}{2} \sum_{i=0}^{Nq'} \left\{ [\mathbf{G}^{\{h,m,\mu\}}(\mathbf{f}^{\{m,\mu\}} - \mathbf{p}^{\{m\}})]_i \right\}^q,
 \end{aligned}$$

where $\mathbf{G}^{\{h,m,\mu\}} \in \mathbb{R}^{(Nq'+1) \times (N+1)}$ is the matrix with entries $G_{i,j}^{\{h,m,\mu\}} = \omega_i^{1/q} T_j^{(\kappa_h)}(\zeta_i^{\{m,\mu\}})$. In summary, defining vectors $\boldsymbol{\rho}^{\{h\}} \in \mathbb{R}^{M'}$, $M' := \mu_1 + \dots + \mu_M$, by $\rho_{m,\mu}^{\{h\}} = \lambda_m^{-\kappa_h - 1/q} \left(\frac{\tau_{m,\mu} - \tau_{m,\mu-1}}{2} \right)^{1/q}$, the problem (4) for the q -norm has been recast as

$$\begin{aligned}
 \underset{\substack{\mathbf{d} \in \mathbb{R}^H, \mathbf{p}^{\{m\}} \in \mathbb{R}^{N+1} \\ \boldsymbol{\delta}^{\{h\}} \in \mathbb{R}^{M'}}}{\text{minimize}} \quad & \|\boldsymbol{\gamma} \odot \mathbf{d}\|_r & \text{subject to} & \quad \|\boldsymbol{\rho}^{\{h\}} \odot \boldsymbol{\delta}^{\{h\}}\|_q \leq d_h, \\
 & & \text{to} & \quad \|\mathbf{G}^{\{h,m,\mu\}}(\mathbf{f}^{\{m,\mu\}} - \mathbf{p}^{\{m\}})\|_q \leq \delta_{m,\mu}^{\{h\}}, \\
 & & \text{and to} & \quad \text{the originally prescribed constraints.}
 \end{aligned}$$

3.3 Dealing with the original constraints

The best approximation problem (4) can include many types of constraints and the convenience of CVX makes it almost effortless to enforce them, so long as they are convex constraints. Their feasibility need not be ensured beforehand, since the status returned by CVX will provide the information. We list below the constraints that have been implemented in Basc along with their reformulations. The reformulations (13), (14), and (15) are linear conditions, while (16), (17), and (18) are semidefinite conditions.

Smoothness constraints: Enforcing smoothness \mathcal{C}^{s_m} at each breakpoint t_m , $m \in \llbracket 1 : M - 1 \rrbracket$, means that $P_m^{(\kappa)}(t_m) = P_{m+1}^{(\kappa)}(t_m)$ for all $\kappa \in \llbracket 0 : s_m \rrbracket$. The latter constraints are equivalent to the $(s_m + 1)$ conditions given for each $\kappa \in \llbracket 0 : s_m \rrbracket$ by $\lambda_m^{-\kappa} \tilde{P}_m^{(\kappa)}(1) = \lambda_{m+1}^{-\kappa} \tilde{P}_{m+1}^{(\kappa)}(-1)$, i.e., by $\lambda_m^{-\kappa} \sum_{k=0}^n p_k^{\{m\}} T_k^{(\kappa)}(1) = \lambda_{m+1}^{-\kappa} \sum_{k=0}^n p_k^{\{m+1\}} T_k^{(\kappa)}(-1)$, where $n := \max_{\ell \in \llbracket 1 : M \rrbracket} n_\ell$ and with the implicit understanding that $p_k^{\{m\}} := 0$ if $n_m < k \leq n$. Introducing the $(s_m + 1) \times (s_m + 1)$ diagonal matrices $\Lambda^{\{m, s_m\}}$ and $\Lambda^{\{m+1, s_m\}}$ with diagonal entries $1, \lambda_m^{-1}, \dots, \lambda_m^{-s_m}$ and $1, \lambda_{m+1}^{-1}, \dots, \lambda_{m+1}^{-s_m}$, as well as the $(s_m + 1) \times n$ matrices $\mathbf{V}^{+, \{s_m\}}$ and $\mathbf{V}^{-, \{s_m\}}$ with entries $V_{\kappa, k}^{\pm, \{s_m\}} = T_k^{(\kappa)}(\pm 1)$, the smoothness conditions reduce to the linear constraints

$$(13) \quad \Lambda^{\{m, s_m\}} \mathbf{V}^{+, \{s_m\}} \mathbf{p}^{\{m\}} = \Lambda^{\{m+1, s_m\}} \mathbf{V}^{-, \{s_m\}} \mathbf{p}^{\{m+1\}} \quad \text{for all } m \in \llbracket 1 : M - 1 \rrbracket.$$

The matrices $\mathbf{V}^{+, \{s_m\}}$ and $\mathbf{V}^{-, \{s_m\}}$ are easily constructed thanks to Chebfun.

