Polynomials with bounds and numerical approximation

Bruno Després

Polynomials with bounds and numerical approximation

Bruno Després*

April 27, 2016

Abstract

We discuss the generation of polynomials with two bounds -an upper bound and a lower bound- on compact sets $\mathcal{C} = [0,1]^d \subset \mathbb{R}^d$. We show that a composition formula based on a weighted 4-squares Euler identity generates all such polynomials in dimension $d = 1$. Higher dimensions are discussed by means of the 8-squares Degen identity and tensorization, and the connection with quaternions algebras is made explicit. Various numerical results illustrate the potentialities of this approach and some implementation details are provided.

1 Introduction

This work presents algorithms of algebraic nature for the generation of polynomials with two bounds on compact sets and explore the use of such methods for the numerical approximation of simple functions. As far as the author is aware, it is the first time that the fundamental algebraic properties of polynomials with two bounds -an upper bound and a lower bound- are discussed in view of their use in numerical analysis and scientific computing.
A first model problem in dimension $d = 1$ writes: Generate all polynomials with two bounds

$$p_n \in U_n \equiv \{p_n \in P_n(x), \quad \text{such that } 0 \leq p_n(x) \leq 1 \quad \forall x \in [0,1]\}. \tag{1}$$

Polynomials with one bound are denoted as

$$p_n \in P_n^+ \equiv \{p_n \in P_n(x), \quad \text{such that } 0 \leq p_n(x) \quad \forall x \in [0,1]\}. \tag{2}$$

So $p_n \in U_n$ if and only if $p_n \in P_n^+$ and $1 - p_n \in P_n^+$. Simpler subsets of U_n exist based on convex combinations $q_n = \sum_{j=0}^n \alpha_j u_j$ where the coefficients satisfy $0 \leq \alpha_j$ and $\sum_{j=0}^n \alpha_j = 1$. The generating polynomials u_j can be either the basis of the monomials x^j, or the basis of the Berstein polynomials

*Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France.

$B_{n,j}(x) = \frac{n!}{j!(n-j)!}x^j(1-x)^{n-j}$, or the basis of the rescaled Tchebycheff polynomials $\frac{T_j(2x-1)+1}{2}$. However none of these subsets is able to generate all polynomials in U_n only by convex combinations.

Our interest in the set U_n stems from its underlying considerable (but non explicit) role in scientific computing and non linear approximation. For example the theory of limiters developed for the numerical approximation of conservation laws [11, 18, 14] intends to guarantee that a high order polynomial approximation of a given function f respects some maximum principle even if f has very low regularity (f can be discontinuous at shocks). The fundamental reason is that non linear equations need control of various L^∞ bounds to be numerical tractable. Another example [13, 7] is the optimal control of polynomial systems and non negative polynomials in P_n^+. We refer to [10] for algorithmic issues in the context of compress sensing. A third example is from reduced modeling, typically with POD techniques [15], and from interpolation on a sparse grid [12]. Uncertainty quantification intrusive techniques are also highly demanding in terms of having polynomial approximations with a priori unconditional respect of the maximum principle [6]. This work addresses a new approach which in principle could be used in all these fields and which achieves a priori robustness in terms of the satisfaction of the maximum principle without any condition on the local accuracy of the approximation.

The basis of the algorithms is a weighted version of the four-squares Euler identity which extends to the eight-squares Degen identity. This is related to classical issues in algebraic geometry for which we refer to [16, 1] and therein, a domain which goes far beyond the expertise of the author. It seems nevertheless that the use of such methods (related to quaternions [9], quaternions algebras and quaternions basis, one can refer to the expository paper [5] and therein) for the generation of signed polynomials on convex sets has not been addressed in the literature. In a different direction we quote the recent works [3] on issues in scientific computing with quaternions and octonions.

The organization of this work is as follows. Section 2 is dedicated to the presentation of algorithms for the generation of polynomials with one bound and with two bounds. The algorithms are based on composition formulas justified by various generalizations with convenient weights of the four-squares Euler identity. The main theorems are proved, theorem 2.5 shows that the loop based on a weighted four-square Euler identity generates all polynomials in U_n and theorem 2.9 gives the fundamental estimate in uniform norm: it has, for $0 \leq f \leq 1$, the same asymptotic accuracy than the classical approximation with polynomials in P_n. The loops for polynomials of many variables in $\mathcal{C} = [0,1]^d$ are addressed in section 3. The Hurwitz theorem yields the standard limitation on many squares identity. This is why an extension with the Degen identity is proposed which extends to higher dimensions by tensorization. The link with quaternions algebras and quaternions basis is made explicit in remark 3.2. Numerical exploration of the possibilities offered by these algorithms is performed in section 4 within a Matlab based test code. It shows good accuracy and stability of the methods. Perspectives are evoked in the final section 5.

Notations. Since we are ultimately interested in application of polynomials with bounds to scientific computing, we privilege natural notations where $p_n(x, y, \dots)$ simply means a polynomial of the real variables $x, y, \dots$ of degree less or equal to n, and P_n in dimension $d = 1$ refers to $P_n(x)$. We use p, a, b, $\dots$, for polynomials when the context makes the notation non ambiguous. Most of the material developed below generalizes immediately to other algebras.

2 Polynomials with bounds in dimension $d = 1$

The starting point of the analysis is the Lukacs theorem [17, 13] which is a representation theorem for non negative polynomial $p_n \in P_n^+$.

Theorem 2.1 (Lukacs theorem)**.** *Two cases occur.*
• *Either $n = 2m \in 2\mathbb{N}$, then there exists $a_m \in P_m$ and $b_{m-1} \in P_{m-1}$ such that*

$$p_n = a_m^2 + b_{m-1}^2 w \qquad w(x) = x(1-x). \tag{3}$$

• *Or $n + 1 = 2m \in 2\mathbb{N}$, then there exists $a_{m-1} \in P_m$ and $b_{m-1} \in P_{m-1}$ such that*

$$p_n = a_{m-1}^2 w_1 + b_{m-1}^2 w_2 \qquad w_1(x) = x, \quad w_2(x) = 1 - x. \tag{4}$$

Since $P_n^+ \subset P_{n+1}^+ \subset P_{n+2}^+$, the two representations formulas apply for any polynomial. The representations are non unique: for example one can always consider $-a_m$ instead of a_m which a trivial case of non uniqueness of the representation; another example of the non uniqueness writes as $1 = 1^0 + 0^2 w = (1-2x)^2 + 2^2 w(x)$.

To obtain a self contained work, we begin with a simple proof of the first representation (3). Consider two polynomials p_n and $q_{n'}$ with the first representation

$$\begin{cases} p_n = a_m^2 + b_{m-1}^2 w, & n = 2m \\ q_{n'} = \alpha_{m'}^2 + \beta_{m'-1}^2 w, & n' = 2m'. \end{cases}$$

Consider the composition

$$\begin{cases} A_{n+n'} = a_m \alpha_{m'} + b_{m-1} \beta_{m'-1} w, \\ B_{n+n'-1} = a_m \beta_{m'-1} - b_{m-1} \alpha_{m'} \end{cases} \tag{5}$$

and define

$$P_{2n+2n'} = A_{n+n'}^2 + B_{n+n'-1}^2 w \in P_{2n+2n'}^+.$$

The Bramagupta-Fibonacci formula yields

$$\begin{aligned} &\left(a_m \alpha_{m'} + b_{m-1} \beta_{m'-1} w\right)^2 + \left(a_m \beta_{m'-1} - b_{m-1} \alpha_{m'}\right)^2 w \\ &\qquad = \left(a_m^2 + b_{m-1}^2 w\right) \left(\alpha_{m'}^2 + \beta_{m'-1}^2 w\right), \end{aligned}$$

that is

$$P_{2n+2n'} = p_n q_{n'} \tag{6}$$

which shows that the product of two polynomials which admit the first representation (3) also admits the first representation (3). It yields a simple proof of the first part of the Lukacs theorem.

Proof of the representation formula (3). Start with a polynomial $p_n \in P_n^+$, with degree exactly $n = 2m$ for simplicity and write the decomposition with the roots $z_i \in \mathbb{C}$

$$p_n(x) = k\Pi_{i=0}^{n}(x - z_i), \quad k \in \mathbb{R}.$$

Since p_n is real, the non real roots have their complex conjugate in the list of roots. Since $p_n(x) \geq 0$ for $0 \leq x \leq 1$, all real real roots $z_i \in (0, 1)$ have a even degree. So one can always group the roots two by two (and group on of these products with the multiplicative constant k) such that

$$p_n(x) = \Pi_{i=1}^{m=n/2} q_i(x) \tag{7}$$

where $q_i \in P_2^+$ for all i. A basic reasoning[1] with second order polynomial shows the existence of $a_i \in P_1$ and $b_i \in \mathbb{R}$ such that

$$q_i = a_i^2 + b_i^2 w \in P_2^+. \tag{8}$$

So p_n is the product of second order polynomials which all admit the first representation (3). It is sufficient to compose these polynomials one after another with the algebra (5-6) to obtain the claim. The proof is ended. □

2.1 Generalisation to polynomials in U_n

To simplify the notations, we will disregard the index which refers to the maximal degree of the polynomials. Let us consider $p \in U_n$. Since $p \in P_n^+$ the Lukacs theorem yields the representation

$$p = a^2 w_1 + b^2 w_2 \tag{9}$$

where a and b are polynomials with convenient degree and w_1 and w_2 are the weights of the representation formula that has been chosen. In the first case $w_1(x) = 1$ and $w_2(x) = x(1-x)$. In the second case $w_1(x) = x$ and $w_2(x) = 1-x$.

