HAL
archives-ouvertes.fr

ALGORITHMS FOR POSITIVE POLYNOMIAL APPROXIMATION

Frédérique Charles, Martin Campos-Pinto, Bruno Després

▶ **To cite this version:**

Frédérique Charles, Martin Campos-Pinto, Bruno Després. ALGORITHMS FOR POSITIVE POLYNOMIAL APPROXIMATION. 2017. hal-01527763

HAL Id: hal-01527763
https://hal.sorbonne-universite.fr/hal-01527763

Submitted on 25 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ALGORITHMS FOR POSITIVE POLYNOMIAL APPROXIMATION

F. CHARLES, M. CAMPOS PINTO AND B. DESPRÉS *

Abstract. We propose several algorithms for positive polynomial approximation. The main tool is a novel iterative method to compute non negative interpolation polynomials at any order, which is shown to converge under conditions that make it suitable for the numerical approximation of positive functions. Our method is based on the special representations of non negative polynomials provided by the Lukács Theorem, and a key point is the use of Chebyshev polynomials for the initial step of the iterations. Numerical results illustrate the convergence properties of the proposed algorithms, and they are completed with a first application of this technique to the positive discretization of the advection equation.

Key words. Polynomial interpolation, positive polynomials, Chebyshev polynomials.

AMS subject classifications. 65D15, 41A29, 41A55

1. Introduction.

Let P_n denote the set of real polynomials of degree $\leq n$ over the interval $[0,1]$. Let $f \in W^{1,\infty}(0,1)$ be a Lipschitz function which is positive over the interval

$$\inf_{x\in[0,1]} f(x) > 0. \tag{1.1}$$

Non negative functions $f \geq 0$ will be considered as well. The basic model problem in this work concerns interpolation in $P_n^+ := \{p_n \in P_n : \quad p_n(x) \geq 0 \ \ \forall x \in [0,1]\} \subset P_n$ of f. Since the interpolation procedure depends on the knowledge of the interpolation points and they must have an influence, it is necessary to introduce some degree of freedom. We will consider the formulation below.

PROBLEM 1.1. *Find $n+1$ interpolation points $0 \leq x_0 < x_1 < \cdots < x_n \leq 1$ such that the polynomial interpolant $p_n \in P_n$ defined by $p_n(x_i) = f(x_i)$ for all $0 \leq i \leq n$ satisfies $p_n \in P_n^+$.*

Problem 1.1 has interest for pure numerical analysis purposes, and also because the question of having good characterization and convenient use of positive polynomials is central in applications and scientific computing. A non exhaustive list of references which reflects some of our own interests is: [1, 10] for cubic polynomials, [4] for automatization of the testing, [7, 5] with sum of squares characterization, [8, 9] on considerations on computer aided design with Bernstein and Bézier curves, [15, 11, 14] for non negative numerical approximation in scientific computing for hyperbolic equations and finally [5, 6] and therein for comprehensive references on polynomial theory.

As Problem 1.1 is difficult to handle in full generality, it is convenient for theoretical purposes to introduce a numerical parameter $0 < h \leq 1$ and to consider the problem of interpolating f on subintervals of size h. Thus, a more general problem is to find $h > 0$ and an element in P_n^+ which interpolates $f_h(x) = f(xh)$ at $n+1$ points of $[0,1]$. When discretizing a partial differential equation, the parameter h is ultimately identified with the mesh size, as evidenced at the end of the numerical section. So we believe the introduction of this parameter h is also very natural in view of PDE discretizations.

The original solution to Problem 1.1 that will be proposed takes the following form.

*1-Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France, 2-CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France

ALGORITHM 1.1. *Let $f \in W^{1,\infty}(0,1)$ satisfy (1.1). Compute a sequence $p_n^m \in P_n^+$ (for $m = 0, 1, \dots$) such that $p_n^m \longrightarrow p_n \in P_n^+$ and p_n solves Problem 1.1.* A byproduct of our analysis is an algorithm for determining wether a given polynomial $f \in P_n$ satisfies the positivity condition (1.1) or not. Such an algorithm is an approximate certificate of positivity [7, 5].

ALGORITHM 1.2. *Let $f \in P_n$. Consider the sequence $(p_n^m)_{m\in\mathbb{N}}$ with $p_n^m \in P_n^+$. If for iteration numbers $m \geq m_0$ large enough, p_n^m does not tend to f, then f is not positive over $[0, 1]$.*

All the details of the algorithms and methods will be given, which will give sense to this algorithm. In practice it is possible to use the algorithm below.

ALGORITHM 1.3. *Let $f \in P_n$. Compute $p_n^m \in P_n^+$ the m-th iterate of the sequence for m large enough. Replace f by p_n^m.*

The interest of this method for scientific computing lies in the fact that if the convergence conditions of the sequence are realized, then p_n^m is close to f so the replacement introduces only a small error. But if in the other hand the convergence conditions of the sequence are not realized, then p_n^m is a positive approximation of f. In case positive polynomials are mandatory (as an ingredient in a given calculus), then such an algorithm provides a practical solution which indeed can be interpreted as a new approximate certificate of positivity (we refer to [7, 5] for this notion). The accuracy of this algorithm will be assessed in the numerical section.

Lukács characterization of P_n^+. It is not a surprise that a convenient characterization of P_n^+ plays a key role in the solution to Problem 1.1. Such one is provided by the Lukács Theorem, see e.g. [13, Sec. 1.21] for a proof based on complex valued trigonometric polynomials. A recent proof in real algebra is available in [2].

THEOREM 1.1 (Lukács).
• *If $n = 2p$, then $p_n \in P_n^+$ if and only if the polynomial can be expressed as $p_n(x) = a_p(x)^2 + x(1-x)\, b_{p-1}(x)^2$ with $a_p \in P_p$ and $b_{p-1} \in P_{p-1}$.*
• *If $n = 2p+1$, then $p_n \in P_n^+$ if and only if the polynomial can be expressed as $p_n(x) = x\, a_p(x)^2 + (1-x)\, b_p(x)^2$ with $a_p, b_p \in P_p$.*

The solution of Problem 1.1 that we propose is based on the possibility of obtaining the interpolation points in combination with a direct definition of a_p and b_p (or b_{p-1}) in the Lukács Theorem. The two main technical ideas involved in our construction, namely **oscillating polynomials** and **sliding interpolation points**, can be explained on the simplest example which uses the Lukács decomposition in P_2^+.
Thus, we consider a given polynomial $p_2 \in P_2^+$ and try to determine $a_1 \in P_1$ and $b_0 \in P_0 = \mathbb{R}$ such that

$$p_2(x) = a_1(x)^2 + x(1-x)b_0^2 \qquad x \in [0,1]. \tag{1.2}$$

Assume for simplicity that $p_2(0) > 0$ and $p_2(1) > 0$. Since the weight $x(1-x)$ vanishes at the endpoints of the interval, one has necessarily that $a_1(0)^2 = p_2(0) > 0$ and $a_2(1)^2 = p_2(1) > 0$. There are two possibilities to finalize the construction of a_1.
• The first solution uses an elementary idea proposed in [2]. Since it has a change of sign, it is called an **oscillating polynomial** in the core of the paper. That is $a_1(0) = \sqrt{p_2(0)}$ and $a_1(1) = -\sqrt{p_2(1)}$. One gets $a_1(x) = \sqrt{p_2(0)}(1-x) - \sqrt{p_2(1)}x$. The polynomial $q = p_2 - a_1^2$ vanishes at the endpoints, so one can write $p_2 - a_1^2 = we$ with $w(x) = x(1-x)$ and $e \in \mathbb{R}$. We use a consequence of the fact that the sign changes at the endpoints, which is that there exists a interpolation point $x_* \in [0,1]$ such that $a_1(x_*) = 0$. One gets that $e = \frac{p_2(x_*)}{x_*(1-x_*)} \geq 0$. So one defines $b_0 = \sqrt{e}$ which

proves the Lukács Theorem in this case.
We note that the interpolation point x_* is defined within the construction. Since its position is a priori unknown and determined self-consitently when solving the interpolation problem, we call it a **sliding interpolation point**.
• One the other hand if one tries to design a_1 with the same sign at the endpoints (it writes generically as $a_1(0) > 0$ and $a_1(1) > 0$), then a simple counterexample shows that the construction does not work. Indeed take $p_2(x) = (1-2x)^2$, then $a_1(x) = \sqrt{p_2(0)}(1-x) + \sqrt{p_2(1)}x \equiv 1$ so $b_0^2 = \frac{p_2(x)-a_1(x)^2}{x(1-x)} = -4$. Therefore the pure imaginary number $b_0 \in i\mathbb{R}^*$ cannot be a solution to the Lukács Theorem.
In summary, the use of oscillating representation allows to construct a generic solution of the Lukács Theorem, and it is a necessary and sufficient condition in this example.

Summary of the main results. All our results are based on various extensions of such representation formulas. Since the case $n = 2$ is almost trivial as seen in the previous discussion, the first non trivial interesting case is $n = 3$ and this is why we present a detailed analysis for the case $n = 2p + 1$ with $p \in \mathbb{N}$ in Sections 2 and 3. The main results however hold for any integer n and can be stated as follows, where the uniform norm over $(0, 1)$ is denoted by $\|\cdot\|$.

THEOREM 1.2. *Let $n \in \mathbb{N}$ and consider a function $f \in W^{q,\infty}(0,1)$, $1 \le q \le n+1$, that is positive over $[0,1]$. Denote $f_h(\cdot) = f(\cdot h)$ for $0 \le h \le 1$.*

Then there exists $h_* > 0$ such that for all $0 \le h \le h_*$, one can construct a sequence of positive polynomials $p_n^m \in P_n^+$ for $m = 0, 1, \ldots,$ which satisfies

$$\|p_n^m - f_h\| \le C h^{\min(q,2(m+1))} \tag{1.3}$$

with a constant C independent of h. For $n = 2p$, resp. $n = 2p+1$, the construction involves oscillating polynomials $\left(a_p^m, b_{p-1}^m\right) \in P_p \times P_{p-1}$, resp. $\left(a_p^m, b_p^m\right) \in P_p^2$, that admit a limit (a_p, b_{p-1}), resp. (a_p, b_p), and the approximations are of the form $p_p^m(x) = a_p^m(x)^2 + x(1-x)b_{p-1}^m(x)^2$, resp. $p_p^m(x) = x a_p^m(x)^2 + (1-x)b_p^m(x)^2$.

This convergence estimate will be given a detailed proof in Sections 2 and 3, see in particular Theorem 3.2. It can be complemented by the best error estimate for which we refer to [2]. A by-product is an analytical proof of the Lukács Theorem, which serves as a foundation for Algorithms 1.2 and 1.3.

THEOREM 1.3. *Assume moreover $f \in P_n^+$. Then one has at the limit $m \to \infty$ $f(hx) = a_p(x)^2 + x(1-x)b_{p-1}(x)^2$, resp. $f(hx) = x a_p(x)^2 + (1-x)b_p(x)^2$, where (a_p, b_{p-1}), resp. (a_p, b_p) is the limit described in the previous Theorem.*

The generalization of the techniques and results to the even case $n = 2p$ is described in the core of the paper. Interest of the various methods for the design of numerical algorithms will be demonstrated at the end of this work by many numerical illustrations either for small h (thus verifying a priori the hypothesis of Theorem 1.2) or for large h, typically $h = 1$. The results for $h = 1$ will clearly show that the range of parameters for which the algorithms converge is much larger than those predicted by Theorem 1.2. We think this is an extremely important information in view of possible use of such methods for practical problems.

Organization. Section 2 is dedicated to the case $n = 3$. The extension to $n = 2p + 1$ for $p \in \mathbb{N}$ is performed in section 3 by a more general method based on a simplified Newton-Raphson algorithm. Section 4 provides the formulas for the case $n = 2p$. Section 5 yields detail informations and provides the result of many numerical experiments which all show that the algorithms proposed in this work display good convergence and approximation properties for arbitrary values of h (not only small

h as in Theorem 1.2). Application to the numerical approximation of a very simple PDE (the advection equation) is performed at the very end of this work.

2. Interpolation by positive cubic polynomials. We focus on cubic polynomials because it is the first non trivial extension of the case $n = 2$ considered in (1.2). According to the Lukács representation theorem, a polynomial in P_3^+ writes

$$p_3(x) = x\, a_1(x)^2 + (1-x)\, b_1(x)^2$$

where a_1 and b_1 are affine polynomials. Since a_1 (resp. b_1) is linear, it can be reconstructed by linear interpolation knowing two values. One is natural: indeed for $x = 1$ the identity recasts as $p_3(1) = a_1(1)^2$. So the main question is to obtain another interpolation point. Being optimistic, let us assume there exists $0 < \alpha < 1$ such that $b_1(\alpha) = 0$. One obtains $p_3(\alpha) = \alpha a_1(\alpha)^2$ and the representation $a_1(x) = \sqrt{p_3(1)}\frac{(x-\alpha)}{1-\alpha} \pm \sqrt{\frac{p_3(\alpha)}{\alpha}}\frac{(1-x)}{1-\alpha}$. As explained in the introduction it is much better to take the minus sign in order to construct an **oscillating polynomial** , which is $a_1(x) = \sqrt{p_3(1)}\frac{(x-\alpha)}{1-\alpha} - \sqrt{\frac{p_3(\alpha)}{\alpha}}\frac{(1-x)}{1-\alpha}$. For similar reasons we will consider by symmetry $b_1(x) = \frac{(\beta-x)}{\beta}\sqrt{p_3(0)} - \sqrt{\frac{p_3(\beta)}{\beta}}\frac{x}{\beta}$. The condition $a_1(\beta) = b_1(\alpha) = 0$ is studied in the following section.