Parity constraints: Assuming consistency with all other conditions (breakpoints, smoothness, interpolation nodes and data, etc.), the parity constraint $P(-x) = \pm P(x)$ for all $x \in [-1, 1]$ translates into $P_m(-x) = \pm P_{M+1-m}(x)$ for all $x \in I_m$ and all $m \in \llbracket 1 : M \rrbracket$. Then, by parity of Chebyshev polynomials, with $\boldsymbol{\eta} := [+1, -1, +1, -1, \dots]^\top$ denoting the vector with alternating $+1$ and -1 entries, the parity constraint reduces to the linear constraints

$$(14) \quad \boldsymbol{\eta} \odot \mathbf{p}^{\{m\}} = \pm \mathbf{p}^{\{M+1-m\}} \quad \text{for all } m \in \llbracket 1 : M \rrbracket,$$

where $\pm = +$ if the approximant is required to be even and $\pm = -$ if it is required to be odd.

Interpolatory constraints: These constraints do not restrict to Lagrange interpolation, as they cover more generally Birkhoff interpolation, meaning that for a set $\kappa = \kappa^{\text{int}}$ of indices $\kappa_1, \dots, \kappa_{H^{\text{int}}}$, it is imposed that all derivatives $P^{(\kappa_h)}$ of the approximant agree at prescribed nodes $x_1^{\{h\}}, \dots, x_{\nu^{\{h\}}}^{\{h\}}$ with prescribed data $y_1^{\{h\}}, \dots, y_{\nu^{\{h\}}}^{\{h\}}$. Often but not always, the data are values of the corresponding derivative of the target function at the nodes, i.e., $y_i^{\{h\}} = F^{(\kappa_h)}(x_i^{\{h\}})$. For each $m \in \llbracket 1 : M \rrbracket$, we determine which $\nu^{\{h, m\}}$ nodes $x_i^{\{h\}}$ belong to the m th subinterval I_m and the interpolatory conditions $P_m^{(\kappa_h)}(x_i^{\{h\}}) = y_i^{\{h\}}$ on this subinterval read $\lambda_m^{-\kappa_h} \tilde{P}_m^{(\kappa_h)}(\lambda_m^{-1}(x_i^{\{h\}} - t_{m-1}) - 1) = y_i^{\{h\}}$, or $\sum_{k=0}^n p_k^{\{m\}} \lambda_m^{-\kappa_h} T_k^{(\kappa_h)}(\lambda_m^{-1}(x_i^{\{h\}} - t_{m-1}) - 1) = y_i^{\{h\}}$. In matrix form, if $\mathbf{B}^{\{h, m\}} \in \mathbb{R}^{\nu^{\{h, m\}} \times (n+1)}$ denotes the matrix with entries $\lambda_m^{-\kappa_h} T_k^{(\kappa_h)}(\lambda_m^{-1}(x_i^{\{h\}} - t_{m-1}) - 1)$ and if $\mathbf{y}^{\{h, m\}} \in \mathbb{R}^{\nu^{\{h, m\}}}$ denotes

the vector with entries $y_i^{\{h\}}$, then the interpolatory conditions reduce to the linear constraints

$$(15) \quad \mathbf{B}^{\{h,m\}} \mathbf{p}^{\{m\}} = \mathbf{y}^{\{h,m\}} \quad \text{for all } m \in \llbracket 1 : M \rrbracket \text{ and all } h \in \llbracket 1 : H_{\text{int}} \rrbracket.$$

The matrices $\mathbf{B}^{\{h,m\}}$ are easily created by `Chebfun`.

Range constraints: These consist in fact of two sets of constraints on the approximant: upper range constraints of the form $P^{(\kappa_h)} \leq U^{\{h\}}$ for some index set $\kappa = \kappa^{\text{ur}}$ and some functions $U^{\{h\}}$ and lower range constraints of the form $P^{(\kappa_h)} \geq L^{\{h\}}$ for some index set $\kappa = \kappa^{\text{lr}}$ and some functions $L^{\{h\}}$. Written as the nonnegativity conditions $U^{\{h\}} - P^{(\kappa_h)} \geq 0$ and $P^{(\kappa_h)} - L^{\{h\}} \geq 0$, the constraints can therefore be expressed as semidefinite conditions. To shorten the argument, we notice the analogy with the two inequalities of (5), with $U^{\{h\}}$ taking the place of $F^{(\kappa_h)} + d_h$ and $L^{\{h\}}$ taking the place of $F^{(\kappa_h)} - d_h$. Thus, without reproducing the reasoning but keeping usual notation and being careful about scaling factors due to the transposition of derivatives when the subinterval I_m is split by breakpoints of $U^{\{h\}}$, the upper range constraints are equivalent to the existence of positive semidefinite matrices $\mathbf{Q}^{\text{ur},\{h,m,\mu\}} \in \mathbb{C}^{(N_{\text{ur}}^{\{h\}}+1) \times (N_{\text{ur}}^{\{h\}}+1)}$ and possibly $\mathbf{R}^{\text{ur},\{h,m,\mu\}} \in \mathbb{C}^{N_{\text{ur}}^{\{h\}} \times N_{\text{ur}}^{\{h\}}}$, where $N_{\text{ur}}^{\{h\}} := \max_{\ell \in \llbracket 1 : M \rrbracket} \max_{\nu \in \llbracket 1 : \mu_\ell \rrbracket} \deg(U_{\ell,\nu}^{\{h\}})$, such that