Considering that $p \in U_n$, one has also that $1 - p \in P_n^+$, that is

$$1 - p = c^2 w_3 + d^2 w_4 \tag{10}$$

where c and d are polynomials with convenient degree. The weights w_3 and w_4 are a priori equal to w_1 and w_2. Nevertheless one could think of using the first representation formula in (9) and the second representation formula in (10): in this case the weights $w_{1,2,3,4}$ are all different. By summation, one sees that $p \in U_n$ if and only there exists polynomials a, b, c, d such $p = a^2 w_1 + b^2 w_2$ and

$$1 = a^2 w_1 + b^2 w_2 + c^2 w_3 + d^2 w_4. \tag{11}$$

[1] Let $p \in P_2^+$ with $p(x) = a+bx+c^2$. So $p(0) = a \geq 0$ and $p(1) = a+b+c \geq 0$. For simplicity assume that $a > 0$. Set $q(x) = \sqrt{a+b+c}x - \sqrt{a}x$ and $x_* = \sqrt{a}/(\sqrt{a} + \sqrt{a+b+c}) \in (0, 1]$. Note that $q(0)^2 = p(0)$, $q(1)^2 = p(1)$ and $q(x_*) = 0$. By construction there exists $d \in \mathbb{R}$ such that $p(x) - q(x)^2 = dx(1-x)$. Then $p(x_*) = dx_*(1-x_*)$ shows that $d \geq 0$. One can write $d = e^2$, which shows that $p(x) = q(x)^2 + e^2(x - x^2)$. If $a = 0$ the proof is adapted by taking $x_* = 0$.

In view of the algebra developed in the previous section, a natural question is to determine if there is a composition formula like (5-6), but for polynomials which satisfy (11). The answer to this question is connected to the celebrated four-squares Euler identity written under the form

$$\widehat{A}^2+\widehat{B}^2+\widehat{C}^2+\widehat{D}^2=\left(\widehat{a}^2+\widehat{b}^2+\widehat{c}^2+\widehat{d}^2\right)\left(\widehat{\alpha}^2+\widehat{\beta}^2+\widehat{\gamma}^2+\widehat{\delta}^2\right) \tag{12}$$

where

$$\left\{\begin{array}{l}\widehat{A}=\widehat{a}\widehat{\alpha}+\widehat{b}\widehat{\beta}+\widehat{c}\widehat{\gamma}+\widehat{d}\widehat{\delta}\\ \widehat{B}=\widehat{a}\widehat{\beta}-\widehat{b}\widehat{\alpha}+\widehat{c}\widehat{\delta}-\widehat{d}\widehat{\gamma}\\ \widehat{C}=\widehat{a}\widehat{\gamma}-\widehat{b}\widehat{\delta}-\widehat{c}\widehat{\alpha}+\widehat{d}\widehat{\beta}\\ \widehat{D}=\widehat{a}\widehat{\delta}+\widehat{b}\widehat{\gamma}-\widehat{c}\widehat{\beta}-\widehat{d}\widehat{\alpha}.\end{array}\right. \tag{13}$$

We introduce the weights by setting

$$\widehat{a}=\sqrt{w_1}a,\quad \widehat{b}=\sqrt{w_2}b,\quad \widehat{c}=\sqrt{w_3}c,\quad \widehat{d}=\sqrt{w_4}d, \tag{14}$$

$$\widehat{\alpha}=\sqrt{w_1}\alpha,\quad \widehat{\beta}=\sqrt{w_2}\beta,\quad \widehat{\gamma}=\sqrt{w_3}\gamma,\quad \widehat{\delta}=\sqrt{w_4}\delta, \tag{15}$$

and

$$\widehat{A}=\sqrt{w_1}A,\quad \widehat{B}=\sqrt{w_2}B,\quad \widehat{C}=\sqrt{w_3}C,\quad \widehat{D}=\sqrt{w_4}D. \tag{16}$$

Let us start from (a,b,c,d) and $(\alpha,\beta,\gamma,\delta)$ which satisfy (11): use the chain (14-15), then (13), then (16) to get the final expressions of (A,B,C,D)

$$\left\{\begin{array}{l}A=\sqrt{w_1}a\alpha+\sqrt{\dfrac{w_2^2}{w_1}}b\beta+\sqrt{\dfrac{w_3^2}{w_1}}c\gamma+\sqrt{\dfrac{w_4^2}{w_1}}d\delta,\\ B=\sqrt{w_1}a\beta-\sqrt{w_1}b\alpha+\sqrt{\dfrac{w_3w_4}{w_2}}c\delta-\sqrt{\dfrac{w_3w_4}{w_2}}d\gamma,\\ C=\sqrt{w_1}a\gamma-\sqrt{\dfrac{w_2w_4}{w_3}}b\delta-\sqrt{w_1}c\alpha+\sqrt{\dfrac{w_2w_4}{w_3}}d\beta,\\ D=\sqrt{w_1}a\delta+\sqrt{\dfrac{w_2w_3}{w_4}}b\gamma-\sqrt{\dfrac{w_2w_3}{w_4}}c\beta-\sqrt{w_1}d\alpha.\end{array}\right. \tag{17}$$

It can be rewritten as

$$\left(\begin{array}{c}A\\B\\C\\D\end{array}\right)=M\left(\begin{array}{c}a\\b\\c\\d\end{array}\right) \tag{18}$$

where M is a 4×4 matrix. A fundamental property is that (A,B,C,D) satisfies (11) provided it holds already for (a,b,c,d) and $(\alpha,\beta,\gamma,\delta)$.

The issue is that (A,B,C,D) are not necessarily polynomials, since the square root of fractions of polynomial weights show up in (17). To investigate the constraints brought by the weights, we simplify M by keeping only the

weights. One obtains the 4×4 matrix of the weights

$$W = \begin{pmatrix} \sqrt{w_1} & \sqrt{\frac{w_2^2}{w_1}} & \sqrt{\frac{w_3^2}{w_1}} & \sqrt{\frac{w_4^2}{w_1}} \\ \sqrt{w_1} & \sqrt{w_1} & \sqrt{\frac{w_3 w_4}{w_2}} & \sqrt{\frac{w_3 w_4}{w_2}} \\ \sqrt{w_1} & \sqrt{\frac{w_2 w_4}{w_3}} & \sqrt{w_1} & \sqrt{\frac{w_2 w_4}{w_3}} \\ \sqrt{w_1} & \sqrt{\frac{w_2 w_3}{w_4}} b & \sqrt{\frac{w_2 w_3}{w_4}} & \sqrt{w_1} \end{pmatrix}. \tag{19}$$

In the general case the coefficients of this matrix cannot be expressed in polynomial form. Therefore the issue is to obtain good combinations of weights such that W has polynomial entries. We detail below two natural solutions.

2.2 Solution with the first representation formula (3)

One takes the first representation formula (3) for p and $1-p$. So $w_1 = w_3 = 1$ and $w_2 = w_4 = w$. In this case one gets a matrix of weights with polynomial entries

$$W = \begin{pmatrix} 1 & w & 1 & w \\ 1 & 1 & 1 & 1 \\ 1 & w & 1 & w \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

The system (17) can be recast as

$$\begin{cases} A & = a\alpha & +wb\beta & +c\gamma & +wd\delta, \\ B & = a\beta & -b\alpha & +c\delta & -d\gamma, \\ C & = a\gamma & -wb\delta & -c\alpha & +wd\beta, \\ D & = a\delta & +b\gamma & -c\beta & -d\alpha. \end{cases} \tag{20}$$

Lemma 2.2. *Assume* $(a,b,c,d) \in P_n \times P_{n-1} \times P_n \times P_{n-1}$ *and* $(\alpha,\beta,\gamma,\delta) \in P_m \times P_{m-1} \times P_m \times P_{m-1}$. *Then* $(A,B,C,D) \in P_{n+m} \times P_{n+m-1} \times P_{n+m} \times P_{n+m-1}$.

Proof. Obvious since the degree of the weight $w(x) = x(1-x)$ is equal to 2. □

Let us note

$$\mathcal{U}_n = \left\{ (a,b,c,d) \in P_n \times P_{n-1} \times P_n \times P_{n-1} \text{ such that } 1 = a^2 + b^2 w + c^2 + d^2 w \right\}.$$

By abuse of notation, any element in this set will be referred to as a polynomial in $\mathcal{U}_n$.

Lemma 2.3. *Take* $(\alpha,\beta,\gamma,\delta) = (a,b,c,d) \in \mathcal{U}_n$. *Then* $(A,B,C,D) = (1,0,0,0)$.

Proof. Obvious □

This lemma can be rephrased by saying that $\mathcal{U}_\infty$ is endowed with a group structure and that an element is its own inverse. The next natural question is to determine if one can generate polynomials in $\mathcal{U}_n$ by composition of polynomials with lesser degree, as in formula (7) in the proof of the Lukacs theorem. Let us first examine the simplest non trivial case which corresponds to $\mathcal{U}_1$.

Lemma 2.4. *The polynomials $(a,b,c,d) \in \mathcal{U}_1$ can be written as*

$$\left\{\begin{array}{ll} a(x) & = \cos\theta x + \cos\varphi(1-x), \\ b & = R\cos\mu, \\ c(x) & = \sin\theta x + \sin\varphi(1-x), \\ d & = R\sin\mu, \end{array}\right. \tag{21}$$

where the angles $(\theta,\varphi,\mu) \in \mathbb{R}^3$ are arbitrary and $R = 2\sin\left(\frac{\theta-\varphi}{2}\right)$.