2.1. A sufficient criterion for positive interpolation. Elaborating on the idea that p_3 interpolates f_h at the 4 points 0, α, β and 1, one obtains a preliminary result.

PROPOSITION 2.1. *Let $f \in W^{1,\infty}(0,1)$ satisfy (1.1). If $a_1, b_1 \in P_1$ and $\alpha, \beta \in (0,1)$ are such that*

$$\begin{cases} a_1(\alpha) = -\sqrt{\frac{f(\alpha)}{\alpha}} & a_1(\beta) = 0 \quad a_1(1) = \sqrt{f(1)} \\ b_1(0) = -\sqrt{f(0)} & b_1(\alpha) = 0 \quad b_1(\beta) = \sqrt{\frac{f(\beta)}{1-\beta}} \end{cases} \tag{2.1}$$

then $0 < \alpha < \beta < 1$ and $p_3(x) = xa_1(x)^2 + (1-x)b_1(x)^2$ is a positive cubic polynomial that interpolates f at 0, α, β and 1.

Proof. Although straightforward, the above criterion is convenient as it allows to restate the positive interpolation of f as a fixed point problem. Indeed we see that if (2.1) holds then a_1 and b_1 read

$$a_1(x) = -\Big(\frac{x-1}{\alpha-1}\Big)\sqrt{\frac{f(\alpha)}{\alpha}} + \Big(\frac{x-\alpha}{1-\alpha}\Big)\sqrt{f(1)},\ b_1(x) = -\Big(\frac{x-\beta}{-\beta}\Big)\sqrt{f(0)} + \Big(\frac{x}{\beta}\Big)\sqrt{\frac{f(\beta)}{1-\beta}} \tag{2.2}$$

and the associated nodes α and β satisfy the relations

$$(\beta-\alpha)\sqrt{f(1)} + (\beta-1)\sqrt{\frac{f(\alpha)}{\alpha}} = 0, \quad (\alpha-\beta)\sqrt{f(0)} + \alpha\sqrt{\frac{f(\beta)}{1-\beta}} = 0. \tag{2.3}$$

Conversely, if α and β are solution to (2.3) then the criterion (2.1) will be satisfied with a_1 and b_1 given by (2.2). □

By looking at (2.3) the first equation allows to express β as a function of α while the second equation gives α as a function of β. In particular, (2.3) rewrites as the fixed point problem

$$(\alpha, \beta) = G(\alpha, \beta) \tag{2.4}$$

with $G(\alpha,\beta) := (\varphi(\beta),\psi(\alpha))$, where φ and ψ are two additional functions from $[0,1] \to [0,1]$ defined as

$$\varphi(\beta) = \frac{\beta\sqrt{(1-\beta)f(0)}}{\sqrt{(1-\beta)f(0)}+\sqrt{f(\beta)}} \quad \text{and} \quad \psi(\alpha) = \frac{\alpha\sqrt{\alpha f(1)}+\sqrt{f(\alpha)}}{\sqrt{\alpha f(1)}+\sqrt{f(\alpha)}}. \tag{2.5}$$

The equation (2.4) recasts as $\tau(\alpha) = 0$ where the function is $\tau(\alpha) = \alpha - \varphi(\psi(\alpha))$.

LEMMA 2.2. *One has $\tau \in C^0([0,1]:[-1,1])$ and $\tau(0) = 0$ with $\tau(1) = 1$. Assuming that $f \in W^{1,\infty}[0,1]$ satisfy (1.1), one that $\frac{d}{d\alpha}\tau(0^+) = -\infty$.*

Proof. The values at the endpoints are $\psi(0) = \psi(1) = 1$, $\varphi(0) = \varphi(1) = 0$ and so $\tau(0) = 0$ and $\tau(1) = 1$.
The derivative at the origin can be checked as follows. One has $\psi(\alpha) = \alpha + (1-\alpha)\frac{\sqrt{f(\alpha)}}{\sqrt{\alpha f(1)}+\sqrt{f(\alpha)}}$. So $\psi(\alpha)' = 1 - \frac{\sqrt{f(1\alpha)}}{\sqrt{\alpha f(1)}+\sqrt{f(\alpha)}} + (1-\alpha)\sqrt{f(\alpha)}'\left(\sqrt{\alpha f(1)}+\sqrt{f(\alpha)}\right)^{-1}$
$-(1-\alpha)\sqrt{f(\alpha)}\left(\sqrt{\alpha f(1)}+\sqrt{f(\alpha)}\right)^{-2}\left(\sqrt{\alpha f(1)}+\sqrt{f(\alpha)}\right)'$.

All terms are bounded uniformly for small $\alpha > 0$ except the last parenthesis because $\sqrt{\alpha f(1)}' = \frac{1}{2}\sqrt{\frac{f(1)}{\alpha}} \to +\infty$ for $\alpha \to 0^+$. Therefore $\psi(0^+)' = -\infty$. Similarly the dominant term in $\varphi(\beta)'$ for β close to 1 is $\varphi(\beta)' \approx -\frac{1}{2\sqrt{1-\beta}}\frac{\beta}{\sqrt{(1-\beta)f(0)}+\sqrt{f(\beta)}} \to -\infty$ for $\beta \to 1^-$. So $\tau(0^+)' \approx -\psi(0^+)'\varphi(0^+)' = -\infty$ and the proof is ended. ∎

COROLLARY 2.3. *There exists two interpolation points $0 < \alpha < \beta < 1$ such that the polynomial $p_3 \in P_3^+$ defined in Proposition 2.1 interpolates f at 0, α, β and 1.*

Proof. Evident since there exists by continuity $0 < \alpha < 1$ such that $\tau(\alpha) = 0$. ∎

2.2. A fixed point algorithm to compute the cubic nodes. The goal in this section is to construct a fixed point algorithm with good convergence properties to compute the solution of (2.4). The convergence is better studied after rescaling the function f, so we systematically replace $f(\cdot)$ by $f_h(\cdot) = f(h\cdot)$ in (2.4-2.5). The function G becomes $G_h(\alpha,\beta) = (\varphi_h(\beta),\psi_h(\alpha))$ where

$$\varphi_h(\beta) = \frac{\beta\sqrt{(1-\beta)f_h(0)}}{\sqrt{(1-\beta)f_h(0)}+\sqrt{f_h(\beta)}} \quad \text{and} \quad \psi_h(\alpha) = \frac{\alpha\sqrt{\alpha f_h(1)}+\sqrt{f_h(1\alpha)}}{\sqrt{\alpha f_h(1)}+\sqrt{f_h(\alpha)}}. \tag{2.6}$$

ALGORITHM 2.1. *Given $X^0 = (\alpha^0,\beta^0) \in (0,1)^2$, consider the fixed point method*

$$X^{m+1} := G_h(X^m). \tag{2.7}$$

Since f is positive we verify that $G_h([0,1]^2) \subset [0,1]^2$, hence it defines an iterative scheme. The goal hereafter is to determine reasonable conditions such that the fixed point converges. The good news is that the function G_h has some good properties for h small enough because its limit G_0 has good properties and the scheme (2.7) converges at a fast rate. This can be understood by considering the limit case $h = 0$ which corresponds to a constant target function $f_0 = f(0)$. Indeed one can recasts G_h as $G_h(\alpha,\beta) = \Big(\mathcal{K}\big(\beta,\sigma_h(\beta)\big), 1-\mathcal{K}\big(1-\alpha,\tau_h(\alpha)\big)\Big)$ with $\mathcal{K}(z,r) := \frac{z\sqrt{1-z}}{r+\sqrt{1-z}}$, $\tau_h(\alpha) := \sqrt{\frac{f(\alpha h)}{f(h)}}$ and $\sigma_h(\beta) := \sqrt{\frac{f(\beta h)}{f(0)}}$. Using the Lipschitz regularity of f we see that both τ_h and σ_h converge towards 1 uniformly on $[0,1]$. Hence G_h converges by continuity uniformly on $[0,1]^2$ towards

$$G_0(\alpha,\beta) = \Big(\mathcal{K}(\beta), 1-\mathcal{K}(1-\alpha)\Big) \tag{2.8}$$

where we have written for simplicity $\mathcal{K}(z) = \mathcal{K}(z,1)$.

LEMMA 2.4. *The function G_0 has the following properties*

i) *it leaves invariant the domain $F := \left[\frac{1}{5},\frac{1}{3}\right] \times \left[\frac{2}{3},\frac{4}{5}\right]$,*

ii) *it is contractant on F in the maximal norm,*

iii) *it has a unique fixed point in $(0,1)^2$, which is*

$$\underline{X} = (\underline{\alpha},\underline{\beta}) = \Big(\frac{1}{4},\frac{3}{4}\Big) \in F \tag{2.9}$$

iv) *the Jacobian matrix $\nabla G_0 = (\partial_j (G_0)_i)_{1\le i,j\le 2}$ vanishes at $\underline{X}$*

$$\nabla G_0(\underline{X}) = 0. \tag{2.10}$$

REMARK 2.5. *On $[0,1]^2$ the function G_0 has a second fixed point $\widehat{X} = (0,1)$, but that is not an admissible solution to Proposition 2.1 since the nodes α and β must be distinct from the end nodes 0 and 1 for (2.1) to make sense.*

Proof. One has that $\mathcal{K}'(\beta) = \frac{\sqrt{1-\beta}}{1+\sqrt{1-\beta}} - \frac{\beta}{2\sqrt{1-\beta}(1+\sqrt{1-\beta})^2} = 1 - \frac{1}{2\sqrt{1-\beta}}$ so that $\mathcal{K}$ is increasing on $[0,\frac{3}{4}]$ and decreasing on $[\frac{3}{4},1]$, with $\mathcal{K}(\frac{3}{4}) = \frac{1}{4}$. This yields

$$\mathcal{K}\left(\Big[\frac{2}{3},\frac{4}{5}\Big]\right) = \left[\min\Big(\mathcal{K}\Big(\frac{2}{3}\Big),\mathcal{K}\Big(\frac{4}{5}\Big)\Big),\frac{1}{4}\right] = \left[\frac{2}{3(1+\sqrt{3})},\frac{1}{4}\right] \subset \Big]\frac{1}{5},\frac{1}{3}\Big[$$

which, using the expression (2.8), leads to

$$G_0(F) \subset \Big]\frac{1}{5},\frac{1}{3}\Big[\times \Big]\frac{2}{3},\frac{4}{5}\Big[\subset F. \tag{2.11}$$

To show the second claim we compute $\nabla G_0(\alpha,\beta) = \begin{pmatrix} 0 & \mathcal{K}'(\beta) \\ -\mathcal{K}'(1-\alpha) & 0 \end{pmatrix}$ and we observe that the bound $|\mathcal{K}'(\beta)| \le 1 - \frac{\sqrt{3}}{2} < 0.15$, valid on $F_\beta = [\frac{2}{3},\frac{4}{5}]$, translates to $|\mathcal{K}'(1-\alpha)| < 0.15$ on $F_\alpha = [\frac{1}{5},\frac{1}{3}]$, which proves the contraction property. The third claim is straightforward to verify, and the last one follows from the fact that $\mathcal{K}'(\underline{\beta}) = \mathcal{K}'(1-\underline{\alpha}) = \mathcal{K}'\big(\frac{3}{4}\big) = 0$. □

Since h is expected to become small in the theory, we consider as an initial value the fixed point (2.9) of the limit case $h = 0$, namely

$$X^0 := \underline{X} = \Big(\frac{1}{4},\frac{3}{4}\Big). \tag{2.12}$$

The following theorem shows that this gives indeed a convergent procedure.

THEOREM 2.1. *Let $f \in W^{1,\infty}(0,1)$ satisfy (1.1). There exist $h_0 > 0$ such that for all $0 \le h \le h_0$, the sequence $(X^m)_{m\ge 0}$ given by (2.7) and (2.12) remains in the domain $F \subset]0,1[^2$ and converges to a fixed point of G_h denoted as $X_h^\infty \in F$. Moreover one has the inequality for all $m \ge 0$*

$$\|X_h^\infty - X^m\| \le 2\Big(\frac{h}{2h_0}\Big)^{m+1}. \tag{2.13}$$

The proof relies on a technical lemma that investigate some properties of G_h for small but non-zero values of h. The properties of G_h are by continuity a consequence of the properties of G_0.