$$(16) \quad \left\{ \begin{array}{l} \frac{[\mathbf{u}^{\{h,m,\mu\}} - \lambda_m^{-\kappa_h} \mathbf{D}^{\{\kappa_h\}}(\mathbf{p}^{\{m\}})]_k}{2}, \quad k \in \llbracket 1 : N_{\text{ur}}^{\{h\}} \rrbracket, \\ \frac{[\mathbf{u}^{\{h,m,\mu\}} - \lambda_m^{-\kappa_h} \mathbf{D}^{\{\kappa_h\}}(\mathbf{p}^{\{m\}})]_0}{2}, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} Q_{i,j}^{\text{ur},\{h,m,\mu\}} \\ + \frac{e^{i\alpha_{m,\mu}^{\text{ur},\{h\}}}}{2} \left[\sum_{i-j=k-1} R_{i,j}^{\text{ur},\{h,m,\mu\}} \right] - \cos(\beta_{m,\mu}^{\text{ur},\{h\}}) \left[\sum_{i-j=k} R_{i,j}^{\text{ur},\{h,m,\mu\}} \right] + \frac{e^{-i\alpha_{m,\mu}^{\text{ur},\{h\}}}}{2} \left[\sum_{i-j=k+1} R_{i,j}^{\text{ur},\{h,m,\mu\}} \right].$$

Likewise, the lower range constraints can be expressed as the existence of positive semidefinite matrices $\mathbf{Q}^{\text{lr},\{h,m,\mu\}} \in \mathbb{C}^{(N_{\text{lr}}^{\{h\}}+1) \times (N_{\text{lr}}^{\{h\}}+1)}$ and possibly $\mathbf{R}^{\text{lr},\{h,m,\mu\}} \in \mathbb{C}^{N_{\text{lr}}^{\{h\}} \times N_{\text{lr}}^{\{h\}}}$, where $N_{\text{lr}}^{\{h\}} := \max_{\ell \in \llbracket 1 : M \rrbracket} \max_{\nu \in \llbracket 1 : \mu_\ell \rrbracket} \deg(L_{\ell,\nu}^{\{h\}})$, such that

$$(17) \quad \left\{ \begin{array}{l} \frac{[-\mathbf{l}^{\{h,m,\mu\}} + \lambda_m^{-\kappa_h} \mathbf{D}^{\{\kappa_h\}}(\mathbf{p}^{\{m\}})]_k}{2}, \quad k \in \llbracket 1, N_{\text{lr}}^{\{h\}} \rrbracket \\ \frac{[-\mathbf{l}^{\{h,m,\mu\}} + \lambda_m^{-\kappa_h} \mathbf{D}^{\{\kappa_h\}}(\mathbf{p}^{\{m\}})]_0}{2}, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} Q_{i,j}^{\text{lr},\{h,m,\mu\}} \\ + \frac{e^{i\alpha_{m,\mu}^{\text{lr},\{h\}}}}{2} \left[\sum_{i-j=k-1} R_{i,j}^{\text{lr},\{h,m,\mu\}} \right] - \cos(\beta_{m,\mu}^{\text{lr},\{h\}}) \left[\sum_{i-j=k} R_{i,j}^{\text{lr},\{h,m,\mu\}} \right] + \frac{e^{-i\alpha_{m,\mu}^{\text{lr},\{h\}}}}{2} \left[\sum_{i-j=k+1} R_{i,j}^{\text{lr},\{h,m,\mu\}} \right].$$

The values of $\alpha^{\text{ur},\{h\}}$, $\beta^{\text{ur},\{h\}}$, $\alpha^{\text{lr},\{h\}}$, and $\beta^{\text{lr},\{h\}}$ in (16) and (17) arise from the splitting of the subinterval I_m by breakpoints of $U^{\{h\}}$ and $L^{\{h\}}$.