Proof. Since $w(0) = 0$, one has that $a(0)^2 + c(0)^2 = 1$. That is $a(0) = \cos\theta$ and $c(0) = \sin\theta$ for some $\theta \in \mathbb{R}$. Similarly $w(1) = 0$ so one can write $a(1) = \cos\varphi$ and $c(0) = \sin\varphi$ for some $\varphi \in \mathbb{R}$. Since a and c are first order polynomials, one gets $a(x) = \cos\theta x + \cos\varphi(1-x)$ and $c(x) = \sin\theta x + \sin\varphi(1-x)$. A direct expansion yields that

$$\begin{aligned} 1 &= \left(a^2 + b^2 w + c^2 + d^2 w\right)(x) \\ &= \left(\cos\theta x + \cos\varphi(1-x)\right)^2 + \left(\sin\theta x + \sin\varphi(1-x)\right)^2 + (b^2+d^2)(x-x^2) \\ &= x^2 + (1-x)^2 + (b^2 + d^2 + 2\cos\theta\cos\varphi + 2\sin\theta\sin\varphi)(x-x^2) \\ &= 1 - 2(x-x^2) + (b^2 + d^2 + 2\cos(\theta-\varphi))(x-x^2) \\ &= 1 + (b^2 + d^2 + 2\cos(\theta-\varphi) - 2)(x-x^2). \end{aligned}$$

Therefore $b^2+d^2 = 2-2\cos(\theta-\varphi) = 4\sin^2\left(\frac{\theta-\varphi}{2}\right)$ from which the representation of b and d in the claim (21) is deduced. The proof is ended. □

Theorem 2.5. *Let $n \geq 1$. Any polynomial in $\mathcal{U}_n$ can be obtained with a repeated use of the formula (20) applied to at most n polynomials in $\mathcal{U}_1$.*

Proof. Consider (20) and assume that $(A,B,C,D) \in \mathcal{U}_n$ with $n \geq 2$ is given. We will construct $(\alpha,\beta,\gamma,\delta) \in \mathcal{U}_1$ and $(a,b,c,d) \in \mathcal{U}_{n-1}$ which all satisfy (20). It will prove the theorem by descending iteration on n. The proof proceeds in several elementary steps.

• A first remark is that (20) can be inverted for $(\alpha,\beta,\gamma,\delta) \in \mathcal{U}_1$ since

$$\begin{pmatrix} \alpha & \beta w & \gamma & \delta w \\ \beta & -\alpha & -\delta & \gamma \\ \gamma & w\delta & -\alpha & -w\beta \\ \delta & -\gamma & \beta & -\alpha \end{pmatrix} \begin{pmatrix} \alpha & \beta w & \gamma & \delta w \\ \beta & -\alpha & \delta & -\gamma \\ \gamma & -w\delta & -\alpha & w\beta \\ \delta & \gamma & -\beta & -\alpha \end{pmatrix} = I_4$$

with I_4 the identity matrix. One finds that (20) is equivalent to

$$\left\{\begin{array}{l} a = A\alpha + wB\beta + C\gamma + wD\delta, \\ b = A\beta - B\alpha - c\delta + d\gamma, \\ c = A\gamma + wb\delta - C\alpha - wd\beta, \\ d = A\delta - b\gamma + c\beta - D\alpha. \end{array}\right. \tag{22}$$

So the cornerstone of the proof amounts to showing that there exists $(\alpha, \beta, \gamma, \delta) \in \mathcal{U}_1$ such that $(a, b, c, d) \in \mathcal{U}_{n-1}$.

• For $n \geq 2$, denote the dominant coefficients of the polynomials as

$$\left\{\begin{array}{lll} A(x) & = A_n x^n & +A_{n-1}x^{n-1} \quad +\ldots \\ B(x) & = & B_{n-1}x^{n-1} \quad +\ldots \\ C(x) & = C_n x^n & +C_{n-1}x^{n-1} \quad +\ldots \\ D(x) & = & D_{n-1}x^{n-1} \quad +\ldots \end{array}\right.$$

Since $1 = A^2 + B^2 w + C^2 + D^2 w$ and $w(x) = x - x^2$, the dominant coefficients at order $2n$ and $2n-1$ satisfy by identification

$$A_n^2 - B_{n-1}^2 + C_n^2 - D_{n-1}^2 = 0 \text{ and } 2A_nA_{n-1} + B_{n-1}^2 + 2C_nC_{n-1} + D_{n-1}^2 = 0. \quad (23)$$

If $A_n^2 + C_n^2 = 0$, then $B_{n-1} = D_{n-1} = 0$. So $A, C \in P_{n-1}$ and $B, D \in P_{n-2}$. In this case one can take immediately $(a, b, c, d) = (A, B, C, D)$ and there is nothing to prove. So we consider below the main case where $A_n^2 + C_n^2 > 0$.

• Define

$$\left\{\begin{array}{ll} \alpha(x) & = \varepsilon(A_n x + A_{n-1}x), \\ \beta & = \varepsilon B_{n-1}, \\ \gamma(x) & = \varepsilon(C_n x + C_{n-1}x), \\ \delta & = \varepsilon D_{n-1} \end{array}\right.$$

where $\varepsilon > 0$ is a scaling parameter to be chosen. A direct expansion yields

$$\alpha(x)^2 + \beta^2 w + \gamma(x)^2 + \delta^2 = \varepsilon^2 \left[\left(A_n^2 - B_{n-1}^2 + C_n^2 - D_{n-1}^2\right) x^2\right.$$

$$\left. + \left(2A_nA_{n-1} + B_{n-1}^2 + 2C_nC_{n-1} + D_{n-1}^2\right) x + \left(A_{n-1}^2 + C_{n-1}^2\right)\right]$$

that is

$$\alpha(x)^2 + \beta^2 w + \gamma(x)^2 + \delta^2 = \varepsilon^2 \left(A_{n-1}^2 + C_{n-1}^2\right).$$

If $A_{n-1}^2 + C_{n-1}^2 = 0$ then (23) shows that $0 = B_{n-1}^2 + D_{n-1}^2 = A_n^2 + C_n^2$ which is excluded at this stage of the discussion. So one can take $\varepsilon = \left(A_{n-1}^2 + C_{n-1}^2\right)^{-\frac{1}{2}} > 0$. It insures that $(\alpha, \beta, \gamma, \delta) \in \mathcal{U}_1$.

• Let us now determine the dominant coefficients of a and c given by (22). An expansion yields

$$a(x) = \varepsilon \left(A_n^2 - B_{n-1}^2 + C_n^2 - D_{n-1}^2\right) x^{n+1}$$

$$+\varepsilon \left(2A_nA_{n-1} + B_{n-1}^2 + 2C_nC_{n-1} + D_{n-1}^2\right) x^n + \text{ low order terms}$$

and

$$c(x) = \varepsilon \left(A_nC_n - B_{n-1}D_{n-1} - C_nA_n + D_{n-1}B_{n-1}\right) x^{n+1}$$

$$+\varepsilon \left(A_nC_{n-1} + A_{n-1}C_n + B_{n-1}D_{n-1} - C_nA_{n-1} - C_{n-1}A_n - D_{n-1}B_{n-1}\right) x^n$$

$$+ \text{ low order terms.}$$

Since these four dominant terms vanish, $a, c \in P_{n-1}$.

• By construction $a^2 + b^2 w + c^2 + d^2 w = 1$, that is

$$(b^2 + d^2)w = 1 - a^2 - c^2 \in P_{2n-2}.$$

Since $w(x) = x - x^2$, the dominant terms of b and d have maximal degree $n-2$. That is $b, d \in P_{n-2}$.
• It proves that one can determine $(\alpha, \beta, \gamma, \delta) \in \mathcal{U}_1$ so that $(a, b, c, d) \in \mathcal{U}_{n-1}$ is one degree less than $(A, B, C, D) \in \mathcal{U}_n$. By descending iteration it ends the proof. □

Corollary 2.6. *There exists a parametrization of U_{2n} with $3n$ real coefficients.*

Proof. Indeed $\mathcal{U}_1$ is parametrized with 3 angles. By composition $\mathcal{U}_n$ is parametrized with $3n$ angles. So U_{2n} which consists of $p = a^2 + b^2 w$ with $(a, b, c, d) \in \mathcal{U}_n$ can also be parametrized with $3n$ angles. The proof is ended. □

2.3 A solution for the second representation formula (4)

The extension to the second representation formula appears to be trickier and probably less efficient for practical computations since the parametrization of U_{2n+1} needs more parameters than $\mathcal{U}_{2n+2}$. The algebra is as follows.

The weights are $w_1(x) = w_3(x) = x$ and $w_2(x) = w_4(x) = 1 - x$ and the matrix of weights recasts as

$$W = \begin{pmatrix} \sqrt{w_1} & \sqrt{\frac{w_2^2}{w_1}} & \sqrt{w_1} & \sqrt{\frac{w_2^2}{w_1}} \\ \sqrt{w_1} & \sqrt{w_1} & \sqrt{w_1} & \sqrt{w_1} \\ \sqrt{w_1} & \sqrt{\frac{w_2^2}{w_1}} & \sqrt{w_1} & \sqrt{\frac{w_2^2}{w_1}} \\ \sqrt{w_1} & \sqrt{w_1} & \sqrt{w_1} & \sqrt{w_1} \end{pmatrix}.$$

The entries are not polynomials because $\sqrt{w_1} = \sqrt{x}$, so it is not possible to use directly the system (20). Nevertheless one observes that by taking the square

$$W^2 = \begin{pmatrix} z & 4w_2 & z & 4w_2 \\ 4w_1 & z & 4w_1 & z \\ z & 4w_2 & z & 4w_2 \\ 4w_1 & z & 4w_1 & z \end{pmatrix}, \quad z = 2w_1 + 2w_2, \tag{24}$$

which has now polynomials entries. It indicates that two successive uses of the weighted-four-squares Euler identity generate polynomials.