LEMMA 2.6. *Let $f \in W^{1,\infty}(0,1)$ satisfy (1.1), and F be the domain defined in Lemma 2.4. Then there exist $h^* > 0$ and a constant C^* such that for all $0 \le h \le h^*$,*

i) *G_h leaves the domain F invariant,*
ii) *G_h is contractant on F in the maximal norm,*
iii) *the Jacobian matrix ∇G_h satisfies*

$$\|\nabla G_h(X)\| \le C^*(h + \|X - \underline{X}\|), \qquad X \in F, \tag{2.14}$$

iv) *the derivative of G_h with respect to h satisfies*

$$\|\partial_h G_h(X)\| \le C^*, \qquad X \in F. \tag{2.15}$$

iguyg ed

Proof. The proof is one step after the other.

i) This claim for h smaller than some h_1^* follows from the uniform convergence $G_h \to G_0$ and the embedding (2.11) which involves a proper subset of F.

ii) The Jacobian matrix is

$$\nabla G_h(\alpha, \beta) = \begin{pmatrix} 0 & \partial_\beta \mathcal{K}(\cdot, \sigma_h)(\beta) \\ -\partial_\alpha \mathcal{K}(1 - \cdot, \tau_h)(\alpha) & 0 \end{pmatrix} \tag{2.16}$$

with $\partial_\beta \mathcal{K}(\cdot, \sigma_h)(\beta) = \partial_1 \mathcal{K}(\beta, \sigma_h(\beta)) + \sigma_h'(\beta)\partial_2 \mathcal{K}(\beta, \sigma_h(\beta))$. We observe from $\sigma_h'(\beta) = hf'(h\beta)/(2\sqrt{f(0)f(\beta h)})$ that for a positive Lipschitz f the limit $\sigma_h'(\beta) \to 0$ holds uniformly on $[0, 1]$, just as $\sigma_h(\beta) \to 1$. We infer that $\partial_\beta \mathcal{K}(\cdot, \sigma_h)(\beta) \to \mathcal{K}'(\beta)$ holds uniformly on $[0, 1]$. As the same arguments apply to the α-dependent term in (2.16), it shows that $\nabla G_h \to \nabla G_0$ holds uniformly on $[0, 1]$, so that the contraction property of G_0 is transferred to G_h for h smaller than some $h_2^* > 0$, and we take $h^* = \min(h_1^*, h_2^*)$.

iii) Observe that $\nabla G_h(X) = \Phi(h, X) + h\Psi(h, X)$ where Φ (resp. Ψ) involves values of f (resp. f') and is Lipschitz (resp. bounded) on $[0, h^*] \times F$. Using (2.10) one has that

$$\begin{aligned} \|\nabla G_h(X)\| &= \|\nabla G_h(X) - \nabla G_0(\underline{X})\| \\ &\le \|\Phi(h, X) - \Phi(0, \underline{X})\| + h\|\Psi(h, X)\| \\ &\le C^*(h + \|X - \underline{X}\|) \end{aligned}$$

holds for all $(h, X) \in [0, h^*] \times F$, with a constant C^* depending on f (as h^* depends only on f).

iv) The last claim is straightforward (with another constant), using the fact that f is Lipschitz and bounded away from 0.

The proof is ended. □

Proof. [Proof of Theorem 2.1] From the claims (i) and (ii) of Lemma 2.6 we see that for $h \le h^*$, the sequence $(X^m)_{m \ge 0}$ lies within F, and from the fixed point theorem of Picard it converges towards X_h^∞, the unique fixed point of G_h in F.
The error estimate (2.13) is shown in two steps, as follows. Firstly we claim that for h small enough, the ball $B(\underline{X}, 2C^*h)$ is left invariant by G_h, i.e., $G_h(B(\underline{X}, 2C^*h)) \subset B(\underline{X}, 2C^*h)$. To see this let $X \in B(\underline{X}, 2C^*h)$ and write, using (2.14)-(2.15), that

$$\begin{aligned} \|G_h(X) - \underline{X}\| &\le \|G_h(X) - G_h(\underline{X})\| + \|G_h(\underline{X}) - G_0(\underline{X})\| \\ &\le C^*(h + \|X - \underline{X}\|)\|X - \underline{X}\| + C^*h \\ &\le 2C^*h\big(C^*(h + 2C^*h) + \tfrac{1}{2}\big), \end{aligned}$$

hence $G_h(X) \in B(\underline{X}, 2C^*h)$ for $h \le h_0 := \min\big(h^*, \big(2C^*(1 + 2C^*)\big)^{-1}\big)$. In particular, all the terms X^m are in this ball. Secondly writing $e_h^m = \|X^m - X_h^\infty\|$ and applying

again (2.14) we bound $e_h^{m+1} = \|G_h(X^m) - G_h(X_h^\infty)\| \le C^*(h + 2C^*h)e_h^m \le \frac{h}{2h_0} e_h^m$ so that $e_h^m \le \big(\frac{h}{2h_0}\big)^m e_h^0$. Estimate (2.13) follows by noticing that $e_h^0 = \|X^0 - X_h^\infty\| = \|\underline{X} - X_h^\infty\| \le 2C^*h \le \frac{h}{h_0}$ using the above considerations. The proof is ended. □

2.3. Accuracy of the approximate interpolants in P_3^+. Denoting the m-th approximation of the iterative scheme (2.7), (2.12) by $(\alpha^m, \beta^m) := X^m$ and observing that $X^m \in]0,1[^2$ (as G_h leaves that domain invariant for all h), we define affine polynomials following (2.2), namely

$$\begin{aligned} a_1^m(x) &:= \Big(\frac{x-\alpha^m}{1-\alpha^m}\Big)\sqrt{f(h)} - \Big(\frac{x-1}{\alpha^m - 1}\Big)\sqrt{\frac{f(h\alpha^m)}{\alpha^m}} \\ b_1^m(x) &:= \Big(\frac{x-\beta^m}{-\beta^m}\Big)\sqrt{f(0)} - \Big(\frac{x}{\beta^m}\Big)\sqrt{\frac{f(h\beta^m)}{1-\beta^m}}. \end{aligned} \tag{2.17}$$

Let

$$p_3^m(x) := x a_1^m(x)^2 + (1-x) b_1^m(x)^2 \tag{2.18}$$

be the corresponding cubic approximation to f_h. Our error estimate then shows that only one iteration in is actually needed for the optimal convergence in h. Indeed the estimate (2.19) is optimal for $q = 4$ and $2(m+1) = q$, that is $m = 1$.

THEOREM 2.2. *Let $f \in W^{q,\infty}(0,1)$, $1 \le q \le 4$, satisfy* (1.1)*, and let $h_0 > 0$ be given by Theorem 2.1. Then for all $0 \le h \le h_0$ and all $m \ge 0$, the cubic polynomial* (2.18) *satisfies*

$$\|p_3^m - f_h\| \le C h^{\min(q, 2(m+1))} \tag{2.19}$$

for a constant C depending on f.

Proof. The result follows by inspecting the values of p_3^m at the four points $0 < \alpha^m < \beta^m < 1$. One has, at the end-nodes $p_3^m(0) = f_h(0)$, $p_3^m(1) = f_h(1)$, and at the interior ones $p_3^m(\alpha^m) = f_h(\alpha^m) + (1-\alpha^m) b_1^m(\alpha^m)^2$ and $p_3^m(\beta^m) = \beta^m a_1^m(\beta^m)^2 + f_h(\beta^m)$. So, as $(\alpha^m, \beta^m) = X^m$ is only an approximation to $(\alpha^\infty, \beta^\infty) = X_h^\infty$, we see that, a priori, p_3^m does not interpolate f_h on the former. However the error can be estimated as follows. If $h \le h_0$ then we know from Theorem 2.1 that

$$(\alpha^m, \beta^m) \in F = \Big[\frac{1}{5}, \frac{1}{3}\Big] \times \Big[\frac{2}{3}, \frac{4}{5}\Big], \quad m \ge 0, \tag{2.20}$$

in particular these nodes are bounded away from 0 and 1. Using also that f is Lipschitz and bounded away from 0, we see that $b_1^m(x) = b_1[\beta^m](x)$ is Lipschitz as a function of $(x, \beta^m) \in [0,1] \times F_\beta$. Writing $b_1^\infty = b_1[\beta^\infty]$ we compute

$$\begin{aligned} |p_3^m(\alpha^m) - f_h(\alpha^m)| \le b_1^m(\alpha^m)^2 &= |b_1^m(\alpha^m) - b_1^\infty(\alpha^\infty)|^2 \\ &\le \big(|b_1^m(\alpha^m) - b_1^\infty(\alpha^m)| + |b_1^\infty(\alpha^m) - b_1^\infty(\alpha^\infty)|\big)^2 \\ &\le C\big(|\beta^m - \beta^\infty| + |\alpha^m - \alpha^\infty|\big)^2 \end{aligned}$$

and the same bound holds for $|p_3^m(\beta^m) - f_h(\beta^m)|$. If we now denote by $\tilde{p}_3^m$ the cubic polynomial that interpolates f_h on the distinct nodes $0, \alpha^m, \beta^m, 1$ (which are distinct and bounded away from each other according to (2.20)), standard polynomial

interpolation estimates and the equivalence of norms on $\{p \in P_3,\, p(0) = p(1) = 0\}$ give

$$\begin{aligned}\|p_3^m - f_h\| &\le \|p_3^m - \tilde{p}_3^m\| + \|\tilde{p}_3^m - f_h\| \\ &\le C\big(|(p_3^m - \tilde{p}_3^m)(\alpha^m)| + |(p_3^m - \tilde{p}_3^m)(\beta^m)| + \|f_h^{(q)}\|\big) \\ &\le C\big(\|X^m - X_h^\infty\|^2 + h^q \|f^{(q)}\|\big)\end{aligned}$$

with a constant depending on f. Using (2.13) this concludes the proof. □

3. Interpolation in P_n^+ with $n = 2p + 1$, $p \ge 2$. The objective of this section is to extend to arbitrary odd high order the method proposed in the previous section for $n = 3$. The extension to even order poses no additional difficulties and will be explained in section 4. One is faced essentially with two difficulties.

Firstly the exact calculation of the roots of a polynomial of arbitrary order is of course not possible. Therefore it is not possible to generalize in a straightforward manner the method (2.4) because it was based on the exact calculation of the root of an affine polynomial. This difficult is easily avoided by a formulating the fixed point problem as the set of polynomial equations that the roots must satisfy, see (3.7). The new algorithm (3.8) in Section 3.1 is based on a standard Newton-Raphson algorithm which can be re-interprated as an iterative procedure for the **sliding interpolation points**.

The second difficulty is perhaps more fundamental. Indeed one needs to start the generalized Newton-Raphson algorithm from a non ambiguous starting point, and one needs to proove that the Jacobian needed to formulate the generalized Newton-Raphson algorithm is a non singular matrix. It appears that it is possible to obtain an efficient solution that solves this second difficulty by using a suitable combination of the Chebyshev polynomials which are natural **oscillating polynomials**. This is detailed in Section 3.2.

3.1. A sufficient criterion for positive interpolation. For higher odd degrees n the sufficient criterion of proposition 2.1 generalizes as follows, see Figure 3.1 with a sketch of the **oscillating polynomials**. Let $f \in W^{1,\infty}(0,1)$ be positive over $[0,1]$, see (1.1), and let $h \ge 0$. We consider two polynomials $a_{p,h}, b_{p,h} \in P_p$ which have a dependence with respect to h, and we state a criterion which will allow to restate the positive interpolation of f as a fixed point problem.

PROPOSITION 3.1. *Assume there exists $2p$ nodes $0 = \alpha_0 < \ldots < \alpha_{p-1}$, $\beta_1 < \cdots < \beta_p = 1$ in $[0,1]$ with the properties:*
a) the nodes are roots of the polynomials in the sense

$$b_{p,h}(\alpha_i) = a_{p,h}(\beta_{i+1}) = 0 \qquad \text{for} \quad 0 \le i \le p-1, \tag{3.1}$$

b) the nodes interpolate $\sqrt{f(hx)/x}$ and $\sqrt{f(h)x/(1-x)}$ with alternating signs

$$a_{p,h}(\alpha_i) = (-1)^{i+p}\sqrt{\frac{f(h\alpha_i)}{\alpha_i}}, \quad b_{p,h}(\beta_i) = (-1)^{i+p}\sqrt{\frac{f(h\beta_i)}{1-\beta_i}} \quad \text{for} \quad 0 \le i \le p. \tag{3.2}$$

Then the nodes interlace $0 = \beta_0 < \alpha_0 < \beta_1 < \cdots < \beta_p < \alpha_p = 1$ and the polynomial $p_n(x) = x a_{p,h}(x)^2 + (1-x) b_{p,h}(x)^2 \in P_n^+$ is the interpolation polynomial of $f_h = f(h\,\cdot)$ on the $n+1$ nodes $\beta_0, \alpha_0, \cdots, \beta_p, \alpha_p$.