Shape constraints: These constraints concern the sign patterns of the derivatives $P^{(\kappa_h)}$ of the approximant given some index set $\kappa = \kappa^{\text{sh}}$. We express them as

$$S^{\{h\}}(x)P^{(\kappa_h)}(x) \geq 0 \quad \text{for all } x \in [-1, 1],$$

where the $S^{\{h\}}$ are some prescribed piecewise polynomials. As an example, if it is required that $P^{(\kappa_h)}$ changes sign at points z_1, \dots, z_{ν_h} and finishes by being nonnegative on $(z_{\nu_h}, 1)$, then we would

set $S^{\{h\}}(x) = (x - z_1) \cdots (x - z_{v_h})$. As another example, if it was required that $P^{(\kappa_h)}$ is nonnegative on some interval $[a, b]$ without constraints outside $[a, b]$, then we would set $S^{\{h\}}(x) = \mathbf{1}_{[a,b]}(x)$, the characteristic function of $[a, b]$. But beware, for instance, that the convexity of P on $[a, b]$ is not necessarily the same as the nonnegativity of P'' on $[a, b]$ because of potential breakpoints, or that the coconvexity of P with F is not necessarily the same as the condition $F''P'' \geq 0$, as the example of $F(x) = |x|$ reveals. At this stage of **Basc**'s development, the user is responsible for choosing the correct $S^{\{h\}}$. With the usual notation, the shape constraints become $\tilde{S}_{m,\mu}^{\{h\}} \tilde{F}_m^{(\kappa_h)} \geq 0$ on $[\tau_{m,\mu-1}, \tau_{m,\mu}]$ for all m and μ . In turn, they are equivalent to the existence of positive semidefinite matrices $\mathbf{Q}^{\text{sh},\{h,m,\mu\}} \in \mathbb{C}^{(N_{\text{sh}}^{\{h\}}+1) \times (N_{\text{sh}}^{\{h\}}+1)}$ and possibly $\mathbf{Q}^{\text{sh},\{h,m,\mu\}} \in \mathbb{C}^{N_{\text{sh}}^{\{h\}} \times N_{\text{sh}}^{\{h\}}}$, where $N_{\text{sh}}^{\{h\}} := n + \max_{\ell \in \llbracket 1:M \rrbracket} \max_{\nu \in \llbracket 1:\mu_\ell \rrbracket} \deg(S_{\ell,\nu}^{\{h\}})$, such that

$$(18) \quad \left\{ \begin{array}{l} \frac{[\mathbf{M}_{\tilde{S}_{m,\mu}^{\{h\}}} \mathbf{D}^{\{\kappa_h\}}(\mathbf{p}^{\{m\}})]_k}{2}, \quad k \in \llbracket 1, N_{\text{sh}}^{\{h\}} \rrbracket \\ [\mathbf{M}_{\tilde{S}_{m,\mu}^{\{h\}}} \mathbf{D}^{\{\kappa_h\}}(\mathbf{p}^{\{m\}})]_0, \quad k = 0, \end{array} \right\} = \sum_{i-j=k} Q_{i,j}^{\text{sh},\{h,m,\mu\}} \\ + \frac{e^{i\alpha_{m,\mu}^{\text{sh},\{h\}}}}{2} \left[\sum_{i-j=k-1} R_{i,j}^{\text{sh},\{h,m,\mu\}} \right] - \cos(\beta_{m,\mu}^{\text{sh},\{h\}}) \left[\sum_{i-j=k} R_{i,j}^{\text{sh},\{h,m,\mu\}} \right] + \frac{e^{-i\alpha_{m,\mu}^{\text{sh},\{h\}}}}{2} \left[\sum_{i-j=k+1} R_{i,j}^{\text{sh},\{h,m,\mu\}} \right].$$

Here, each $\mathbf{M}_{\tilde{S}_{m,\mu}^{\{h\}}}$ is the $(N_{\text{sh}}^{\{h\}} + 1) \times (n + 1)$ matrix that transforms the Chebyshev coefficients of a polynomial into the Chebyshev coefficients of this polynomial multiplied by $\tilde{S}_{m,\mu}^{\{\ell\}}$. It is easily constructed via **Chebfun**.

4 Usage Examples for Basc

This section gives a brief description of the **MATLAB** package **Basc** and of its basic usage. More details are available in the User's Guide found on the first author's webpage. After downloading **Basc**'s files, which include all **Chebfun**'s and **CVX**'s files, one need to type

```
>>basc_setup
```

in order to launch the functionalities of these two external packages. It is also convenient to set

```
>>x=chebfun('x')
```

in order to easily define chebfuns — the objects manipulated by the software **Chebfun** — since **Basc** takes chebfuns as inputs and also returns chebfuns as outputs. Thus, given a chebfun F to be approximated by a polynomial (or spline) of degree at most n (or degrees at most n_1, \dots, n_M), the generic command is

```
>>[e,P]=basc(F,n,options)
```