Let us denote

$$\mathcal{V}_n = \left\{ (a, b, c, d) \in P_n^4 \text{ such that } 1 = a^2 w_1 + b^2 w_2 + c^2 w_1 + d^2 w_2 \right\}. \tag{25}$$

Lemma 2.7. *Assume $(a, b, c, d) \in \mathcal{V}_n$, $(\alpha_1, \beta_1, \gamma_1, \delta_1) \in \mathcal{V}_m$ and $(\alpha_2, \beta_2, \gamma_2, \delta_2) \in \mathcal{V}_p$. Then two successive uses of (17) yield $(A, B, C, D) \in \mathcal{V}_{n+m+p+1}$.*

Proof. The algebra can be rewritten as in (18) where the entries of the matrix

$M = (m_{ij})_{1\leq i,j\leq 4}$ are given in functions of $(\alpha_1, \beta_1, \gamma_1, \delta_1)$ and $(\alpha_2, \beta_2, \gamma_2, \delta_2)$

$$\left\{\begin{array}{ll} m_{11} & = (\alpha_1\alpha_2 + \gamma_1\gamma_2)w_1 + (\beta_1\beta_2 + \delta_1\delta_2)w_2, \\ m_{22} & = (\alpha_1\alpha_2 - \gamma_1\gamma_2)w_1 + (\beta_1\beta_2 - \delta_1\delta_2)w_2, \\ m_{33} & = (\alpha_1\alpha_2 + \gamma_1\gamma_2)w_1 - (\beta_1\beta_2 + \delta_1\delta_2)w_2, \\ m_{44} & = (\alpha_1\alpha_2 - \gamma_1\gamma_2)w_1 - (\beta_1\beta_2 - \delta_1\delta_2)w_2, \\ m_{12} & = (\alpha_1\beta_2 - \beta_1\alpha_2 - \gamma_1\delta_2 + \delta_1\gamma_2)w_2, \\ m_{13} & = (\alpha_1\gamma_2 - \gamma_1\alpha_2)w_1 + (\beta_1\delta_2 + \delta_1\beta_2)w_2, \\ m_{14} & = (\alpha_1\delta_2 + \beta_1\gamma_2 + \gamma_1\beta_2 + \delta_1\alpha_2)w_2, \\ m_{21} & = (\beta_1\alpha_2 - \alpha_1\beta_2 + \delta_1\gamma_2 - \gamma_1\delta_2)w_1, \\ m_{23} & = (\beta_1\gamma_2 - \alpha_1\delta_2 - \delta_1\alpha_2 + \gamma_1\beta_2)w_1, \\ m_{24} & = (\alpha_1\gamma_2 + \gamma_1\alpha_2)w_1 + (\beta_1\delta_2 + \delta_1\beta_2)w_2, \\ m_{31} & = (\gamma_1\alpha_2 - \alpha_1\gamma_2)w_1 + (-\delta_1\beta_2 + \beta_1\delta_2)w_2, \\ m_{32} & = (\gamma_1\beta_2 + \delta_1\alpha_2 + \alpha_1\delta_2 + \beta_1\gamma_2)w_2, \\ m_{34} & = (\gamma_1\delta_2 + \delta_1\gamma_2 - \alpha_1\beta_2 - \beta_1\alpha_2)w_2, \\ m_{41} & = (\delta_1\alpha_2 + \gamma_1\beta_2 - \beta_1\gamma_2 - \alpha_1\delta_2)w_1, \\ m_{42} & = -(\gamma_1\alpha_2 + \alpha_2\gamma_1)w_1 + (\delta_1\beta_2 + \beta_1\delta_2)w_2, \\ m_{41} & = (\delta_1\gamma_2 + \gamma_1\delta_2 + \beta_1\alpha_2 + \alpha_1\beta_2)w_1. \end{array}\right. \tag{26}$$

It yields the claim since all terms have degree $m+p+1$. The proof is ended. □

This algebra does not have a clear classical group structure and is less elegant than the previous one. No similar algebra has been found so far in the literature, even for example in [5] and therein. Nevertheless it can be used to generate polynomials with higher and higher degree by iterations with $(\alpha_1, \beta_1, \gamma_1, \delta_1) \in \mathcal{V}_0$ and $(\alpha_2, \beta_2, \gamma_2, \delta_2) \in \mathcal{V}_0$.

Lemma 2.8. *The polynomials $(a, b, c, d) \in \mathcal{V}_0$ can be written as*

$$a = \cos\theta, \quad b = \cos\varphi, \quad c = \sin\theta, \quad d = \sin\varphi \tag{27}$$

where the real angles θ and φ are arbitrary.

Proof. Obvious. □

The numerical tests show without ambiguity that the composition (18)-(26) with $(\alpha_1, \beta_1, \gamma_1, \delta_1) \in \mathcal{V}_0$ and $(\alpha_2, \beta_2, \gamma_2, \delta_2) \in \mathcal{V}_0$ generates all possible polynomials in $\mathcal{V}_n$. We remark that 4 parameters (that is four angles) are needed to increase the degree by 1. Therefore this parametrization is less efficient than the previous one which needs only 3 parameters to increase the degree by 1 and will not be discussed further.

2.4 Accuracy of the approximation

The next result is a fundamental inequality for best approximation of a real valued continuous function by polynomial in U_n. The norm of a uniform convergence is noted

$$\|f\| = \max_{0\leq x\leq 1} |f(x)|, \quad f \in C^0[0, 1].$$

Theorem 2.9. *Assume $f \in C^0[0,1]$ and $0 \leq f(x) \leq 1$ for $0 \leq x \leq 1$. Then*

$$\inf_{p_n \in U_n} \|f - p_n\| \leq 2 \inf_{g_n \in P_n} \|f - g_n\|. \tag{28}$$

Proof. Any polynomial $g_n \in P_n$ satisfies

$$\|g_n - 1/2\| \leq \|f - 1/2\| + \|f - g_n\| \leq 1/2 + \|f - g_n\|.$$

Define

$$p_n = 1/2 + \frac{1}{1 + 2\|f - g_n\|} (g_n - 1/2).$$

By construction $\|p_n - 1/2\| \leq \frac{1}{1+2\|f-g_n\|} \|g_n - 1/2\| \leq 1/2$, so $p_n \in U_n$. One has the identity

$$f - p_n = f - \frac{1}{2} - \frac{1}{1 + 2\|f - g_n\|} (g_n - 1/2)$$

$$= \left(1 - \frac{1}{1 + 2\|f - g_n\|}\right) (f - 1/2) + \frac{1}{1 + 2\|f - g_n\|} (f - g_n)$$

$$= \frac{2\|f - g_n\|}{1 + 2\|f - g_n\|} (f - 1/2) + \frac{1}{1 + 2\|f - g_n\|} (f - g_n)$$

from which one gets the triangular inequality

$$\|f - p_n\| \leq \|f - g_n\| + \|f - g_n\| = 2\|f - g_n\|.$$

Taking the infimum ends the proof. □

So, for a function $0 \leq f \leq 1$, the best approximation in the uniform norm with polynomial in U_n has the same asymptotic accuracy than the best approximation with polynomials in P_n. Various bounds can be derived by characterizing the convergence of $\inf_{g_n \in P_n} \|f - g_n\|$ with respect to n. For example spectral convergence $O(n^{-m})$ for all m is achieved if $f \in C^\infty[0,1]$. For all $1 \leq p \leq \infty$, standard interpolation and/or regularization techniques [8] show the convergence in $L^p(0,1)$ for $f \in L^p(0,1)$ and $0 \leq f \leq 1$.

3 Higher dimensions

It is well known that the characterization of polynomials of many variables which are non negative over a compact set in $\mathbb{R}^d$ poses fundamental problems, most of them linked to the 17th Hilbert problem [16, 1]. In relation with applications, we refer to the recent and comprehensive textbook [13]. We also note that many different representations of non negative polynomials exist, but none of them seems definitely superior to the others for applications.

That is why we restrict the presentation to the main features of the generating formulas over the academic square $\mathcal{C} = [0,1]^d$. In the context of this work generating formulas are related to augmented versions of a weighted four-squares Euler relation but the Hurwitz theorem brings an important restriction.

3.1 Hurwitz theorem and Degen identity

One could ask about generalization of the classical four-squares Euler relation to n-squares identities. Such formulas exist but only for $n = 1, 2, 4, 8$: this is the Hurwitz theorem. For $n = 8$, the solution is given in the form of the Degen identity

$$(a^2+b^2+c^2+d^2+e^2+f^2+g^2+h^2)(m^2+n^2+o^2+p^2+q^2+r^2+s^2+t^2)$$

$$= (am-bn-co-dp-eq-fr-gs-ht)^2+(bm+an+do-cp+fq-er-hs+gt)^2$$

$$+(cm-dn+ao+bp+gq+hr-es-ft)^2+(dm+cn-bo+ap+hq-gr+fs-et)^2$$

$$+(em-fn-go-hp+aq+br+cs+dt)^2+(fm+en-ho+gp-bq+ar-ds+ct)^2$$

$$+(gm+hn+eo-fp-cq+dr+as-bt)^2+(hm-gn+fo+ep-dq-cr+bs+at)^2.$$

It is convenient to rewrite it as an Euler identity with complex numbers. We define ($i^2 = -1$)

$$u = a + ib, \quad v = c + id, \quad w = e + if, \quad z = g + ih \tag{29}$$

and

$$\alpha = m + in, \quad \beta = o + ip, \quad \gamma = q + ir, \quad \delta = s + it. \tag{30}$$

The Degen identity rewrites as

$$|A|^2+|B|^2+|C|^2+|D|^2 = \left(|u|^2+|v|^2+|w|^2+|z|^2\right)\left(|\alpha|^2+|\beta|^2+|\gamma|^2+|\delta|^2\right) \tag{31}$$

with

$$\begin{cases} A = u\alpha - v^*\beta - w\gamma^* - z^*\delta, \\ B = v\alpha + u^*\beta + z\gamma^* - w^*\delta, \\ C = w\alpha - z\beta^* + u^*\gamma + v^*\delta, \\ D = z\alpha^* + w\beta - v\gamma + u\delta. \end{cases} \tag{32}$$

3.2 Polynomials with bounds

In order to generate non negative polynomials, one can start from the possible representation (this is arbitrary)

$$p(x,y) = a(x,y)^2 w_1(x) + c(x,y)^2 w_2(x) + e(x,y)^2 w_3(y) + g(x,y)^2 w_4(x) \tag{33}$$

with $w_1(x) = x$, $w_2(x) = 1 - x$, $w_3(y) = y$ and $w_4(y) = 1 - y$, and a, c, e, g polynomials with respect to x and y. By construction $p(x, y) \geq 0$ for all $0 \leq x, y \leq 1$. Many other representations are possible, see [13].