REMARK 3.2. *Actually the polynomial p_n depends also on h by construction, but to simplify the notation we disregard the index h. On the other hand, it is important*

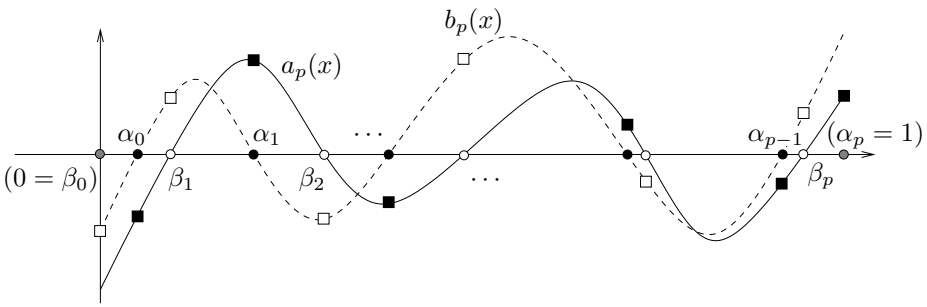

FIG. 3.1. *Sketch of two* **oscillating polynomials** $a_{p,h}$ *and* $b_{p,h}$ *in* P_p *satisfying the criterion of Proposition 3.1. The black (resp. white) circles indicate the positions of the nodes* $\alpha = (\alpha_0, \dots, \alpha_{p-1})$ *(resp.* $\beta = (\beta_1, \dots, \beta_p)$*) where* $b_{p,h}$ *(resp.* $a_{p,h}$*) vanishes, and the squares represent the data to be interpolated at these nodes, see* (3.2)*. The end nodes (*$\beta_0 = 0$ *and* $\alpha_p = 1$*) are in gray as neither* $a_{p,h}$ *nor* $b_{p,h}$ *vanish there, but they are involved in the interpolation process.*

to keep the indication of h *in the polynomials* $a_{p,h}$ *and* $b_{p,h}$ *to make a clear distinction with* $\underline{a}_p$ *and* $\underline{b}_p$ *which will be defined later.*

Proof. Since f is positive over $[0,1]$, then the sign of $a_{p,h}(\alpha_i)$ is alternating: so its roots β_i alternate with the α_i. The the same starting from the sign of of $b_{p,h}(\beta)$. By construction $p_n(\alpha_i) = f(h\alpha_i)$ for all $1 \le i \le p$ and $p_n(\beta) = f(h\beta_i)$ for $0 \le i \le p-1$. It yields $2p$ interpolation points so $p_n \in P_n$ is the interpolation polynomial at these points. The proof is ended. □

We denote

$$I_p = \{(x_1, \dots x_p) \subset (0,1)^p,\ 0 < x_1 < \cdots < x_p < 1\}\,. \tag{3.3}$$

For

$$(\alpha, \beta) = (\alpha_0, \dots, \alpha_{p-1};\, \beta_1, \dots, \beta_p) \in I_p^2,$$

we let $a_{p,h}[\alpha]$ and $b_{p,h}[\beta]$ be the polynomials which satisfy the interpolation relations (3.2)

$$a_{p,h}[\alpha](x) = \sum_{0 \le i \le p} (-1)^{i+p} \sqrt{\frac{f(h\alpha_i)}{\alpha_i}} \prod_{0 \le j \ne i \le p} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \tag{3.4}$$

and

$$b_{p,h}[\beta](x) = \sum_{0 \le i \le p} (-1)^{i+p} \sqrt{\frac{f(h\beta_i)}{1 - \beta_i}} \prod_{0 \le j \ne i \le p} \frac{x - \beta_j}{\beta_i - \beta_j}. \tag{3.5}$$

Define $\Theta_{p,h} : I_p^2 \longrightarrow \mathbb{R}^{2p}$ by

$$\Theta_{p,h}(\alpha, \beta) = (b_{p,h}[\beta](\alpha_0), \dots, b_{p,h}[\beta](\alpha_{p-1}), a_{p,h}[\alpha](\beta_1), \dots, a_{p,h}[\alpha](\beta_p))\,. \tag{3.6}$$

Then the root relation (3.1) in Proposition 3.1 is equivalent to say that $(\alpha, \beta) \in I_p^2$ satisfies

$$\Theta_{p,h}(\alpha, \beta) = 0. \tag{3.7}$$

This equation is highly non linear and might a priori degenerate, for example if $\alpha_i = \alpha_{i+1}$ or if $\beta_i = \beta_{i+1}$. The whole point of this work is that such degeneracy is easy to avoid using convenient simplified Newton-Raphson algorithms such as the next one.

ALGORITHM 3.1 (Simplified Newton-Raphson algorithm). *Given a starting point $X^0 \in I_p^2$, compute*

$$X^{m+1} := X^m - J_p(X^0)^{-1}\Theta_{p,h}(X^m) \tag{3.8}$$

where $J_p(X^0) = \nabla\Theta_{p,0}(X^0) \in \mathbb{R}^{2p\times 2p}$ is the Jacobian matrix of $\Theta_{p,0}$ evaluated at the starting point X^0

$$J_p(X^0) = \begin{pmatrix} \nabla_\alpha b_{p,0}[\beta](\alpha) & \nabla_\beta b_{p,0}[\beta](\alpha) \\ \nabla_\alpha a_{p,0}[\alpha](\beta) & \nabla_\beta a_{p,0}[\alpha](\beta) \end{pmatrix}\Big|_{(\alpha,\beta)=X^0}. \tag{3.9}$$

Our goal is now to justify this method. This will be done in three steps. The first step is the definition of a proper starting point $X^0 \in I_p^2$, the second step is the justification that $J_p(X^0)$ is a non singular matrix and the obtention of additional technical properties, and the last step is the proof of the convergence of the fixed point algorithm (3.8) for small enough conveniently chosen $h > 0$.

3.2. Definition of the starting point X^0. In the case $p = 1$ (that is $n = 3$) the starting point was obtained by considering the case $h = 0$ and we may assume here that $f(0) = 1$ to simplify. Another interpretation is that we desire the algorithm (3.8) to be exact for the simplest non trivial case which is $f \equiv 1$ because it is difficult to expect any good property of (3.8) if this trivial case cannot be addressed efficiently.

In view of the interpolation identity described in Proposition 3.1, the solution is related to the determination of two polynomials denoted as $\underline{a}_p, \underline{b}_p \in P_p$ such that $x\underline{a}_p(x)^2 + (1-x)\underline{b}_p(x)^2 = 1$ for all x. Since it is a weighted sum of squares identically equal to 1, it is natural to look for a solution based on the Chebyshev polynomials $(T_p, U_p) \in P_p \times P_{p-1}$,

$$T_p(\cos\theta) = \cos(p\theta) \quad \text{and} \quad U_p(\cos\theta) = \frac{\sin(p\theta)}{\sin\theta}, \quad p \geq 0$$

which satisfy $T_p(-x) = (-1)^p T_p(x)$ and $U_p(-x) = (-1)^{p-1}U_p(x)$ (note: the usual notation for U_p is U_{p-1}).

LEMMA 3.3. *Given $p \in \mathbb{N}$ and $i = 0, \dots, p$, let*

$$\underline{\alpha}_i := \frac{1}{2}\left[1 - \cos\left(\frac{(2i+1)\pi}{2p+1}\right)\right] \quad \text{and} \quad \underline{\beta}_i := \frac{1}{2}\left[1 - \cos\left(\frac{2i\pi}{2p+1}\right)\right] \tag{3.10}$$

and let $\underline{a}_p$ and $\underline{b}_p$ be the polynomials defined according to (3.4)-(3.5) with a constant function $f = 1$. We have the following properties.

i) *Interlacing and symmetry of the nodes: we have*

$$0 = \underline{\beta}_0 < \underline{\alpha}_0 < \underline{\beta}_1 < \cdots < \underline{\beta}_p < \underline{\alpha}_p = 1 \tag{3.11}$$

and $\underline{\alpha}_i + \underline{\beta}_{p-i} = 1$, for $0 \leq i \leq p$.

ii) *Chebyshev form: the above polynomials read*

$$\begin{aligned} \underline{a}_p(x) &= T_p(2x-1) - 2(1-x)U_p(2x-1) \\ \underline{b}_p(x) &= T_p(2x-1) + 2xU_p(2x-1). \end{aligned} \tag{3.12}$$

iii) *Symmetry: for all x, we have $\underline{a}_p(1-x) = (-1)^p \underline{b}_p(x)$.*

iv) *Root property: $\underline{a}_p$ and $\underline{b}_p$ have p simple roots in $]0,1[$, which coincide with $\underline{\beta} = (\underline{\beta}_1, \dots, \underline{\beta}_p)$ and $\underline{\alpha} = (\underline{\alpha}_0, \dots, \underline{\alpha}_{p-1})$ respectively. In particular, we have*

$$\underline{a}_p(\underline{\beta}) = \underline{b}_p(\underline{\alpha}) = 0. \tag{3.13}$$

v) *Weighted sum of squares: for all x, we have*

$$x\underline{a}_p(x)^2 + (1-x)\underline{b}_p(x)^2 = 1. \tag{3.14}$$

The formula (3.15) shows that $\underline{a}_p$ and $\underline{b}_p$ correspond more precisely to the (shifted) third and fourth kind Chebyshev polynomials [12], table 18.3.1.

Proof. Property i) follows from a direct computation. To show the others we rely on the fact that if the polynomials $\hat{a}_p$ and $\hat{b}_p$ defined as the right hand sides of (3.12), namely

$$\begin{aligned}\hat{a}_p(x) &:= T_p(2x-1) - 2(1-x)U_p(2x-1)\\ \hat{b}_p(x) &:= T_p(2x-1) + 2xU_p(2x-1),\end{aligned}$$

satisfy iv) and v), then they coincide with $\underline{a}_p$ and $\underline{b}_p$. Indeed, (3.13) and (3.14) yield $\hat{a}_p(\underline{\alpha}_i)^2 = 1/\underline{\alpha}_i$ and $\hat{b}_p(\underline{\beta}_i)^2 = 1/(1-\underline{\beta}_i)$ for $i = 0, \dots, p$. Using $\hat{a}_p(\underline{\alpha}_p) = \hat{a}_p(1) = T_p(\cos 0) = 1$ and the interlacing of the nodes (3.11), we see that $\hat{a}_p(\underline{\alpha}_i)$ has the sign of $(-1)^{i+p}$. In particular, $\hat{a}_p$ is determined by (3.4) with $f = 1$ and hence coincides with $\underline{a}_p$. The case of $\hat{b}_p$ is similar, starting from $\hat{b}_p(\beta_0) = \hat{b}_p(0) = T_p(\cos\pi) = \cos(p\pi) = (-1)^p$. We are then left to show that $\hat{a}_p$ and $\hat{b}_p$ satisfy Properties iii) to v). For iii), we compute

$$\begin{aligned}\hat{a}_p(1-x) &= T_p(1-2x) - 2xU_p(1-2x)\\ &= (-1)^p T_p(2x-1) - 2x(-1)^{p-1}U_p(2x-1) = (-1)^p \hat{b}_p(x).\end{aligned}$$

For iv), we let θ be such that $\cos\theta = 2x-1$. Then $1-x = \sin(\theta/2)^2$ and

$$\hat{a}_p(x) = \cos(p\theta) - \frac{\sin(\theta/2)}{\cos(\theta/2)}\sin(p\theta) = \frac{1}{\cos(\theta/2)}\cos\left(\left(p+\frac{1}{2}\right)\theta\right) \tag{3.15}$$

holds for $\theta < \pi$. This shows that $\hat{a}_p$ has p distinct roots which coincide with the nodes $\beta_1, \dots, \beta_p \in]0,1[$. The case of $\hat{b}_p$ follows from Properties i) and iii). Turning to Property v), we compute

$$x\hat{a}_p(x)^2 + (1-x)\hat{b}_p(x)^2 = T_p(2x-1)^2 + 4\left(x-x^2\right)U_p(2x-1)^2.$$

Again with $\cos\theta = 2x-1$ we find $\sin^2\theta = 1-(2x-1)^2 = 4(x-x^2)$, and

$$T_p(2x-1)^2 + 4\left(x-x^2\right)U_p(2x-1)^2 = \cos^2(p\theta) + \sin^2\theta\left(\frac{\sin(p\theta)}{\sin\theta}\right)^2 = 1.$$

This shows (3.14) for the polynomials $\hat{a}_p$, $\hat{b}_p$, and the proof is complete. □

Definition 3.4 (Starting point of algorithm (3.8)). *Using the reference nodes (3.10), we set*

$$X^0 := (\underline{\alpha}, \underline{\beta}) = (\underline{\alpha}_0, \dots, \underline{\alpha}_{p-1}; \underline{\beta}_1, \dots, \underline{\beta}_p) \in I_p^2 \tag{3.16}$$

Some elementary formulas which are used in practical implementation are derived hereafter. By definition of the polynomials $\underline{a}_p$ and $\underline{b}_p$ we have (for all i)

$$\underline{a}_p(\underline{\alpha}_i) = \frac{(-1)^{i+p}}{\underline{\alpha}_i^{1/2}} \qquad \text{and} \qquad \underline{b}_p(\underline{\beta}_{-i}) = \frac{(-1)^{i+p}}{(1-\underline{\beta}_{-i})^{1/2}}. \tag{3.17}$$

For the derivatives we have the following result, which will also be useful in the subsequent analysis. The polynomials $\underline{a}_p$ and $\underline{b}_p$ defined satisfy

$$\underline{a}'_p(\alpha_i) = \frac{(-1)^{i+p+1}}{2\underline{\alpha}_i^{3/2}}, \qquad i = 0, \dots, p-1 \tag{3.18}$$

and similarly

$$\underline{b}'_p(\beta_i) = \frac{(-1)^{i+p}}{2(1-\underline{\beta}_{-i})^{3/2}}, \qquad i = 1, \dots, p. \tag{3.19}$$

These equalities are derived by differentiating the identity (3.14) and using the values of $\underline{a}_p$ and $\underline{b}_p$ on the inner nodes $\underline{\alpha}_i$ and $\underline{\beta}_{-i}$.

3.3. Study of the reference Jacobian matrix $J_p(X^0)$. The main result of this section is a proof that the reference Jacobian matrix $J_p(X^0)$ has a very simple structure and is non singular. We also provide an explicit formula.