The output **e** is the error of best approximation, while the output **P** is a best approximant. The optional input **options** contain a variable number of fields that can be combined freely and specified in any order. They are inventoried in Table 1. There are also various shortcuts, such as 'above' or 'below' for the lower range $L^{\{0\}} = F$ or the upper range $U^{\{0\}} = F$, 'positive' or 'negative'

field	math. objects	format
'norm'	q	an even integer of <code>inf</code>
'weight'	W	a classic chebfun representing W
'bkpts'	t_1, \dots, t_M	a row-vector containing the t_i
'smoothness'	s_1, \dots, s_m	a row-vector containing the s_i
'parity'	parity of P	'even' or 'odd'
'interpolation'	κ^{int} , the $(\mathbf{x}^{\{h\}}, \mathbf{y}^{\{h\}})$	a cell of cells, each inner cell containing first κ_h , then $(\mathbf{x}^{\{h\}}, \mathbf{y}^{\{h\}})$ as a $2 \times \nu^{\{h\}}$ matrix (or a row vector if $y_i^{\{h\}} = F(x_i^{\{h\}})$)
'upper range'	κ^{ur} , the $U^{\{h\}}$	a cell of cells, each inner cell containing first κ_h , then the chebfun for $U^{\{h\}}$
'lower range'	κ^{lr} , the $L^{\{h\}}$	a cell of cells, each inner cell containing first κ_h , then the chebfun for $L^{\{h\}}$
'shape'	κ^{sh} , the $S^{\{h\}}$	a cell of cells, each inner cell containing first κ_h , then the chebfun for $S^{\{h\}}$
'objective'	$r, \kappa^{\text{obj}}, \gamma^{\text{obj}}$	a cell whose first element is r and other elements are cells containing κ_h and γ_h

Table 1: The different fields that can be specified in `Basc`'s options

for the lower range $L^{\{0\}} = 0$ or the upper range $U^{\{0\}} = 0$, 'increasing' or 'decreasing' for the shape function $S^{\{1\}} = 1$ or $S^{\{1\}} = -1$, and 'convex' or 'concave' for the shape function $S^{\{2\}} = 1$ or $S^{\{2\}} = -1$.

Below is a list of examples illustrating the usage of `Basc`. These computations can be replicated after the reproducible file of this article is downloaded from the first author's webpage.

One-sided polynomial approximation: One-sided approximants to a target function F can be computed by entering `{0,F}` in the fields 'lower range' and 'upper range' or by using the shortcuts 'above' and 'below'. So to obtain the approximants by polynomials of degree 4 to $F(x) = T_5(x)/(4 + \cos(x) + \sin(x))$, say, relative to the max-norm and to the 2-norm, one would type

```
>>n = 4; F = chebpoly(5)./(4+cos(x)+sin(x));
>>[e,P] = basc(F,n); [e2,P2] = basc(F,n,'norm',2);
>>[ea,Pa] = basc(F,n,'above'); [ea2,Pa2] = basc(F,n,'above','norm',2);
>>[eb,Pb] = basc(F,n,'below'); [eb2,Pb2] = basc(F,n,'below','norm',2);
```

Some of the resulting approximants P , as well as the errors of approximation $F - P$, are displayed on Figure 1. On the middle graph, one recognizes the equioscillation property for unconstrained

approximation relative to the max-norm and one also observes that the one-sided approximants are just shifted versions of the unconstrained approximant — a fact that is easy to establish. The rightmost graph reveals the failure of this phenomenon relative to the 2-norm.

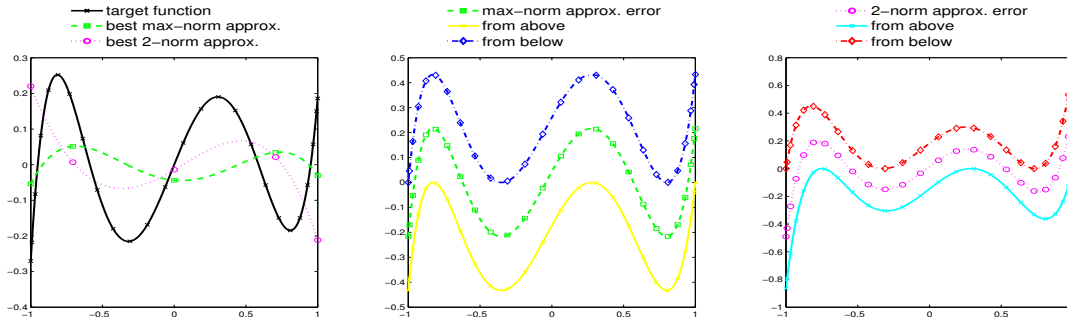


Figure 1: Unconstrained approximants (left) and errors of one-sided approximation relative to the max-norm (center) and to the 2-norm (right).

Weighted and interpolatory polynomial approximation: Computing best approximants relative to a weighted max-norm is done by entering the weight W as a chebfun after the field 'weight'. We may append further constraints, for instance, the field 'parity' takes the value 'even' or 'odd', and the field 'interpolation' takes the indices of the derivatives to interpolate together with the corresponding interpolation nodes and optionally the interpolation data. The outcome of the commands

```
>>n = 5; F = chebpoly(7)./(4+cos(x)+sin(x));
>>[ew,Pw] = basc(F,n,'weight',x.^2);
>>[ewo,Pwo] = basc(F,n,'weight',x.^2,'parity','odd');
>>[ewi,Pwi] = basc(F,n,'weight',x.^2,'interpolation',{0,[-1 1]});
```

are displayed in Figure 2. Note that the weight $W(x) = x^2$ vanishes two-fold at $x = 0$, which forces the approximants to agree two-fold with the target function at $x = 0$.