Let us impose that $p(x, y) \leq 1$ for all $0 \leq x, y \leq 1$ by writing (this is still arbitrary)

$$1-p(x,y) = b(x,y)^2 w_1(x)+d(x,y)^2 w_2(x)+f(x,y)^2 w_3(y)+h(x,y)^2 w_4(x). \tag{34}$$

By summation one obtains a weighted 8-squares relation

$$1 = \left(a(x,y)^2 + b(x,y)^2\right) w_1(x) + \left(c(x,y)^2 + d(x,y)^2\right) w_2(x)$$

$$+\left(e(x,y)^2+f(x,y)^2\right)w_3(y)+\left(g(x,y)^2+h(x,y)^2\right)w_4(x).$$

Define (u,v,w,z) as in (29). One obtains

$$1=|u|^2w_1+|v|w_2+|w|^2w_3+|z|^2w_4.$$

Using the approach used successfully in dimension $d=1$, we modify the Degen identity with the weights $w_{1,2,3,4}$ which are now all different. The algebraic similarity of (13) with (32) shows that, after introducing the weights, one obtains a form similar to (17) but with complex numbers. The matrix of weights W obtained after simplification is still (19). It means that we have to analyze W with four different weights. In general the entries of W cannot be polynomials. A similar situation has been encountered in dimension $d=1$ with the second representation formula and it was sufficient to take the square of W to generate a matrix with polynomial entries, see (24). In the new situation one finds that the two first coefficients of $W^2=(h_{ij})_{1\leq i,j\leq 4}$ are

$$h_{11}=w_1+w_2+w_3+w_4,\quad h_{12}=2w_2+2\sqrt{\frac{w_2w_3w_4}{w_1}},\quad \dots$$

Therefore even taking the square does not yield a polynomial matrix for general weights, because the structure of the weights brings rigidity to the method. A convenient solution is nevertheless the following.

Lemma 3.1. *Consider the weights $w_1=1$, $w_2=1-x^2$, $w_3=1-y^2$ and a redundant fourth weight $w_4=w_2w_3$. Then W has polynomial entries.*

Proof. Indeed a direct calculation shows that $W=\begin{pmatrix}1&w_2&w_3&w_2w_3\\1&1&w_3&w_3\\1&w_2&1&w_2\\1&1&1&1\end{pmatrix}$. □

With these weights one modifies (32) which becomes

$$\left\{\begin{array}{rcccccl}A&=&u\alpha&-v^*\beta w_2&-w\gamma^*w_3&-z^*\delta w_2w_3,\\B&=&v\alpha&+u^*\beta&+z\gamma^*w_3&-w^*\delta w_3,\\C&=&w\alpha&-z\beta^*w_2&+u^*\gamma&+v^*\delta w_2,\\D&=&z\alpha^*&+w\beta&-v\gamma&+u\delta.\end{array}\right.\tag{35}$$

The solutions of this system are endowed with a weighted 8-squares identity written as a weighted complex-4 squares identity

$$|A|^2+|B|^2w_2+|C|^2w_3+|D|^2w_2w_3\tag{36}$$

$$=\left(|u|^2+|v|^2w_2+|w|^2w_3+|z|^2w_2w_3\right)\left(|\alpha|^2+|\beta|^2w_2+|\gamma|^2w_3+|\delta|^2w_2w_3\right).$$

It shows that (35) preserves identities like

$$1=|\alpha|^2+|\beta|^2w_2+|\gamma|^2w_3+|\delta|^2w_2w_3\tag{37}$$

which models polynomials with bounds on the square. Let us note

$$\mathcal{U}_n = \left\{(\alpha,\beta,\gamma,\delta) \in P_n(x,y) \times P_{n-1}(x,y)^2 \times P_{n-2}(x,y) \text{ such that (37) holds}\right\}$$

and

$$U_{2n} = \{p \in P_{2n}(x,y) : \exists (\alpha,\beta,\gamma,\delta) \in \mathcal{U}_n$$
$$\text{such that } p = \text{Real}(\alpha)^2 + \text{Real}(\beta)^2 w_2 + \text{Real}(\gamma)^2 w_3 + \text{Real}(\delta)^2 w_2 w_3\}\,.$$

Remark 3.2 (Links with Quaternions algebras)**.** *In a different context [5] and in the language of quaternions $i^2 = j^2 = k^2 = ijk = -1$, this structure can be reformulated as a quaternions algebra with the quaternion basis $(1, w_2 i, w_3 j, w_2 w_3 k)$ over the field of polynomials. It is is worthwhile to note that (20) can also be rewritten in terms of the quaternion basis $(1, wi, j, wk)$. On the contrary, the algebra (24)-(27) does not have a clear formulation in terms of a quaternion algebra. In the two first case, the polynomials in U_n are also called unit quaternions.*

The next step in the construction is the finding of complex valued polynomials $(\alpha,\beta,\gamma,\delta)$ with minimal degree which satisfy (37). This problem shows issues in order to decide of a convenient parametrization, a necessary task in view of efficient implementation. This is where a even greater simplification can be introduced in the search of such polynomials, adding the requirement that the polynomials are independent either with respect to x or with respect to y.

Lemma 3.3. *Elementary solutions of (37) with low degree are*

$$\text{either } \alpha = e^{i\theta}x + e^{i\varphi}(1-x), \quad \beta = Re^{i\mu}, \quad \gamma = \delta = 0, \tag{38}$$

$$\text{or } \alpha = e^{i\theta}y + e^{i\varphi}(1-y), \quad \gamma = Re^{i\mu}, \quad \beta = \delta = 0, \tag{39}$$

where the angles $\theta, \varphi, \mu \in \mathbb{R}$ are arbitrary and $R = 2\sin\left(\frac{\theta-\varphi}{2}\right)$.

Proof. Obvious from (21). □

3.3 Tensorization in higher dimensions

The solution developed so far in (37-38-39) can be reinterpreted as a tensorization procedure. It immediately yields a formal theoretical extension of the generation of polynomials with bounds on the hypercube $\mathcal{C} = [0,1]^d \subset \mathbb{R}^d$ in any dimension. We restrict hereafter the presentation to the main idea.

Let

$$\mathbf{j} = (j_1, \ldots, j_d) \in \mathcal{J}_d \equiv \{0,1\}^d$$

be a multi-integer which is one corner of the hypercube. Its length is $|\mathbf{j}| = \sum_{k=1}^d j_k \leq d$. We consider complex valued polynomials $\alpha_{\mathbf{j}}$ (indexed by $\mathbf{j}$) in the variables $\mathbf{x} = (x_1, \ldots, x_d)$. We take the weights $w_k = x_k - x_k^2$ for $1 \leq k \leq d$. The notation $w^{\mathbf{j}}$ stems for $w^{\mathbf{j}} = \Pi_{k=1}^d w_k^{j_k}$. We consider the relation

$$\sum_{\mathbf{j}\in\mathcal{J}_d} w^{\mathbf{j}} |\alpha_{\mathbf{j}}(x)|^2 = 1 \tag{40}$$

which is a generalization of (37).

Lemma 3.4. *One has the rearrangement*

$$\sum_{\mathbf{j}\in\mathcal{J}_d} w^{\mathbf{j}}|\alpha_j(x)|^2 = \sum_{\mathbf{j}'\in\mathcal{J}_{d-1}} w^{(0,\mathbf{j}')}\left(|\alpha_{(0,\mathbf{j}')}|^2 + w_1|\alpha_{(1,\mathbf{j}')}|^2\right). \tag{41}$$

Proof. Obvious. □

Similar rearrangements can be written in any direction after a preliminary permutation of the directions. It shows that a simple four square identity can be used to generate sequences with preserve the value of $|\alpha_{(0,\mathbf{j}')}|^2 + w_1|\alpha_{(1,\mathbf{j}')}|^2$ for all $\mathbf{j}'$ provided one uses polynomials like (38) which leave invariant by composition the value of $|\alpha_{(0,\mathbf{j}')}|^2 + w_1|\alpha_{(1,\mathbf{j}')}|^2$. Therefore these sequences preserve $\sum_{\mathbf{j}\in\mathcal{J}_d} w^{\mathbf{j}}|\alpha_j(x)|^2$. The evaluation of this structure for practical calculations on a computer goes far beyond the scope of this work, and so is left for future research.

4 Application to numerical approximation

We use the simplicity of the parametrization offered by the different formulas discussed previously to generate polynomials with bounds and to compute a numerical solution to various problems of interest formulated as minimization problems.

4.1 Implementation issues

We firstly discuss some implementation issues. The tests have been performed within a Matlab test code, mainly with the procedure which corresponds to theorem 2.5. Implementation is as easy with the second representation procedure (see results in the right part of figure 3) and in dimension $d = 2$ (results in section 4.4).

4.1.1 Exact algebra

In Matlab, it is possible in dimension $d = 1$ to use exact manipulations of polynomials which are treated as arrays. Some results where the parameters $\alpha(k), \varphi(k), \mu(k)$ are randomized are given in figure 1. Note that the randomization of the parameters brings a destructive phenomenon which lessens the effective degree of the polynomials. The respect of the bounds is perfect for the first three results.