LEMMA 3.5. *The reference Jacobian matrix defined by* (3.9) *has the form*

$$J_p(X^0) = \sqrt{f(0)} \begin{pmatrix} \nabla_\alpha b_p[\beta](\alpha) & \nabla_\beta b_p[\beta](\alpha) \\ \nabla_\alpha a_p[\alpha](\beta) & \nabla_\beta a_p[\alpha](\beta) \end{pmatrix} \Bigg|_{(\alpha,\beta)=X^0} = \sqrt{f(0)} \begin{pmatrix} D_\alpha & 0 \\ 0 & D_\beta \end{pmatrix} \tag{3.20}$$

where $D_\alpha = \operatorname{diag}\big(\underline{b}'_p(\underline{\alpha}_i) : i = 0, \dots, p-1\big)$ *and* $D_\beta = \operatorname{diag}\big(\underline{a}'_p(\underline{\beta}_{-i}) : i = 1, \dots, p\big)$ *are two diagonal matrices with non zero entries given by*

$$\begin{cases} \underline{a}'_p(\underline{\beta}_{-i}) = \dfrac{2p\cos(p\underline{\eta}_i)}{\cos\underline{\eta}_i + 1} + \dfrac{\sin(p\underline{\eta}_i)}{\sin\underline{\eta}_i}\left(2p + \dfrac{2}{\cos\underline{\eta}_i + 1}\right) & \text{for } i = 1, \dots, p \\ \underline{b}'_p(\underline{\alpha}_i) = \dfrac{2p\cos(p\underline{\theta}_i)}{\cos\underline{\theta}_i - 1} + \dfrac{\sin(p\underline{\theta}_i)}{\sin\underline{\theta}_i}\left(2p - \dfrac{2}{\cos\underline{\theta}_i - 1}\right) & \text{for } i = 0, \dots, p-1 \end{cases}$$

with $\underline{\eta}_i = \frac{(2(p-i)+1)\pi}{2p+1}$ *and* $\underline{\theta}_i = \frac{2(p-i)\pi}{2p+1}$.

Proof. The $\sqrt{f(0)}$ comes from the representation formulas (3.4-3.5) for $h = 0$ and

$$a_{p,0} = \sqrt{f(0)}\underline{a}_p \text{ and } b_{p,0} = \sqrt{f(0)}\underline{b}_p.$$

The other properties are proved as follows.

• Firstly, the diagonal form of the extra-diagonal blocks of $J_p(X^0)$ is clear, since for $j \neq i$, $a_p[\alpha](\beta_i)$ does not depend on β_j (and similarly $b_p[\beta](\alpha_i)$ does not depend on α_j). The values of the non-zero entries then follow by direct computation, using the formulas $\partial_x\{T_p(2x-1)\} = 2pU_p(2x-1)$ and $\partial_x\{U_p(2x-1)\} = \big((2x-1)U_p(2x-1) - pT_p(2x-1)\big)/\big(4x(1-x)\big)$. It is also easy to see that these values are non-zero: from Lemma 3.3 we know that $\underline{a}_p$ vanishes on the p nodes $\underline{\beta}_{-1}, \dots, \underline{\beta}_{-p}$, so that the Rolle theorem yields $p-1$ distinct roots for $\underline{a}'_p \in P_{p-1}$, one inside every interval of the form

$]\underline{\beta}_i, \underline{\beta}_{i+1}[$ with $0 \le i \le p-1$. Thus if $\underline{a}_p'$ would vanish at one of the $\underline{\beta}_i$'s then it would be identically zero, which is not possible due to the alternating sign of $\underline{a}_p$. The same argument shows that $\underline{b}_p'$ cannot vanish on a node $\underline{\alpha}_i$.

• Secondly, we need to study how $a_p[\alpha]$ defined by (3.4), namely

$$a_p[\alpha](x) = \sum_{0\le i\le p} (-1)^{i+p} \sqrt{\frac{f(h\alpha_i)}{\alpha_i}} \prod_{0\le j\ne i\le p} \frac{x-\alpha_j}{\alpha_i-\alpha_j},$$

varies as a function of the inner nodes $\alpha = (\alpha_0, \dots, \alpha_{p-1}) \in I_p$ (and similarly for $b_p[\beta]$). We focus on the dependency with respect to α_0 of the above quantity, which means that we freeze all the other variables and study

$$q(x, \alpha_0) := a_p[(\alpha_0, \underline{\alpha}_1, \dots, \underline{\alpha}_{p-1})](x) - a_p[(\underline{\alpha}_0, \underline{\alpha}_1, \dots, \underline{\alpha}_{p-1})](x).$$

One has the property that

$$q(\underline{\alpha}_i, \alpha_0) = (-1)^{i+p} \sqrt{\frac{f(h\underline{\alpha}_i)}{\underline{\alpha}_i}} - (-1)^{i+p} \sqrt{\frac{f(h\underline{\alpha}_i)}{\underline{\alpha}_i}} = 0 \quad \text{for } i = 1, \dots, p.$$

Since $q(\cdot, \alpha_0) \in P_p$, this yields $q(x, \alpha_0) = \lambda(\alpha_0) \prod_{1\le i\le p}(x - \underline{\alpha}_i)$ where λ can be obtained by identification at an arbitrary point. We take this point equal to α_0 and observe that since $h = 0$ in the reference Jacobian we can assume that $f = 1$. This gives

$$\begin{aligned} \lambda(\alpha_0) &= \frac{a_p[\alpha_0, \underline{\alpha}_1, \cdots, \underline{\alpha}_{p-1}](\alpha_0) - a_p[\underline{\alpha}_0, \underline{\alpha}_1, \cdots, \underline{\alpha}_{p-1}](\alpha_0)}{\prod_{1\le i\le p}(\alpha_0 - \underline{\alpha}_i)} \\ &= \frac{1}{\prod_{1\le i\le p}(\alpha_0 - \underline{\alpha}_i)} \left((-1)^p \sqrt{\frac{1}{\alpha_0}} - \underline{a}_p(\alpha_0) \right). \end{aligned} \tag{3.21}$$

By differentiating the representation $a_p[\alpha_0, \underline{\alpha}_1, \cdots, \underline{\alpha}_{p-1}](x) = a_p[\underline{\alpha}_0, \underline{\alpha}_1, \cdots, \underline{\alpha}_{p-1}](x) + \lambda(\alpha_0) \prod_{1\le i\le p}(x - \underline{\alpha}_i)$ with respect to α_0, we find

$$\frac{\partial}{\partial \alpha_0} a_p[\alpha_0, \underline{\alpha}_1, \cdots, \underline{\alpha}_{p-1}](x) = \lambda'(\alpha_0) \prod_{1\le i\le p} (x - \underline{\alpha}_i)$$

and using (3.21) we write

$$\begin{aligned} \lambda'(\alpha_0) =& \frac{\partial}{\partial \alpha_0} \left[\frac{1}{\prod_{1\le i\le p}(\alpha_0 - \underline{\alpha}_i)} \right] \left[(-1)^p \frac{1}{\alpha_0^{1/2}} - \underline{a}_p(\alpha_0) \right] \\ &+ \frac{1}{\prod_{1\le i\le p}(\alpha_0 - \underline{\alpha}_i)} \left[\frac{1}{2} (-1)^{p+1} \frac{1}{\alpha_0^{3/2}} - \underline{a}_p'(\alpha_0) \right]. \end{aligned}$$

Thanks to (3.17) and (3.18) this shows that $\lambda'(\underline{\alpha}_0) = 0$. Hence $\frac{\partial}{\partial \alpha_0} \left(a_p[\alpha](x) \right) \Big|_{\alpha = \underline{\alpha}} = 0$. By rotation of the indices we find the same result for the differentiation with respect to $\alpha_1, \cdots, \alpha_{p-1}$. Since the same method applies to b_p, we have finally proven that the two diagonal blocks of $J_p(X^0)$ indeed vanish, and this completes the proof. □

3.4. Node separation. To avoid $\Theta_{p,h}$ to become unbounded in the fixed point algorithm (3.8), one must guarantee that the approximated nodes stay away from each other. If note the whole construction fails apart. In the cubic case ($p = 1$) studied above this was guaranteed (for small values of h) by exhibiting a convex set F of $[0,1]^2$ that was preserved by the fixed point algorithm. To treat the general case $p \geq 2$ we define the separation set

$$I_{p,\varepsilon} = \{(x_1, \dots x_p) \subset [\varepsilon, 1-\varepsilon]^p : \varepsilon \leq x_i - x_{i-1} \ \text{ for } \ 1 \leq i \leq p\} \tag{3.22}$$

for some given $\varepsilon > 0$. Clearly we must have $\varepsilon \leq 1/(p+1)$ so that $I_{p,\varepsilon}$ is non empty. The separation set is used in the proof of the convergence Theorem in next section because it is needed to obtain conditions such that the sequence X^m stays inside $I_{p,\varepsilon}$ for some $\varepsilon > 0$. For technical reasons, it is also possible to "project" the iterates X^m inside $I_{p,\varepsilon}$ so that the iterates X^m are defined for all m. Additionally the separation operator defined below is used in our implementation, see the numerical section. The two steps of the construction of the separation operator are as follows:

- Given $\alpha = (\alpha_0, \dots, \alpha_{p-1}) \in \mathbb{R}^p$, a vector $\alpha^* = (\alpha_0^*, \dots, \alpha_{p-1}^*) \in [0,1]^p$ is first obtained by projecting $\alpha_i \mapsto \min(\max(\alpha_i, 0), 1) \in [0,1]$ and reordering the resulting values so that $0 \leq \alpha_0^* \leq \cdots \leq \alpha_{p-1}^* \leq 1$.
- A vector $\tilde{\alpha} \in I_{p,\varepsilon}$ is then obtained as follows. We define the differences $\Delta_i = \alpha_i^* - \alpha_{i-1}^*$ for $i = 0, \dots p$, where we have denoted $\alpha_{-1}^* = 0$ and $\alpha_p^* = 1$. By construction, we have $\Delta_i \geq 0$ for $0 \leq i \leq p$ and $\sum_{i=0}^p \Delta_i = 1$. We set

$$\widetilde{\Delta_i} := \frac{\max(\Delta_i, 2\varepsilon)}{\sum_{j=0}^p \max(\Delta_j, 2\varepsilon)}.$$

  We have $\sum_{j=0}^p \max(\Delta_j, 2\varepsilon) \leq \sum_{j=0}^p (\Delta_j + 2\varepsilon) = 1 + 2(p+1)\varepsilon$, hence

$$\widetilde{\Delta_i} \geq \frac{2\varepsilon}{1 + 2(p+1)\varepsilon} \geq \varepsilon$$

  by using that $\varepsilon \leq 1/(2(p+1))$. On the other hand, we still have $\sum_{i=0}^p \widetilde{\Delta_i} = 1$ and we can define

$$\tilde{\alpha}_0 := \widetilde{\Delta_1} \quad \text{ and } \quad \tilde{\alpha}_i := \widetilde{\Delta_i} + \tilde{\alpha}_{i-1} \quad \text{ for } \ i = 1, \dots, p-1.$$

  By construction the resulting vector $\tilde{\alpha}$ is indeed in $I_{p,\varepsilon}$.

DEFINITION 3.6. *For $X = (\alpha, \beta) \in \mathbb{R}^{2p}$, we define $S_{p,\varepsilon} X = (\tilde{\alpha}, \tilde{\beta})$ by applying the above construction to α and β separately.*

If $X = (\alpha, \beta) \in I_{p,2\varepsilon}$ then this process does not change the nodes, that is $S_{p,\varepsilon} X = X$. For a general $X \in \mathbb{R}^{2p}$, one obtains that $S_{p,\varepsilon}^2 X = S_{p,\varepsilon}(S_{p,\varepsilon} X) = S_{p,\varepsilon} X$. So $S_{p,\varepsilon}$ is a projector.

ALGORITHM 3.2 (Simplified Newton-Raphson algorithm with node separation). *The fixed point algorithm* (3.8) *with guaranteed node separation reads*

$$X^{m+1} = S_{p,\varepsilon} G_h(X^m) \quad \text{ where } \quad G_h(X) := X - J_p(X^0)^{-1} \Theta_{p,h}(X) \tag{3.23}$$

where the $2p \times 2p$ Jacobian matrix of $\Theta_{p,0}$ is $J_p(X^0) = \sqrt{f(0)} \begin{pmatrix} D_\alpha & 0 \\ 0 & D_\beta \end{pmatrix}$ and the diagonal matrices D_α and D_β are defined in Lemma 3.5.