Spline approximation: In order to approximate a target function by a piecewise polynomial instead of a mere polynomial, one specifies the internal breakpoints as a row-vector in the field 'bkpts'. By default, no continuity conditions are imposed, but if smoothness at the breakpoints is desired, then we also enter the smoothness parameters as a row-vector in the field 'smoothness', with -1 for discontinuous, 0 for continuous, 1 for continuously differentiable, etc. For instance, the commands

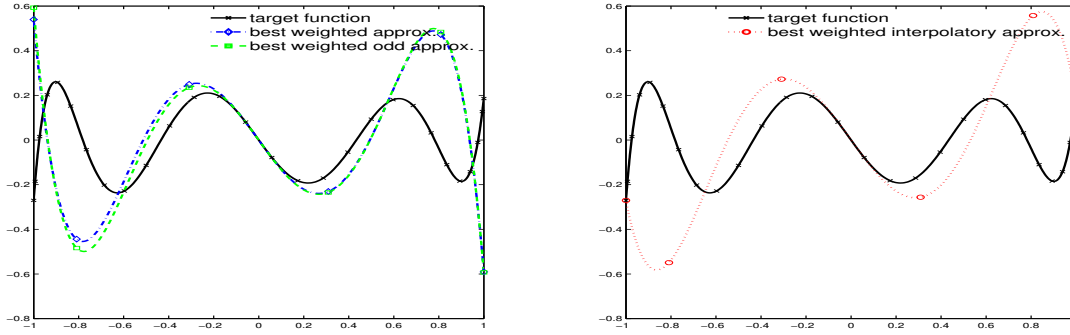


Figure 2: Best approximants relative to the weighted norm $\|G\|_W = \|G/W\|_\infty$ with $W(x) = x^2$ (left) and with the additional interpolatory constraints $P(-1) = F(-1)$ and $P(1) = F(1)$ (right).

```
>>n = 2; F = chebfun(rand(2*n+1,1),'coeffs');
>>[es,Ps] = basc(F,2*n-1,'bkpts',[0],'smoothness',[-1]);
>>[es2,Ps2] = basc(F,2*n-1,'bkpts',[0],'smoothness',[1]);
```

compute the approximants to a polynomial of degree $2n$ by splines of degree $2n - 1$ with one breakpoint at $x = 0$. The first approximant is a priori discontinuous and the second approximant is a priori continuously differentiable. These spline approximants are shown on Figure 3 (left), and we notice that the \mathcal{C}^{-1} approximant is automatically \mathcal{C}^0 — a fact that is again easy to establish — and that the \mathcal{C}^1 approximant is automatically \mathcal{C}^2 .

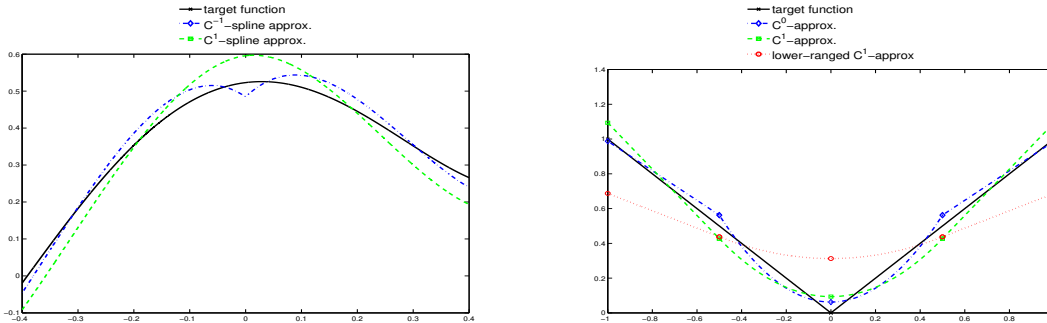


Figure 3: The a priori \mathcal{C}^{-1} and \mathcal{C}^1 best approximants to a polynomial of degree $2n$ by splines of degree $2n - 1$ with one breakpoint at $x = 0$ (left); the \mathcal{C}^0 , \mathcal{C}^1 , and lower-ranged \mathcal{C}^1 best approximants to the absolute value function by splines of degrees $(1, 2, 1)$ with breakpoints at $x = -1/2$ and $x = 1/2$ (right).