However an instability is visible for the last one. This is systematically observed for high degree and is due to a conditioning issue with high order terms. Indeed the function x^n tends to zero almost everywhere when n tends to infinity. At the same time the coefficient may tend to infinity.

An example is the rescaled Tchebycheff polynomial

$$\widehat{T}_n(x) = \frac{\cos(n\arccos(x)) + 1}{2} = 2^{n-1}x^n + \text{low order terms.}$$

It is clear that the accuracy of the numerical computation of $2^{n-1}x^n$ is weak for large n.

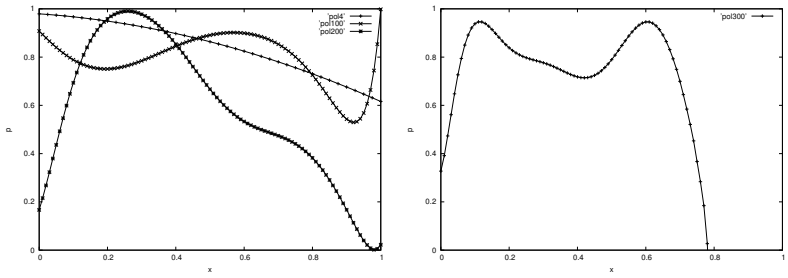

Figure 1: Results with exact polynomial calculations with degree $n = 2, 100, 200$ on the left figure and $n = 300$ on the right figure. The last result shows an instability: the lower bound is not respect due to bad conditioning.

4.1.2 Evaluation at given points

The other procedure for the numerical evaluation of polynomials with bounds uses the fact that it can be coded at quadrature points with isometries.

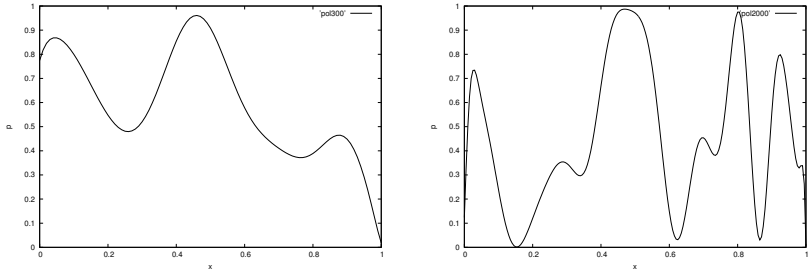

Figure 2: Polynomials computed at equi-distributed quadrature points with $n = 300$ and $n = 1000$.

Define the diagonal matrix

$$D(x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & x - x^2 & 0 \\ 0 & 0 & 0 & x - x^2 \end{pmatrix}$$

which is non negative for $0 \leq x \leq 1$. Take $\alpha = x\cos\theta + (1-x)\cos\varphi$, $\beta = R\cos\mu$,

$\gamma = x\sin\theta + (1-x)\sin\varphi$, $\delta = R\sin\mu$ with $R = 2\sin\left(\frac{\theta-\varphi}{2}\right)$. Define

$$R_k(x) = R(x;\theta_k,\varphi_k,\mu_k) \text{ with } R(x;\theta,\varphi,\mu) = \begin{pmatrix} \alpha & \beta & \gamma & \delta \\ \beta & -\alpha & \delta & -\gamma \\ \gamma & -\delta & -\alpha & \beta \\ \delta & \gamma & -\beta & -\alpha \end{pmatrix}.$$

Denote $\theta_k, \varphi_k, \mu_k$ the parameters at step k for $1 \le k \le n$. The loop with yields the numerical value of $(a,b,c,d)(x)$ at step N rewrites

$$\begin{pmatrix} a(x) \\ b(x) \\ c(x) \\ d(x) \end{pmatrix} = R_n(x)R_{n-1}(x)\dots R_2(x)R_1(x)X_0, \quad X_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{42}$$

The numerical value computed at the end of the loop writes

$$p(x) = a(x)^2 + c(x)^2(x-x^2), \quad p \in U_{2n}. \tag{43}$$

The stability of the loop (42) is insured by the property

$$R_k(x)^t D(x) R_k(x) = D(x) \tag{44}$$

which means that $D^{\frac{1}{2}}(x)R_k(x)D^{-\frac{1}{2}}(x)$ is an isometry for $0 < x < 1$. Two results are displayed in figure 2 with a polynomial degree $2n = 300$ (which was unstable with the previous implementation) and with a total degree equal to $2n = 2000$. Stability is now achieved unconditionally with respect to n.

4.2 Calculation of the exact gradients

It is of interest to develop a numerical procedure for the exact calculation of the gradient of $p(x)$ with respect of the parameters $(\theta_r, \varphi_r, \mu_r)_{1\le r\le n}$. Such methods provide an exact gradient which accelerates gradient algorithms for finding the minimum of certain differentiable cost functions. To explain the procedure we consider the calculation of $\partial_{\alpha_r} p(x)$ where $1 \le r \le n$ is arbitrary. The procedure is the same for $\partial_{\varphi_r} p(x)$ and $\partial_{\mu_r} p(x)$.

One has

$$\partial_{\alpha_r} p(x) = \partial_{\alpha_r} a(x)\,(2a(x)) + \partial_{\alpha_r} c(x)\left(2c(x)(x-x^2)\right)$$

where

$$\partial_{\alpha_r}\begin{pmatrix} a(x) \\ b(x) \\ c(x) \\ d(x) \end{pmatrix} = R_n(x)\dots R_{r+1}(x)\partial_{\alpha_r}R_r(x)R_{r-1}(x)\dots R_1(x)X_0.$$

An efficient procedure for the calculation of derivatives for all r is as follows.

Firstly one computes an adjoint vector

$$Y_{\rm ad} = \begin{pmatrix} a_{\rm ad}(x) \\ b_{\rm ad}(x) \\ c_{\rm ad}(x) \\ d_{\rm ad}(x) \end{pmatrix} = R_1(x)^{-t} R_2(x)^{-t} \dots R_{n-1}(x)^{-t} R_n(x)^{-t} \begin{pmatrix} 2a(x) \\ 0 \\ 2c(x)(x-x^2) \\ 0 \end{pmatrix} \tag{45}$$

where $R_k(x)^{-t} = D(x)R_k(x)D(x)^{-1}$ as a consequence of (44). Let $\langle , \rangle$ denotes the scalar euclidian product in $\mathbb{R}^4$.

Lemma 4.1. *One has the formula*

$$\partial_{\alpha_r} p(x) = \langle \partial_{\alpha_r} R_k(x) X_{r-1}, Y_r \rangle \tag{46}$$

where

$$X_{r-1} = R_{r-1}(x) \dots R_1(x) X_0 \text{ and } Y_r = R_r(x)^t \dots R_1(x)^t Y_{\rm ad}. \tag{47}$$

Remark 4.2. *The interest of the formula (46) is that the partial derivatives are computed for all $1 \leq r \leq n$ with a cost approximatively 3 times the cost of the evaluation of the polynomial p itself. This counting corresponds to the three loops described in (46)-(47). This extra-cost is therefore independent of n.*

Proof. By definition of the adjoint vector, one has

$$\begin{aligned} Y_r &= R_r(x)^t \dots R_1(x)^t R_1(x)^{-t} \dots R_n(x)^{-t} \begin{pmatrix} 2a(x) \\ 0 \\ 2c(x)(x-x^2) \\ 0 \end{pmatrix} \\ &= R_{r+1}(x)^{-t} \dots R_n(x)^{-t} \begin{pmatrix} 2a(x) \\ 0 \\ 2c(x)(x-x^2) \\ 0 \end{pmatrix}. \end{aligned}$$

Plug in (46). It yields

$$\langle \partial_{\alpha_r} R_k(x) X_{r-1}, Y_r \rangle = \langle \partial_{\alpha_r} \dots R_n(x) \dots R_{r+1}(x) R_k(x) R_{r-1}(x) \dots R_1(x) X_0, Y_{\rm ad} \rangle .$$

The proof is ended. □

A similar technique can be used for the exact computation of the derivative of $p'(x)$.

Lemma 4.3. *One has the formula*

$$p'(x) = c^2(x)(1-2x) + \sum_{r=1}^{n} \langle \partial_x R_k(x) X_{r-1}, Y_r \rangle . \tag{48}$$

Proof. Obvious from the proof of the previous lemma. □

4.3 Minimization of functionals

All problems considered below are written like

$$\text{Find } p_n \in U_n \text{ such that } J(p_n) \leq J(q_n) \quad \forall q_n \in U_n$$

where J is some functional. For example it can be the L^2 norm between p_n and a given objective function f: in this case $J(q_n) = \left(\int_0^1 |f(x) - q_n(x)|^2 dx\right)^{\frac{1}{2}}$. We also use the L^1 norm, typically for a discontinuous objective function. The L^∞ norm is less appealing, because it might slow down the rate of convergence of algorithms. Such problems correspond to the calculation of a best approximation of f by polynomials in U_n.