In practice, ε can be taken very small. Clearly, to avoid spoiling the convergence process it should be significantly smaller than the minimal distance between the reference nodes (3.10). In all the sequel we will consider that it has a fixed value depending only on n, such that the following condition holds,

$$\varepsilon \leq \min\left(\frac{1}{2(p+1)}, \min_{1\leq i\leq 2p+1}\Big(\frac{\underline{\gamma}_i - \underline{\gamma}_{i-1}}{4}\Big)\right) \tag{3.24}$$

where we have used a common notation $\underline{\gamma}_i := \frac{1}{2}\left[1-\cos\left(\frac{i\pi}{2p+1}\right)\right]$, $i=0,\dots,2p+1$, for the $n+1$ reference nodes in $[0,1]$. The first bound is required for the definition of the separation operator $S_{p,\varepsilon}$, and the other one guarantees that $X^0 = (\underline{\alpha}, \underline{\beta})$ is inside $(I_{p,4\varepsilon})^2$, so that

$$B(X^0,\varepsilon) \subset (I_{p,2\varepsilon})^2. \tag{3.25}$$

In fact, (3.24) guarantees the stronger property that the $n-1$ inner nodes $\underline{\gamma} = (\underline{\gamma}_1,\dots,\underline{\gamma}_{2p})$ are in the set $I_{2p,4\varepsilon}$, hence

$$(\alpha_0,\dots,\alpha_{p-1};\beta_1,\dots,\beta_p) \in B(X^0,\varepsilon) \implies (\alpha_0,\beta_1,\dots,\alpha_{p-1},\beta_p) \in I_{2p,2\varepsilon}. \tag{3.26}$$

3.5. Convergence. The proof of convergence is a generalization of the case $p=1$. The main condition is that h must be taken sufficiently small which means in view of (3.8) or (3.23) that the algorithm is very close to a true Newton-Raphson algorithm $X^{m+1} = X^m - J_p(X^0)^{-1}\Theta_{p,0}(X^m)$ where the function is evaluated for $h=0$. It is therefore not a surprise that, for small $h>0$, the simplified Newton-Raphson algorithm inherits by continuity of a degenerated version of the very strong contraction properties of a true Newton-Raphson algorithm: it results in the geometric convergence rate evidenced in (3.27) below. This rate can be interpreted in two ways. Since $0<h\leq h_0$, one obtains a rate of convergence better than $\approx \frac{1}{2^m}$. For a given h_0, it can be viewed as $O(h^m)$ convergence: in this second regime, the exact value of h_0 does not matter.

THEOREM 3.1 (Convergence). *Let $f\in W^{1,\infty}(0,1)$ satisfy (1.1), and $\varepsilon=\varepsilon(n)>0$ be such that (3.24) holds. Then there exist $h_0>0$ such that for all $0\leq h\leq h_0$ the following properties hold:*

- ***i)*** *the separation operator is not active, that is the algorithms (3.8) and (3.23) with starting point (3.16) compute the same iterates $(X^m)_{m\geq 0}$ which belong to the set $(I_{p,2\varepsilon})^2 \subset]0,1[^{2p}$*
- ***ii)*** *the sequence $(X^m)_{m\geq 0}$ converges towards a fixed point of G_h in the ball $B(X^0,\varepsilon)$, $X_h^\infty = (\alpha_0^\infty,\dots,\alpha_{p-1}^\infty;\beta_1^\infty,\dots\beta_p^\infty)$, consisting of interlaced nodes bounded away from each other and from the end nodes, see (3.22),*

$$(\alpha_0^\infty,\beta_1^\infty,\dots,\alpha_{p-1}^\infty,\beta_p^\infty) \in I_{2p,2\varepsilon}$$

- ***iii)*** *the error estimate holds for all $m\geq 0$*

$$\|X_h^\infty - X^m\| \leq 2\Big(\frac{h}{2h_0}\Big)^{m+1}. \tag{3.27}$$

Again the proof relies on a technical lemma that generalizes Lemma 2.6.

LEMMA 3.7. *Let $f\in W^{1,\infty}(0,1)$ satisfy (1.1), and $\varepsilon>0$ be such that (3.24) holds. Then there exist a constant $C_{p,\varepsilon}^*$ independent of $h\in[0,1]$ such that*

i) *the Jacobian matrix* ∇G_h *satisfies*

$$\|\nabla G_h(X)\| \le C^*_{p,\varepsilon}(h + \|X - X^0\|), \qquad X \in (I_{p,\varepsilon})^2, \tag{3.28}$$

ii) *the derivative of* G_h *with respect to* h *satisfies*

$$\|\partial_h G_h(X)\| \le C^*_{p,\varepsilon}, \qquad X \in (I_{p,\varepsilon})^2. \tag{3.29}$$

Proof. The proof is very similar to the one of (2.14)-(2.15). Using (3.23) we compute

$$\nabla G_h(X) = I - \nabla\Theta_{p,0}(X^0)^{-1}\nabla\Theta_{p,h}(X) \tag{3.30}$$

and from the expressions (3.4)-(3.6) we observe that $\nabla\Theta_{p,h}(X)$, and hence $\nabla G_h(X)$, can be written under the form $\Phi(h, X) + h\Psi(h, X)$ where Φ (resp. Ψ) involves values of f (resp. f') and is Lipschitz (resp. bounded) on $[0, 1] \times (I_{p,\varepsilon})^2$, with Lipschitz constant (resp. L^∞ norm) depending on f, p and ε, but not on h. From (3.30) we also see that $\nabla G_0(X^0) = 0$. Using that $(I_{p,\varepsilon})^2$ is convex, this gives

$$\begin{aligned}\|\nabla G_h(X)\| &= \|\nabla G_h(X) - \nabla G_0(X^0)\| \\ &\le \|\Phi(h, X) - \Phi(0, X^0)\| + h\|\Psi(h, X)\| \\ &\le C^*_{p,\varepsilon}(h + \|X - X^0\|)\end{aligned}$$

for all $(h, X) \in [0, 1] \times (I_{p,\varepsilon})^2$, with a constant $C^*_{p,\varepsilon}$ independent of h. The last claim is again straightforward (with another constant), using the fact that f is Lipschitz and bounded away from 0. □

Proof. [Proof of Theorem 3.1]

Let $h_0 := (2C^*_{p,\varepsilon})^{-1} \min\left(\varepsilon, (1 + 2C^*_{p,\varepsilon})^{-1}\right)$. Then for $h \le h_0$ the condition (3.24) gives $2C^*_{p,\varepsilon}h \le \varepsilon$ and, using (3.25),

$$B(X^0, 2C^*_{p,\varepsilon}h) \subset B(X^0, \varepsilon) \subset (I_{p,2\varepsilon})^2. \tag{3.31}$$

Like in Theorem 2.1, it shows that $G_h(B(X^0, 2C^*_{p,\varepsilon}h)) \subset B(X^0, 2C^*_{p,\varepsilon}h)$. In particular, for $h \le h_0$ all the iterates X^m are in this ball and also in $(I_{p,2\varepsilon})^2$ so that the separation operator $S_{p,\varepsilon}$ has no effect. Since $I_{p,2\varepsilon} \subset I_{p,\varepsilon}$, estimate (3.28) holds and gives $\|\nabla G_h(X)\| \le \frac{h}{2h_0}$. This shows that G_h is contractant on $B(X^0, 2C^*_{p,\varepsilon}h)$ so that the fixed point theorem of Picard applies: G_h has a unique fixed point X^∞_h in the ball. Writing $e^m_h = \|X^m - X^\infty_h\|$ we have $e^{m+1}_h = \|G_h(X^m) - G_h(X^\infty_h)\| \le \frac{h}{2h_0}e^m_h$ so that $e^m_h \le \left(\frac{h}{2h_0}\right)^m e^0_h$. Estimate (3.27) follows by noticing that $e^0_h = \|X^0 - X^\infty_h\| \le 2C^*_{p,\varepsilon}h \le \frac{h}{h_0}$ using the above considerations. □

An easy corollary of the above convergence theorem 3.1 is a local version of the Lukács Theorem which specifies Theorem 1.3.

COROLLARY 3.8. *Assume* $f \in P^+_n$ *and* $f > 0$ *on* $[0, 1]$. *Then there exist* $h_0 > 0$ *such that for* $0 \le h \le h_0$, $f(hx) = xa_{p,h}(x)^2 + (1-x)b_{p,h}(x)^2$ *with* $a_{p,h} = a_{p,h}[\alpha^\infty]$ *and* $b_{p,h} = b_{p,h}[\beta^\infty]$ *given by* (3.4)-(3.5), *using the nodes* $(\alpha^\infty, \beta^\infty) = X^\infty_h$ *corresponding to a fixed point of* G_h *in* $(I_p)^2$.

Proof. Indeed both sides of the equality are equal at $n + 1$ different points which are $0 = \beta^\infty_0 < \alpha^\infty_0 < \cdots < \beta^\infty_p < \alpha^\infty_p = 1$. Since it is an equality between polynomials of degree $\le n$, it yields the claim. □

3.6. Accuracy of the approximate interpolants in P_n^+. Proceeding like in Section 2.3, we denote $(\alpha^m, \beta^m) := X^m$ the m-th approximation of $(\alpha^\infty, \beta^\infty) := X_h^\infty$ in the iterative Newton scheme (3.23), (3.16), and since $X^m \in (I_{p,\varepsilon})^2$ thanks to the speration operator we can define polynomials in P_p following (3.4)-(3.5), namely

$$a_{p,h}^m = a_{p,h}[\alpha^m](x) = \sum_{0\le i\le p} (-1)^{i+p} \sqrt{\frac{f(h\alpha_i^m)}{\alpha_i^m}} \prod_{0\le j\ne i\le p} \frac{x-\alpha_j^m}{\alpha_i^m - \alpha_j^m} \tag{3.32}$$

and

$$b_{p,h}^m = b_{p,h}[\beta^m](x) = \sum_{0\le i\le p} (-1)^{i+p} \sqrt{\frac{f(h\beta_i^m)}{1-\beta_i^m}} \prod_{0\le j\ne i\le p} \frac{x-\beta_j^m}{\beta_i^m - \beta_j^m}. \tag{3.33}$$

and let

$$p_n^m(x) := x a_{p,h}^m(x)^2 + (1-x) b_{p,h}^m(x)^2 \tag{3.34}$$

be the corresponding approximation to f_h from P_n^+ (the dependence of p_n^m on h is left implicit for simplicity). The following result specifies Theorem 1.2 for odd degrees.

THEOREM 3.2 (Optimal h convergence). *Let $f \in W^{q,\infty}(]0,1[)$, $1 \le q \le n+1$, satisfy* (1.1)*, and let $h_0 > 0$ be given by Theorem 3.1. Then for all $0 \le h \le h_0$ and all $m \ge 0$, the polynomial* (3.34) *satisfies*

$$\|p_n^m - f_h\| \le C h^{\min(q, 2(m+1))} \tag{3.35}$$

for a constant C independent of h.

Proof. The result essentially follows by inspecting the values of p_n^m on 0, α^m, β^m and 1. On the extremal nodes one has $p_n^m(0) = f_h(0)$ and $p_n^m(1) = f_h(1)$. On the interior ones one has $p_n^m(\alpha_i^m) = f_h(\alpha_i^m) + (1-\alpha_i^m) b_{p,h}^m(\alpha_i^m)^2$ and $p_n^m(\beta_i^m) = \beta_i^m a_{p,h}^m(\beta_i^m)^2 + f_h(\beta_i^m)$. Since $h \le h_0$ we know from Theorem 3.1 that

$$(\alpha^m, \beta^m) \in (I_{p,\varepsilon})^2, \quad m \ge 0, \tag{3.36}$$

and using also that f Lipschitz and bounded away from 0, we see that $b_{p,h}^m(x) = b_{p,h}[\beta^m](x)$ is Lipschitz as a function of $(x, \beta^m) \in [0,1] \times I_{p,\varepsilon}$. In particular, we have (for all i)

$$\begin{aligned} |p_n^m(\alpha_i^m) - f_h(\alpha_i^m)| \le b_{p,h}^m(\alpha_i^m)^2 &= |b_{p,h}^m(\alpha_i^m) - b_{p,h}^\infty(\alpha_i^\infty)|^2 \\ &\le \big(|b_{p,h}^m(\alpha_i^m) - b_{p,h}^\infty(\alpha_i^m)| + |b_{p,h}^\infty(\alpha_i^m) - b_{p,h}^\infty(\alpha_i^\infty)|\big)^2 \\ &\le C\big(\|\beta^m - \beta^\infty\| + \|\alpha^m - \alpha^\infty\|\big)^2, \end{aligned}$$

where we readily observe the squaring of the right hand side which is the reason of the doubling of the rate of convergence. The same bound holds for $|p_n^m(\beta_i^m) - f_h(\beta_i^m)|$. Let us denote by $\tilde{p}_n^m$ the polynomial in P_n that interpolates f_h on the $n+1$ nodes of $\{0,1\} \cup \{\alpha_0^m, \dots, \alpha_{p-1}^m\} \cup \{\beta_1^m, \dots, \beta_p^m\}$ which are distinct and bounded away from each other by at least 2ε according to (3.31) and (3.26). Standard polynomial interpolation estimates yield

$$\begin{aligned} \|p_n^m - f_h\| &\le \|p_n^m - \tilde{p}_n^m\| + \|\tilde{p}_n^m - f_h\| \\ &\le C\Big(\max_{0\le i\le p-1} |(p_n^m - \tilde{p}_n^m)(\alpha_i^m)| + \max_{1\le i\le p} |(p_n^m - \tilde{p}_n^m)(\beta_i^m)| + \|f_h^{(q)}\| \Big) \\ &\le C\big(\|X^m - X_h^\infty\|^2 + h^q \|f^{(q)}\|\big) \end{aligned}$$

with a constant depending on f and n. Using (3.27) this concludes the proof. ∎

4. Extension to the case $n = 2p$. For even degrees, the results are essentially the same as in previous section. The main difference comes from the fact that now the polynomials $a_p \in P_p$ and $b_{p-1} \in P_{p-1}$ do not have the same degree a priori. Since the core parts of the proofs do not differ, we just state the results without further justification (the interested reader can easily recover the arguments by comparison with the material in the previous section). However the properties in terms of Chebyshev polynomials and definition of the starting point of the simplified algorithm need to be precisely stated because they are different.

A sufficient criterion for positive interpolation. For even degrees n the sufficient criterion of proposition 3.1 generalizes without difficulty.