In the next example, we produce best approximants by splines to a target function that is a spline itself — the prototypical absolute value function. With breakpoints at $x = -1/2$ and $x = 1/2$ and polynomial pieces of degree 1, 2, and 1, we compute the best approximants that are \mathcal{C}^0 , \mathcal{C}^1 , and \mathcal{C}^1 with the additional constraint of having a derivative lower-bounded by $L(x) = -1/2$ via the commands

```
>>F = abs(x);
>>[e0,P0]=basc(F,[1 2 1], 'bkpts', [-1/2 1/2], 'smoothness', [0 0]);
>>[e1,P1]=basc(F,[1 2 1], 'bkpts', [-1/2 1/2], 'smoothness', 'maximal');
>>[elr,Plr]=basc(F,[1 2 1], 'bkpts', [-1/2 1/2], 'smoothness', 'maximal',...
    'lower range', {{1, chebfun(-0.5)}});
```

The approximants are reported on Figure 3 (right). Note that the example reveals that a best approximation by splines is not necessarily unique, since we can observe that perturbing the slopes of the outside pieces of the \mathcal{C}^0 -spline does not affect the overall error.

Bernstein-type constants: It is well known that the rate of approximation error to the absolute value function by polynomials of degree n is governed by the fact that

$$(19) \quad \beta := \lim_{n \rightarrow \infty} n E_n(|\cdot|)$$

exists. This result is due to Bernstein [1], who conjectured the value $1/(2\sqrt{\pi}) \approx 0.282094791773$ for β . This conjecture was refuted in [14] where it was shown that $\beta \approx 0.280169499023$. It is also known (see the survey [6] and the references therein for more results) that the rate of approximation error to the absolute value function by convex polynomials of degree n is governed by the fact that

$$\beta^{\text{conv}} := \lim_{n \rightarrow \infty} n E_n^{\text{conv}}(|\cdot|)$$

exists, too. The value of β^{conv} can be investigated experimentally using Basc (save for current limitations with the scale of the problem). The result of such an experiment⁵ together with a numerical validation of the value of (19) are presented in Figure 5. As an aside, we first display in Figure 4 what the best approximant, the best convex approximant, and the best approximant convex on $[-2/3, 2/3]$ to the absolute value function look like when $n = 5$. These approximants are obtained via the commands

```
>>n = 5; F = abs(x);
>>[e,P] = basc(F,n);
>>[ecvx,Pcvx] = basc(F,n, 'convex');
>>[ecvx2,Pcvx2] = basc(F,n, 'shape', {{2, -(x.^2-4/9)}});
```

⁵ $E_n^{\text{conv}}(|\cdot|)$ was not computed by Basc directly, but by the semidefinite program written specifically for this case in order to limit the defect of a multipurpose code.

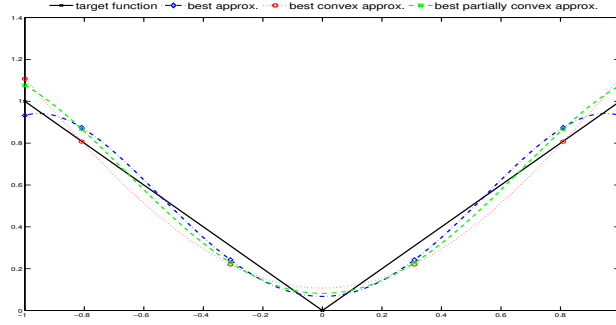


Figure 4: Degree-5 approximants to $|\cdot|$: unconstrained, convex, and convex on $[-2/3, 2/3]$.

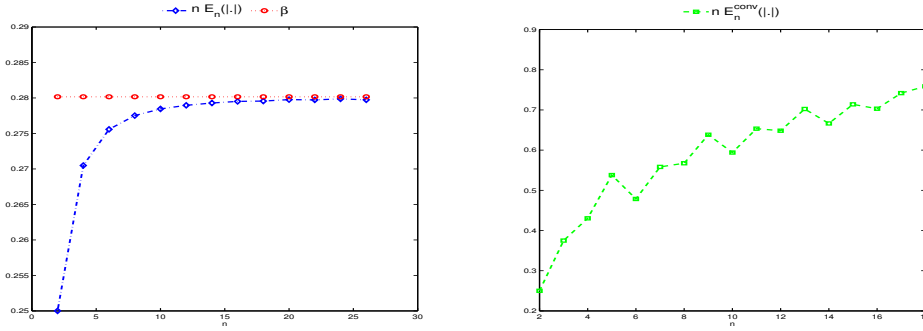


Figure 5: Verification of the value of the Bernstein constant (left) and exploration of the value of the convex-Bernstein constant (right).