The initial problem is discretized with quadrature points x_i and with $\alpha \in \mathbb{R}^{3n}$ which is the vector that contains all the angles for the generation of the polynomials with bounds. The result is written as $q_n(x_i; \alpha)$. The functional $J(q_n)$ is in practice discretized as

$$\mathcal{J}_h(\alpha) = \sum_i \omega_i J(q_n(x_i; \alpha)), \quad \alpha \in \mathbb{R}^{3n}.$$

Matlab is used for the numerical calculations of minimizers

$$\mathcal{J}_h(\alpha_*) \leq \mathcal{J}_h(\alpha) \quad \forall \alpha \in \mathbb{R}^{3n}.$$

An important feature is that even if J is convex, $\mathcal{J}_h$ may be non convex because the parametrization $\alpha \mapsto p(\alpha)$ is non convex. Moreover also that $p(\alpha)$ and $\mathcal{J}_h(\alpha)$ are 2π periodic for all variables

$$\mathcal{J}_h(\alpha) = \mathcal{J}_h(\alpha + 2\pi\mathbf{m}) \quad \forall \mathbf{m} \in \mathbb{Z}^{3n}.$$

We rely on the Matlab function `fminunc` to determine the minimum of $\mathcal{J}_h$. Actually $\mathcal{J}_h$ may have local minima α_1, α_2, ...: we systematically run the calculation between 1 and 5 times and keep the best candidate. It is possible to run `fminunc` with exact calculation of the gradient of the functional by either using the procedure described in lemma 4.1 or asking `fminunc` to compute an approximation of the gradients by itself. The CPU time is not reported because the implementation is not optimal and it is related to the Matlab function `fminunc`, whom complexity and rate of convergence change if one provides the exact gradient or not. It is sufficient to know that the CPU time is, depending on the problem, between 5s and 2 minutes on a MacBookAir. Of course a 2D problem with a lot of quadrature points increases a lot this CPU time.
Finally once a discrete minimizer is determined, the result is plot on a grid with a number of visualization points which may be larger than the number of quadrature points.

4.3.1 Minimization of L^p-based functionals

We first consider the L^2 norm between $p(\alpha)$ and the Runge function properly rescaled in the bounds $[0,1]$ as

$$f_1(x) = \frac{26}{25}\left(\frac{1}{1+25(2x-1)^2} - \frac{2}{6}\right).$$

Three computations are performed with $n = 2, 4, 6$, $\alpha \in \mathbb{R}^{3n}$ and $p \in P_{2n}$. For

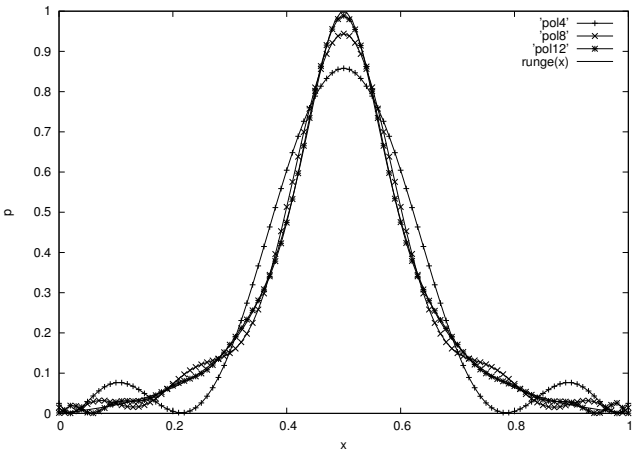

Figure 3: Minimization of a discrete L^2 norm between the Rescaled Runge function ad polynomials with bounds, $n = 4, 8, 12$.

each n the functional $\mathcal{J}_h$ is evaluated with $n+1$ equi-distributed quadrature points. In terms of complexity it corresponds to Lagrange interpolation on a uniform grid. The calculation is performed with exact gradients. The result is displayed in figure 3. One observes that the oscillations are controlled near the endpoints of the interval and the numerical convergence is achieved.

Next we change the objective function which is the rescaled Tchebycheff polynomial

$$f_2(x) = \frac{T_{20}(x)+1}{2}.$$

We use 21 equi-distribued quadratures points. The calculation are performed with a polynomial degree $p \in U_{10}$ and $p \in U_{20}$. The results displayed in figure 5 show that the numerical solution is the exact one for $n = 20$ (note that the final plot is evaluated on a grid with 400 points to reach good resolution for the oscillatory part on the right of the profile).

A second series of similar test is performed with the rescaled Tchebycheff polynomial

$$f_3(x) = \frac{T_{21}(x)+1}{2}$$

with 22 quadrature points, and with the generation of polynomial with bounds based on the second representation formula: see (25). The numerical solution with $n = 10$ is the exact one.

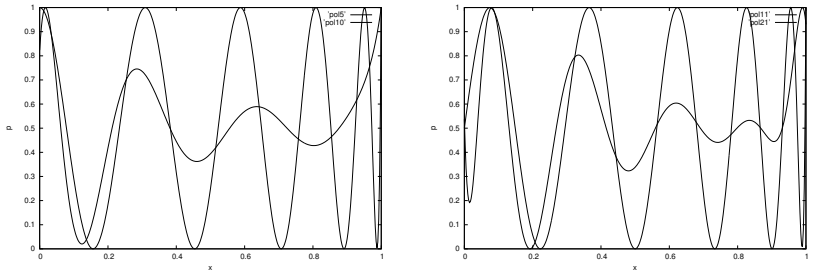

Figure 4: On the left: minimization of a discrete L^2 norm between the rescaled Tchebycheff polynomial $\frac{T_{20}(x)+1}{2}$ and polynomials with bounds, $n = 10, 20$; the numerical solution is exact for $n = 20$. On right: same problem for $\frac{T_{21}(x)+1}{2}$ and the polynomials based on the set $\mathcal{V}_5$ and $\mathcal{V}_{10}$; the numerical solution is also exact for $\mathcal{V}_{10}$ which corresponds to polynomials of maximal degree equal to 21.

In the next test, the objective function is a step function

$$f_4(x) = 0 \text{ for } x < 0.4 \text{ and } z(x) = 1 \text{ for } 0.4 < x.$$

The number of quadrature points is 25. The degree of the polynomials is 8, 16 and 24. The convergence in a discrete L^1 norm is observed. Here we do not use the exact gradient to accelerate the descent method since the L^1 norm is not differentiable. The accuracy in L^1 norm is satisfactory and the respect of the bounds is perfect.

4.3.2 Minimization of integrals with polynomial weights

The next tests minimize functionals like

$$J(p_n) = \int_0^1 (t - \lambda_n(x))\, p_n(x) dx, \quad p_n \in U_n \tag{49}$$

where $\lambda_n \in P_n$ is given and t may vary. A reference is provided by a recent work [7] where it is proved that p_n has not less than $n + 1$ points of contact counted with order of multiplicity (this is similar to one-sided L^1 minimization for which we refer to [2]) for almost all t. We use this theoretical property to check the accuracy of the approximation. We remark that the optimal solution p_n has the natural tendency to vanish where $t - \lambda_n(x) > 0$ and to be equal to 1 where $t - \lambda_n(x) < 0$, which is clearly a good strategy to minimize the cost function (49).

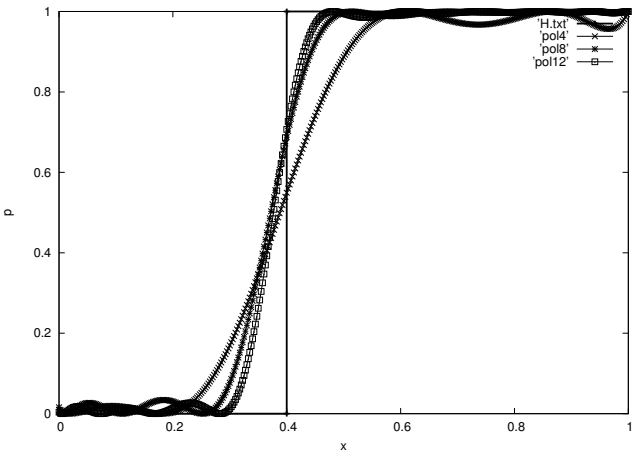

Figure 5: Minimization of the discrete L^1 between a step function and polynomials in bounds of degree 8, 16 and 24. One observes the convergence of the numerical profile to the objective function.

A numerical result representative of all the tests is the following. Take

$$\lambda_2(x) = T_2(2x-1) - t + x \text{ and } t = 0.3.$$

A first numeral simulation yields the function displayed on figure 6, the numerical value of the cost function is $J(p_n^1) \approx -0.16737$. This function does not have the required number of contacts on the figure. But another minimum is captured by numerical simulations with another starting point, for which $J(p_n^2) \approx -0.188478 < J(p_n^1)$: its total order of contact is large enough (equal to $2n+1 = 7$ since $n = 3$) and this is in accordance with the theory. No other minimum with lower value of the cost have been obtained by simulations, so it is the best candidate. See figure 7. Note that the exact calculation of the derivative $p_n'(x)$ is convenient to count without ambiguity the number of derivatives which vanish at points of contact.

4.4 Approximation in dimension $d = 2$

The 2D implementation is based on the loop (35) with the basic polynomials of lemma 3.3 and has been coded in complex algebra.
We display in figure 8 the result of numerical tests with a cost function which is the L^2 distance between p_n and the objective function

$$f_5(x,y) = \frac{T_8((2x+y)/3)+1}{2}.$$

One observes clear convergence when increasing n (up to 16). For $n = 16$ the numerical value of the cost function is ≈ 0.0195. The number of quadrature

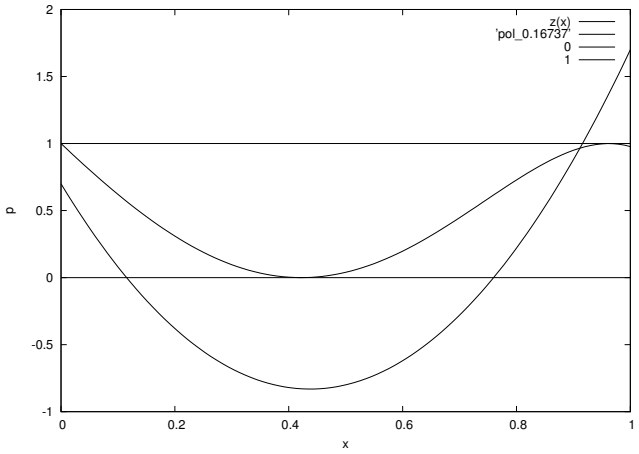

Figure 6: Plot of $\lambda_2(x) - t$ and of a local minimum p_n^1 with $J(p_n^1) \approx -0.16737$. The total order of contact if $1 + 2 + 2 = 5$.

points used to discretize the cost function is 11×11 is all tests.