PROPOSITION 4.1. *Let $f \in W^{1,\infty}(]0,1[)$ satisfy (1.1), and let $h \geq 0$. Let $a_{p,h} \in P_p$, $b_{p-1,h} \in P_{p-1}$ and the $2p-1$ nodes $0 = \alpha_0 < \alpha_1 < \ldots < \alpha_{p-1} < \alpha_p = 1$ and $0 < \beta_1 < \cdots < \beta_p < 1$ in $(0,1)$ are such that*

$$\begin{aligned} a_{p,h}(\beta_i) &= 0 \qquad \text{for} \quad 1 \leq i \leq p, \\ b_{p-1,h}(\alpha_i) &= 0 \qquad \text{for} \quad 1 \leq i \leq p-1, \end{aligned} \tag{4.1}$$

and such that

$$\begin{aligned} a_{p,h}(\alpha_i) &= (-1)^{i+p}\sqrt{f(h\alpha_i)}, \qquad \text{for} \quad 0 \leq i \leq p, \\ b_{p-1,h}(\beta_i) &= (-1)^{i+p}\sqrt{\frac{f(h\beta_i)}{\beta_i(1-\beta_i)}} \qquad \text{for} \quad 1 \leq i \leq p. \end{aligned} \tag{4.2}$$

Then one has that: a) $0 = \alpha_0 < \beta_1 < \alpha_1 < \cdots < \beta_p < \alpha_p = 1$; b) the polynomial $p_n(x) = a_{p,h}(x)^2 + x(1-x)b_{p-1,h}(x)^2 \in P_n^+$ interpolates $f_h = f(h\,\cdot)$ on the $n+1$ nodes $\alpha_0, \beta_1, \cdots, \beta_p, \alpha_p$.

Simplified Newton Algorithms. The principle of the algorithms (3.8) and (3.23) is adapted in a straightforward manner. For

$$(\alpha, \beta) = (\alpha_1, \ldots, \alpha_{p-1};\, \beta_1, \ldots, \beta_p) \in I_{p-1} \times I_p,$$

we let $a_{p,h}[\alpha]$ and $b_{p-1,h}[\beta]$ be the polynomials solving the interpolation problems (4.2), that is

$$a_{p,h}[\alpha](x) = \sum_{0 \leq i \leq p} (-1)^{i+p}\sqrt{f(h\alpha_i)} \prod_{0 \leq j \neq i \leq p} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \tag{4.3}$$

and

$$b_{p-1,h}[\beta](x) = \sum_{1 \leq i \leq p} (-1)^{i+p}\sqrt{\frac{f(h\beta_i)}{\beta_i(1-\beta_i)}} \prod_{1 \leq j \neq i \leq p} \frac{x - \beta_j}{\beta_i - \beta_j}. \tag{4.4}$$

Let us define the function $\Gamma_{p,h} : I_{p-1} \times I_p \longrightarrow \mathbb{R}^{2p-1}$ by

$$\Gamma_{p,h}(\alpha, \beta) = \left(b_{p-1,h}[\beta](\alpha_1), \ldots, b_{p-1,h}[\beta](\alpha_{p-1}), a_{p,h}[\alpha](\beta_1), \ldots, a_{p,h}[\alpha](\beta_p)\right). \tag{4.5}$$

As in section 3 the sufficient criterion of Proposition 4.1 applies as soon as $(\alpha, \beta) \in I_{p-1} \times I_p$ satisfies

$$\Gamma_{p,h}(\alpha, \beta) = 0. \tag{4.6}$$

and we introduce, starting from $X^0 \in I_{p-1} \times I_p$, the following simplified Newton-Raphson algorithms

$$X^{m+1} = X^m - [\nabla \Gamma_{p,0}(X^0)]^{-1} \Gamma_{p,h}(X^m) \tag{4.7}$$

where $\nabla \Gamma_{p,0}(X^0) \in \mathbb{R}^{2p-1 \times 2p-1}$ is the Jacobian matrix of $\Gamma_{p,h}$ with $h = 0$, evaluated at the starting point X^0. The algorithm with separation operator recasts as

$$X^{m,1} = X^m - [\nabla \Gamma_{p,0}(X^0)]^{-1} \Gamma_{p,h}(X^m) \qquad X^{m+1} = S_{p,\varepsilon} X^{m,1}. \tag{4.8}$$

Using the compact notation $\Gamma_{p,h}(\alpha, \beta) = (b_{p-1,h}[\beta](\alpha), a_{p,h}[\alpha](\beta))$, the Jacobian matrix takes the 2×2 block form

$$\nabla \Gamma_{p,0}(X^0) = \left(\begin{array}{cc} \nabla_\alpha b_{p-1,h}[\beta](\alpha) & \nabla_\beta b_{p-1,h}[\beta](\alpha) \\ \nabla_\alpha a_{p,h}[\alpha](\beta) & \nabla_\beta a_{p,h}[\alpha](\beta) \end{array} \right) \Bigg|_{(\alpha,\beta)=X^0}. \tag{4.9}$$

Definition of the starting point X^0. Again here, we define the starting point X^0 thanks to the use of the Chebyshev polynomial $(T_p, U_p) \in P_p \times P_{p-1}$. We seek two polynomials $\underline{a}_p, \underline{b}_{p-1} \in P_p$ such that

$$\underline{a}_p(x)^2 + x(1-x)\underline{b}_{p-1}(x)^2 = 1 \quad \text{ for all } \quad x \in [0,1].$$

The following Lemma can be prooved in the same way as Lemma 3.3

LEMMA 4.2. *Given $p \in \mathbb{N}$ let $\underline{a}_p(x) = T_p(2x-1)$, $\underline{b}_{p-1}(x) = 2U_p(2x-1)$ and*

$$\underline{\alpha}_i := \frac{1}{2}\left[1 - \cos\left(\frac{i\pi}{p}\right)\right] \ i = 0, \ldots, p, \quad \underline{\beta}_i := \frac{1}{2}\left[1 - \cos\left(\frac{(2i-1)\pi}{2p}\right)\right], \ i = 1, \ldots, p.$$

We have the following properties.

- ***i)*** *Interlacing of the nodes: we have $0 = \underline{\alpha}_0 < \underline{\beta}_1 < \underline{\alpha}_1 < \cdots < \underline{\beta}_p < \underline{\alpha}_p = 1$.*
- ***ii)*** *Symmetry: for all x, we have $\underline{a}_p(1-x) = (-1)^p a_p(x)$ and $\underline{b}_{p-1}(1-x) = (-1)^{p-1}\underline{b}_{p-1}(x)$.*
- ***iii)*** *Root property: $\underline{a}_p$ (respectively $\underline{b}_{p-1}$) has p (respectively $p-1$) simple roots in $]0,1[$, which coincide with $\underline{\beta} = (\underline{\beta}_1, \ldots, \underline{\beta}_p)$ and $\underline{\alpha} = (\underline{\alpha}_1, \ldots, \underline{\alpha}_{p-1})$ respectively. In particular, we have $\underline{a}_p(\underline{\beta}) = \underline{b}_{p-1}(\underline{\alpha}) = 0$.*
- ***iv)*** *Weighted sum of squares: for all x, we have $\underline{a}_p(x)^2 + x(1-x)\underline{b}_{p-1}(x)^2 = 1$.*
- ***v)*** *the polynomials $\underline{a}_p$ and $\underline{b}_p$ correspond to the ones defined according to (4.3)-(4.4) with a constant function $f = 1$.*

COROLLARY 4.3. *The polynomials $\underline{a}_p$ and $\underline{b}_p$ satisfy $\underline{a}'_p(\alpha_i) = 0$ for $i = 0, \ldots, p$ and similarly $\underline{b}'_p(\beta_i) = (-1)^{i+p+1} \frac{(1-2\beta_i)}{[\beta_i(1-\underline{\beta}_i)]^{3/2}}$ for $i = 1, \ldots, p$.* We may then set, as starting point of the algorithm (3.8),

$$X^0 := (\underline{\alpha}, \underline{\beta}) = (\underline{\alpha}_1, \ldots, \underline{\alpha}_{p-1}; \underline{\beta}_1, \ldots, \underline{\beta}_p) \in I_{p-1} \times I_p \tag{4.10}$$

using the reference nodes (3.10). The main result of this section is a proof that the reference Jacobian matrix $\nabla \Gamma_{p,0}(X^0)$ has a very simple structure and is non singular. We also provide an explicit formula.

LEMMA 4.4. *The reference Jacobian matrix defined by (3.9) has the form*

$$\nabla \Gamma_{p,0}(X^0) = \sqrt{f(0)} \left(\begin{array}{cc} \nabla_\alpha b_{p-1}[\beta](\alpha) & \nabla_\beta b_{p-1}[\beta](\alpha) \\ \nabla_\alpha a_p[\alpha](\beta) & \nabla_\beta a_p[\alpha](\beta) \end{array} \right) \Bigg|_{(\alpha,\beta)=X^0}$$

that is $\nabla\Gamma_{p,0}(X^0) = \sqrt{f(0)}\begin{pmatrix} D_\alpha & 0 \\ 0 & D_\beta \end{pmatrix}$ *where* $D_\alpha = \mathrm{diag}\left(\underline{b}'_p(\underline{\alpha}_i) : i = 1, \dots, p-1\right)$ *and* $D_\beta = \mathrm{diag}\left(\underline{a}'_p(\underline{\beta}_i) : i = 1, \dots, p\right)$ *are diagonal matrices given by*

$$\begin{cases} \underline{a}'_p(\underline{\beta}_i) = \dfrac{2p\sin(p\underline{\xi}_i)}{\sin(\underline{\xi}_i)} \neq 0 \quad \textit{for } i = 1, \dots, p \\ \underline{b}'_p(\underline{\alpha}_i) = 4\dfrac{\sin(p\underline{\gamma}_i)}{\sin(\underline{\gamma}_i)}\dfrac{\cos(\underline{\gamma}_i)}{1-\cos(\underline{\gamma}_i)^2} - 4p\dfrac{\cos(p\underline{\gamma}_i)}{1-\cos(\underline{\gamma}_i)^2} \neq 0 \quad \textit{for } i = 0, \dots, p \end{cases}$$

with $\underline{\xi}_i = \frac{(2(p-i)+1)\pi}{2p}$ *and* $\underline{\gamma}_i = \frac{(p-i)\pi}{p}$. The convergence estimates for even degrees $n = 2p$ take the same form than for odd degrees, which allows to state Theorem 1.2 as a general result.

5. Numerical illustrations. We provide implementation details then present results which either validate the different algorithms and convergence estimates or, and we think this is much more valuable, show that the range of parameters for which the algorithms can be used is much larger than what is predicted by the theory.

5.1. Implementation details. The practical implementation of the algorithms described above requires elementary modifications, so as to run smoothly even if the hypothesis of the convergence theorems are not entirely satisfied. We describe these modifications for the simplified Newton-Raphson algorithm (3.8) in the case of odd degrees $n = 2p + 1$

$$X^{m+1} = G_h(X^m) \quad \text{where} \quad G_h(X) := X - J_p(X^0)^{-1}\Theta_{p,h}(X).$$

The case of cubic (2.7) and even degrees (4.10) poses no real difficulties and is left to the reader. In practice, the modified loop writes

$$X^{m+1} = S_{p,\varepsilon}\widehat{G}_h(X^m) \quad \text{where} \quad \widehat{G}_h(X) := X - J_{p,\varepsilon}(X^0)^{-1}\Theta_{p,h,\varepsilon}(X). \tag{5.1}$$

There are three ingredients in this formula which all of them contribute to get a non singular algorithm up to a small truncation error of order $\varepsilon > 0$.

- The first one is the separation operator $S_{p,\varepsilon}$ introduced in Definition 3.6.
- The second one is a new Jacobian replacing $J_p(X^0)$, see (3.20),

$$J_{p,\varepsilon}(X^0) := \sqrt{f^m_\varepsilon}\begin{pmatrix} D_\alpha & 0 \\ 0 & D_\beta \end{pmatrix} \qquad (\text{matrices } D_{\alpha,\beta} \text{ defined in Lemma 3.5})$$

where f^m_ε stands for a maximal value of f evaluated at iteration m. A convenient choice writes $f^m_\varepsilon := \max\left(\max_{z\in\mathcal{V}^m} f(z), \varepsilon\right) \geq \varepsilon > 0$ where $\mathcal{V}^m = \{X^m_i\}_{1\leq i\leq 2p}$ is the set of coordinates of X^m (the α^m_i and β^m_i).

- The third ingredient is based on the introduction of the offset $\varepsilon > 0$ in the interpolation polynomials which are now

$$a_{p,h,\varepsilon}[\alpha](x) = \sum_{0\leq i\leq p}(-1)^{i+p}\sqrt{\frac{\max\left(f(h\alpha_i), \varepsilon\right)}{\alpha_i}}\prod_{0\leq j\neq i\leq p}\frac{x-\alpha_j}{\alpha_i-\alpha_j} \tag{5.2}$$

and

$$b_{p,h,\varepsilon}[\beta](x) = \sum_{0\leq i\leq p}(-1)^{i+p}\sqrt{\frac{\max\left(f(h\beta_i), \varepsilon\right)}{1-\beta_i}}\prod_{0\leq j\neq i\leq p}\frac{x-\beta_j}{\beta_i-\beta_j}. \tag{5.3}$$

We note that this offset is not necessary to define the polynomials (5.2) and (5.3), but it is needed for them to oscillate. The new function $\Theta_{p,h,\varepsilon} : I_p^2 \longrightarrow \mathbb{R}^{2p}$ is

$$\Theta_{p,h,\varepsilon}(\alpha,\beta) = (b_{p,h,\varepsilon}[\beta](\alpha_0),\ldots,b_{p,h,\varepsilon}[\beta](\alpha_{p-1}),a_{p,h,\varepsilon}[\alpha](\beta_1),\ldots,a_{p,h,\varepsilon}[\alpha](\beta_p))\,.$$

Our best implementation of $\Theta_{p,h,\varepsilon}$ is based on the Newton divided differences method.