Natural cubic spline: We finally consider an example where we modify the objective function from its default setting in order to solve an optimization problem not immediately recognizable as an approximation problem. Precisely, we are interested in the function $P \in \mathcal{C}^2[-1, 1]$ that minimizes the quasinorm $\|P''\|_2$ under the constraints $P(t_i) = y_i$ with $-1 = t_0 < t_1 < \dots < t_{M-1} < t_M = 1$. It is well known that the unique solution to this problem is the so-called natural spline determined by the fact that it has degree 3 and overall smoothness \mathcal{C}^2 , that is satisfies the interpolatory constraints, and that $P''(t_0) = 0$ and $P''(t_M) = 0$. To corroborate this statement, we perform the minimization not over all $P \in \mathcal{C}^2[-1, 1]$ but only over all \mathcal{C}^2 splines of degree 5, say, with breakpoints t_1, \dots, t_M by solving the problem (1) with $F = 0$, $H = 1$, $\kappa_1 = 2$, and $\gamma_1 = 1$. This is done with the commands

```
>>n = 5; t = [-1/3 1/3]; s = [2 2];
>>[e,P] = basc(chebfun(0),n,'objective',{inf,{2,1}},'norm',2,'bkpts',t,...
    'smoothness',s,'interpolation',{0,[-1 t 1;1 2 0 1]},{2,[-1 1;0 0]});
```

We can verify on Figure 6 (right) that the second derivative of the minimizer is indeed a piecewise linear function that vanishes at the endpoints -1 and 1 .

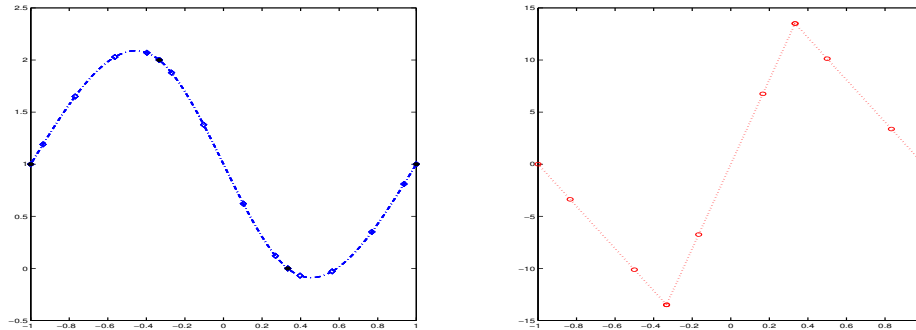


Figure 6: The natural cubic spline with interpolations nodes $(-1, -1/3, 1/3, 1)$ and interpolation data $(1, 2, 0, 1)$, as well as its second derivative.

5 Conclusion

Basc adds a modest contribution to a vision of Approximation Theory becoming more computation-embracing (a vision that is implicit in [13]). When creating this package, our goal was to provide a tool for gaining insight on theoretical problems through numerical experimentation. Although the underlying method is conceptually sound, we consider the current implementation to be at the ‘proof-of-concept’ stage, as issues with reliability and speed sometimes occur. It is critical to determine if, for instance, difficulties with large scales can be bypassed in some plausibly simple way (by a better implementation or by the use of a targeted solver instead of multipurpose software) or if they are inherent to the task at hand. In the latter case, could changing the semidefinite-programming-based strategy have the desired effect? Answering these questions is concomitant with the development of several Basc extensions that we currently envision.

References

- [1] S. Bernstein. Sur la meilleure approximation de $|x|$ par des polynomes de degrés donnés. *Acta Mathematica*, 37(1), 1–57, 1914.
- [2] B. L. Chalmers. The Remez exchange algorithm for approximation with linear restrictions. *Transactions of the American Mathematical Society*, 223, 103–131, 1976.

-
- [3] CVX Research, Inc. CVX: MATLAB software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, 2014.
 - [4] B. A. Dumitrescu. Positive trigonometric polynomials and signal processing applications. Springer Science & Business Media, 2007.
 - [5] K. R. Gehner. Characterization theorems for constrained approximation problems via optimization theory. *Journal of Approximation Theory*, 14(1), 51–76, 1975.
 - [6] K. A. Kopotun, D. Leviatan, A. Prymak, and I. A. Shevchuk. Uniform and Pointwise Shape Preserving Approximation by Algebraic Polynomials. *Surveys in Approximation Theory*, 6, 24–74, 2011. <http://www.math.technion.ac.il/sat>
 - [7] A. Kroó and A. Pinkus. Strong Uniqueness. *Surveys in Approximation Theory*, 5, 1–91, 2010. <http://www.math.technion.ac.il/sat>
 - [8] J. T. Lewis. Approximation with convex constraints. *SIAM Review*, 15(1), 193–217, 1973.
 - [9] T. Roh and L. Vandenberghe. Discrete transforms, semidefinite programming, and sum-of-squares representations of nonnegative polynomials. *SIAM Journal on Optimization*, 16(4), 939–964, 2006.
 - [10] G. Szegő. *Orthogonal polynomials* (rev. ed.). Amer. Math. Soc, 1959.
 - [11] G. D. Taylor, and M. J. Winter. Calculation of best restricted approximations. *SIAM Journal on Numerical Analysis*, 7(2), 248–255, 1970.
 - [12] L. N. Trefethen et al.. *Chebfun Version 5*, The Chebfun Development Team, 2014, <http://www.chebfun.org>.
 - [13] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2013.
 - [14] R. S. Varga and A. J. Carpenter. On the Bernstein conjecture in approximation theory. *Constructive Approximation* 1(1), 333–348, 1985.