The next result concerns the minimization of the discrete L^2 distance to the objective function

$$f_6(x, y) = H(2x + y) \text{ with H the Heaviside function.}$$

The L^2 distance is better in terms of the smoothness and speed of convergence with the respect to the L^1 distance. The number of quadrature points is 11×11 and $n = 10$ then 20. One observes good accuracy in figure 9.

5 Perspectives

The connections between on the one hand the algebraic properties of the four-squares Euler identity and on the other hand the use of polynomials with two bounds in numerical analysis and scientific computing raise many questions which could be the subject of future researches. The following non exhaustive list reflects the own interests of the author.

- In view of a simpler implementation, one could ask wether convenient interpolation techniques are possible within U_n. The point is that the loops discussed so far are by composition, so very different from the usual summation formulas used in interpolation [2].
- In a similar vein, it would be valuable to use these polynomials to address maximum principle properties for the discretization of non linear equations. The usual way is in a first stage to approximate a function (a flux typically for a finite volume scheme) with polynomials in P_n, in a second stage to inquire

about the verification on a local maximum principle, and in the final stage to use limiters [11] to clip the polynomials near extrema. With the set U_n the respect of the maximum principle is a priori.
- Since the loops which generate the polynomials can be coded only at quadrature points, one could think of using them to explore new sparse composition algorithms for having a polynomial approximation of a given function in very high dimension. An appealing idea is probably to couple such loops with sparse grid techniques [12].

Acknowledgements. The author thanks E. Trelat for bringing his expertise in polynomial optimization and related computations and G. Poette for many discussions on the interest of signed polynomials for uncertainty quantification.

References

[1] J. Bochnak, Michel Coste and Marie-Françoise Roy, Real algebraic geometry, A series of modern surveys in Mathematics 36, Springer 1998.

[2] R. Bojanovic and R.A. Devore, On polynomials of best one side approximation, L'enseignement mathématique, 12, 1966.

[3] F. Chatelin, A computational journey into the mind. Nat. Comput. 11 (2012), no. 1, 67-79.

[4] A. Cohen, R. DeVore and C. Schwab, Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs, Anal. Appl.9(1)(2011)11-47.

[5] K. Conrad, Quaternions algebras, expository paper online at K. Conrad webpage http://www.math.uconn.edu/~kconrad/

[6] B. Després and B. Perthame, Uncertainty propagation; intrusive kinetic formulations of scalar conservation laws, submitted to SIAM J. Uncertainty Quantification 2015.

[7] B. Després and E. Trelat, Space-time two sided L^1 approximation and optimal control of polynomial systems, in preparation, 2016.

[8] R.A. Devore and G.G. Lorenz, Constructive approximation, Springer, 1981.

[9] H.-D. Ebbinghaus et al., Numbers, Springer-Verlag, New York, 1991.

[10] S. Foucart, A mathematical introduction to compressive sensing, Applied and Numerical Harmonic Analysis, Birkhäuser/Springer, New York, 2013.

[11] E. Godlevski and P.A. Raviart, Numerical approximation of hyperbolic systems of conservation laws, Springer Verlag New York, AMS 118, 118, (1996).

[12] M. Griebel, A. Hullmann and P. Oswald, Peter Optimal scaling parameters for sparse grid discretizations. Numer. Linear Algebra Appl. 22 (2015), no. 1, 76-100.

[13] J.B. Lasserre, Moments, Positive Polynomials and Their Applications, Imperial college press, 2010.

[14] R.J. LeVeque, Numerical methods for conservation laws. (ETHZ Zurich, Birkhauser, Basel 1992).

[15] Y. Maday and Olga Mula, A Generalized Empirical Interpolation Method: Application of Reduced Basis Techniques to Data Assimilation, F. Brezzi et al. (eds.), Analysis and Numerics of Partial Differential Equations, Springer INdAM Series 4, Springer-Verlag Italia 2013

[16] D. Shapiro, Compositions of Quadratic Forms, de Gruyter, New York, 2000

[17] G. Szego, Orthogonal polynomials, AMS 1939.

[18] E. F. Toro, Riemann solvers and numerical methods in fluid dynamics, a practical introduction, Springer, 1997.

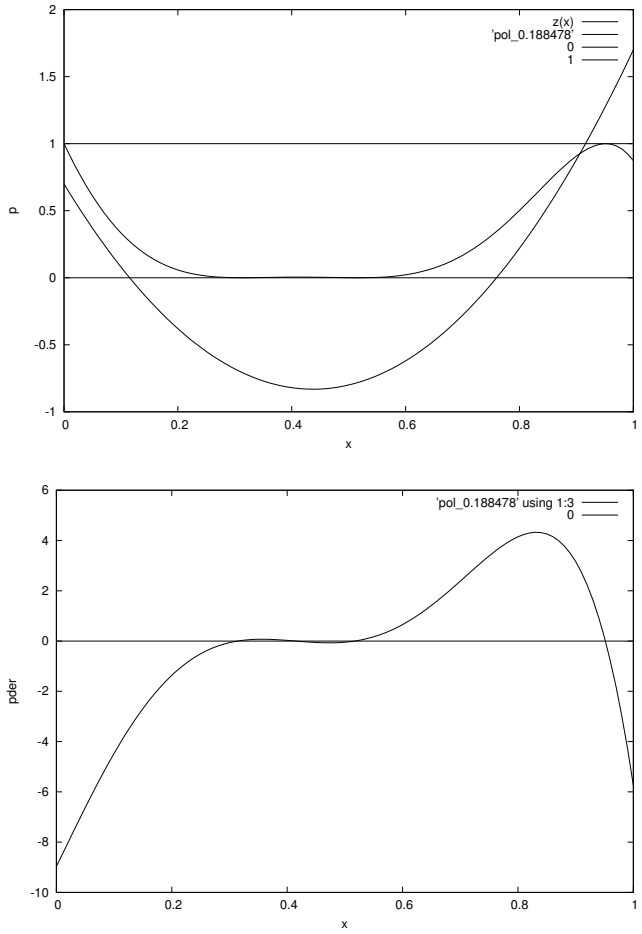

Figure 7: On the top, plot of $\lambda_2(x) - t$ and of another local minimum p_n^2 with $J(p_n^2) \approx -0.188478$. The total order of contact if $1 + 2 + 2 + 2 = 7 = 2n + 1$, and so is the best candidate to be the global minimum. On the bottom, plot of the exact derivative $(p_n^2)'$.

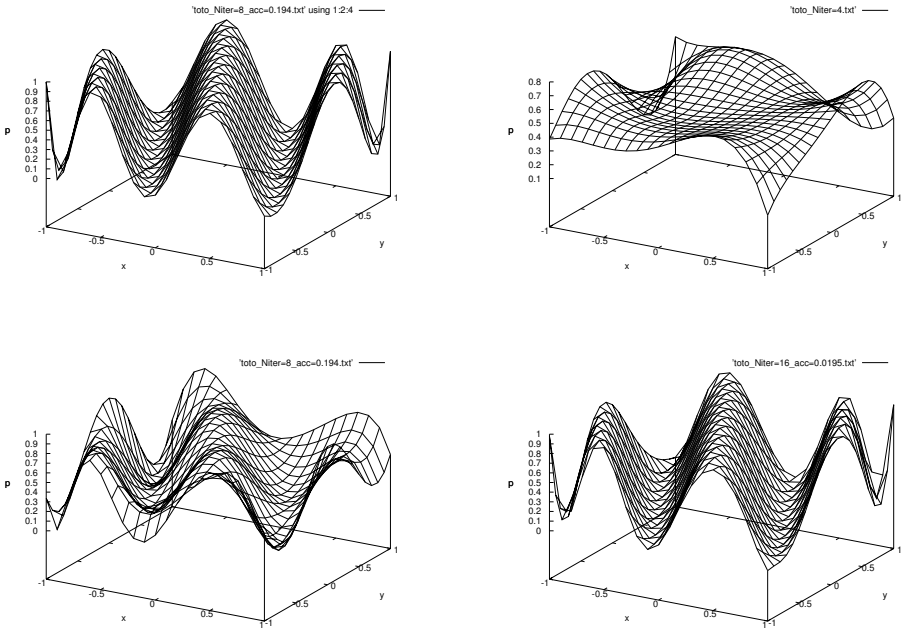

Figure 8: 2D results with parameters: objective function is f_5, we use 11×11 quadrature points, $n = 4, 8, 16$ and L^2 norm. Top left: objective function. Top right numerical solution $n = 4$. Bottom left: numerical solution $n = 8$. Bottom right: numerical solution $n = 16$.

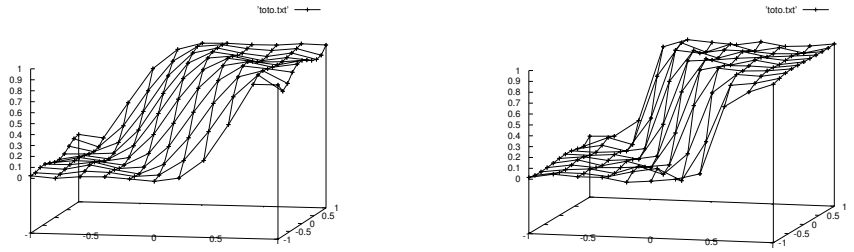

Figure 9: 2D results: objective function f_6, 11×11 quadrature points, $n = 10$, L^2 norm. Right: result with $n = 10$. Left: result with $n = 20$ which shows a numerical convergence to the step function.