5.2. Interpolation. The application to interpolation problems is presented.

5.2.1. Cubic nodes. We illustrate in table 5.1 the convergence of the fixed point Algorithm 2.1 which computes the interpolation nodes for the cubic case. We consider the function

$$f(x) = 0.5 + |x - 0.5| \text{ for } x < 0.5, \quad f(x) = 0.5 + \frac{1}{2}|x - 0.5| \text{ for } 0.5 \le x. \tag{5.4}$$

The numerical values of the nodes $X^m = (\alpha^m, \beta^m)$ are given in function of the iteration marker m. One observes very fast convergence of the sliding interpolation points to their limit value. This convergence behavior is in some sense better than the one predicted by Theorem 2.1 because $h = 1$ in this numerical simulation.

m	α^m	β^m
0	0.250000,	0.750000
1	0.290569	0.747017
2	0.290678	0.747013
...	0.290678	0.747013

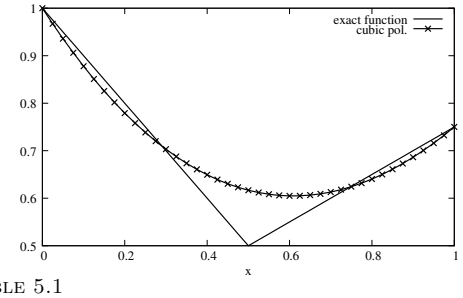

TABLE 5.1
Algorithm 2.1: convergence of the sliding interpolation points for the function (5.4).

The next series of numerical results illustrate the accuracy estimate (2.19) of Theorem 2.2 for the positive polynomial approximation of a given function. For different values of h, we consider the functions

$$f_h(x) = \frac{1}{1 - hx}, \quad 0 \le x \le 1, \quad h = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \cdots \tag{5.5}$$

The results are given in Table 5.2. The relative $L^\infty(0,1)$ error between f_h and its approximation p_3^m is provided as a function of the iteration number m which shows up in estimate (2.19). For $m = 0$, the accuracy is second order. For $m = 1$ and beyond, the accuracy is fourth order. It cannot be greater than fourth order since this is optimal for the approximation with cubic polynomials.

5.2.2. General order Newton-Raphson algorithm . We illustrate the efficiency of the general order Newton-Raphson algorithm (3.8) (for even degrees it is given by equation (4.8)). The initial point is (3.16). The first iterates of the algorithm are always well defined since the Jacobian $J_p(X^0)$ is a non singular matrix by Lemma 3.5. The sliding nodes are well separated provided h is small enough, as explained in Section 3.4 and Theorem 3.1. However we have observed in many numerical experiments excellent convergence properties even for $h = 1$.

h	$m=0$		$m=1$		$m=2$
1/2	0.0205988		0.0024350220		0.0024422952
1/4	0.0044347	4.64	0.0000881270	27.6	0.0000893219
1/8	0.0010400	4.26	0.0000045399	19.4	0.0000046098
1/16	0.0002519	4.13	0.0000002579	17.6	0.0000002619
1/32	0.0000619	4.07	0.0000000153	16.8	0.0000000156
order		≈ 2		≈ 4	

TABLE 5.2

Relative L^∞ errors between the function (5.5) and its approximated cubic interpolant, with the reduction factors. The observed convergence order is in accordance with Theorem 2.2. The last column with $m=2$ shows no improvement with respect to $m=1$, as expected.

The first series of plots compare the approximation of the (Runge) function

$$R(x) = \frac{1}{(1+25(2x-1)^2} \tag{5.6}$$

for interpolation with positive polynomials of degree $n = 7k$ for $k = 1, 2, 3, 4$. The function R and its positive interpolant are represented on the same plot, with in bullets the position of the interpolation points. The number of iterations is systematically the same $m = 10$. One observes stability and convergence of the interpolation points as n increases, either for even or odd degrees.

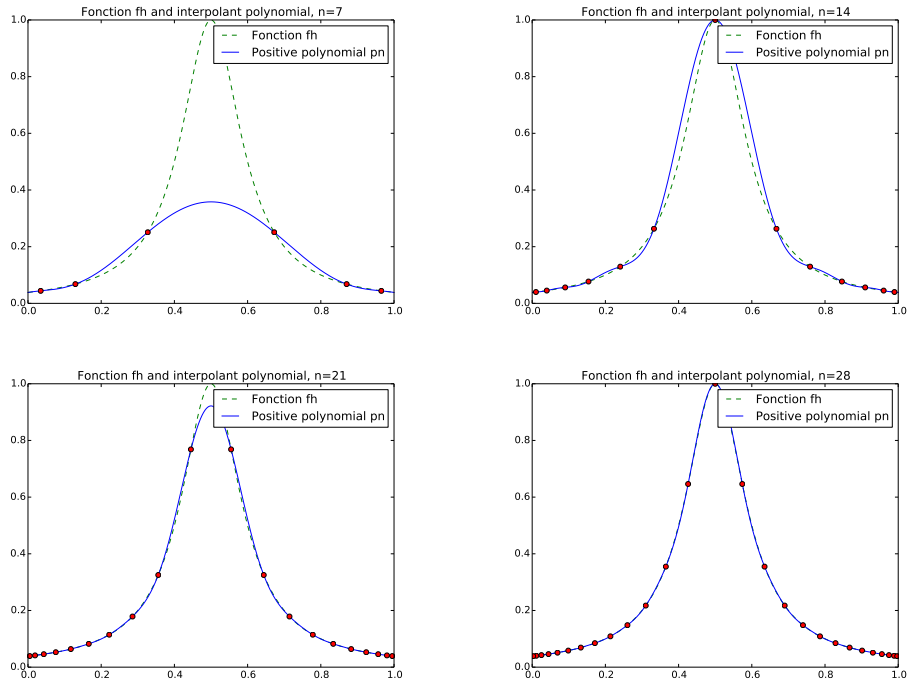

FIG. 5.1. *Degree $n = 7, 14, 21, 28$. The (Runge) function (5.6) is in dashed lines. The positive interpolation is the continuous line. Interpolation points are represented in bullets $m = 10$.*

The second series of experiments illustrates the result of the local Lukács Theo-

rem 1.3. We take a function $f \in P_{17}^+$ which is now polynomial

$$f(x) = 10^5 x^{10}(1-x)^7 + 0.01 \tag{5.7}$$

and very close to zero at the boundaries. We use the general order Newton-Raphson algorithm (3.8) until convergence. The results are given in Table 5.3. The approximation is exact to machine accuracy for $p = 8$ which yields the exact degree of f since $n = 2p + 1 = 17$. The curves for $0 \leq p \leq 8$ (not shown here) indicate that the approximation is always non negative, which is a property of the method.

p	0	1	2	3	4	5	6	7	8
rel. $L^\infty \approx$ error	1.	0.8	0.3	0.2	0.1	0.05	0.03	0.003	$\varepsilon_{mach.}$

TABLE 5.3

Convergence of the sliding interpolation points to their limit for the polynomial function f (5.7). The errors in L^∞ norm are provided in the table above the figure which displays in bold both f and its exact approximation for $p = 8$.

The convergence order of estimate (1.3) in Theorem 1.2, which deals with non negative polynomial approximation of high order, is illustrated in Table 5.4, using the same method than in Table 5.2. The objective function is f_h with f given in (5.7). Here we consider odd degrees $n = 2p$, and to obtain the optimal accuracy with the minimal algorithmic cost, we equate p the degree of a_p and b_p with m which is the number of iterations of the fixed point, indeed $n+1 = 2p+2 = 2(m+1) \Longleftrightarrow m = p$.

h	$p = m = 0$	$p = m = 1$	$p = m = 2$	$p = m = 3$
1/2	0.08574	0.002774567	0.0000780726648	0.000002586969712
1/4	0.01794	0.000083124	0.0000005857086	0.000000002761407
1/8	0.00417	0.000003792	0.0000000065231	0.000000000007073
1/16	0.00100	0.000000204	0.0000000000866	0.000000000000023
1/32	0.00024	0.000000011	0.0000000000012	$\varepsilon_{mach.}$
order	≈ 2	≈ 4	≈ 6	≈ 8

TABLE 5.4

Relative L^∞ errors between the function $f_h(\cdot) = f(h\cdot)$ with f provided in (5.7) and its approximated interpolant p_n^m with $n = 2p + 1$, as a function of $p = m$. The observed convergence order is in accordance with Theorem 1.2, namely $2(m + 1)$.

5.2.3. Optimality with respect to the polynomial degreee. The approximation by positive interpolation polynomials is optimal in terms of polynomial degree, and so is optimal in terms of accuracy. This can be visualized by comparison with another trivial positive approximation which writes

$$\widehat{p}_n = \left(\mathcal{I}_p\left(\sqrt{f}\right)\right)^2, \quad n = 2p, \tag{5.8}$$

where $\mathcal{I}_p$ is the standard Lagrange interpolation operator with degree $p = n/2$. For $n = 10$, we compare $\widehat{p}_n$ with p_n provided by Algorithm (3.8). The result displayed in Figure 5.2 shows without surprise that p_n which uses 11 interpolation points is much more accurate than $\widehat{p}_n$ which uses only 6 interpolation points. The target is the Runge function R, see (5.6).

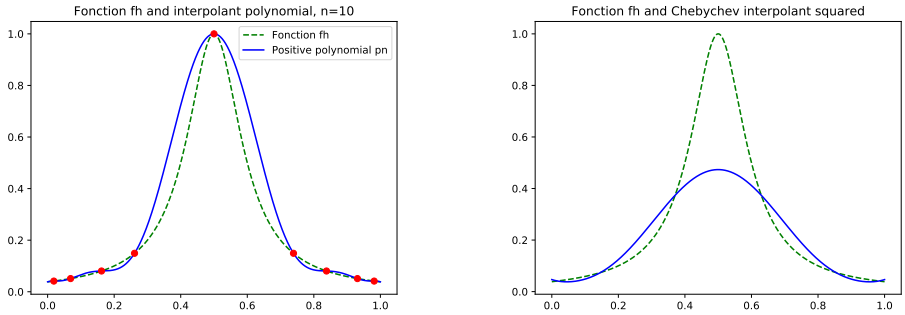

FIG. 5.2. *Approximation of R by two polynomial of degree $n = 10$. Left: p_n in dots is the positive polynomial obtained by iteration. Right: $\widehat{p}_n$ is obtained by squaring procedure (5.8) of standard Chebyshev interpolation, so only 6 interpolation points are involved at the intersection of the curves.*

5.3. A simple certificate of positivity: Algorithms 1.2 and 1.3. We show how to interpret the positive polynomials as a certificate of positivity, as in Algorithms 1.2 and 1.3. This method is the basis of the algorithms in the next section.

Instead of developing a general theory, we propose a simple example. We consider the polynomial function $q_\lambda \in P_4$

$$q_\lambda(x) = 10(x - 1/2)^4 + \lambda \tag{5.9}$$

and consider different values of λ. For $\lambda \geq 0$, one clearly has $q_\lambda \in P_4^+$, and on the other hand, for $\lambda < 0$ then $q_\lambda \notin P_4^+$. Of course, it is evident for such a polynomial to know wether q_λ is negative of not. The point is that for a general polynomial of arbitrary order, it can be quite difficult.

In order to propose a general solution, we construct the sequence of positive polynomials $p_4^m \in P_4^+$ with the simplified Newton-Raphson Algorithm (5.1) with a small offset $\varepsilon > 0$. The iterations are performed up to a given arbitrary degree which is taken a priori sufficiently large. The two main cases are

- either $q_\lambda \in P_4^+$, then $p_4^m \in P_4^+$ is very close to q_λ
- or $q_\lambda \notin P_4^+$, then $p_4^m \in P_4^+$ can be used as a non negative polynomial surrogate to the objective function q_λ.

This is illustrated in Figure 5.3 where we approximate $q_{\lambda=0.1}$ and $q_{\lambda=-0.1}$ with positive polynomials of degree $n = 4$ and $n = 9$. For $\lambda = 0.1 > 0$, one observes without surprise that the two top results in Figure 5.3 are extremely accurate. On the other hand for $\lambda = -0.1 < 0$, the two bottom figures in Figure 5.3 show that positive polynomials have the ability to capture a very good (polynomial) non negative approximation of $\max(q_\lambda, \varepsilon)$. The iteration of positive polynomials constructs in this case a practical (in the sense of Algorithm 1.3) certificate of positivity. One is nevertheless forced to increase the polynomial degree to obtain good accuracy (in this case, a doubling).

5.4. Numerical approximation of the advection equation. This section can be considered as an ultimate justification of the introduction of the parameter h in the various theorems of approximation, such as the main one Theorem 1.2. This parameter is now proportional to the mesh size $\Delta x > 0$ used to discretize partial differential equations. We consider the numerical discretization of the advection

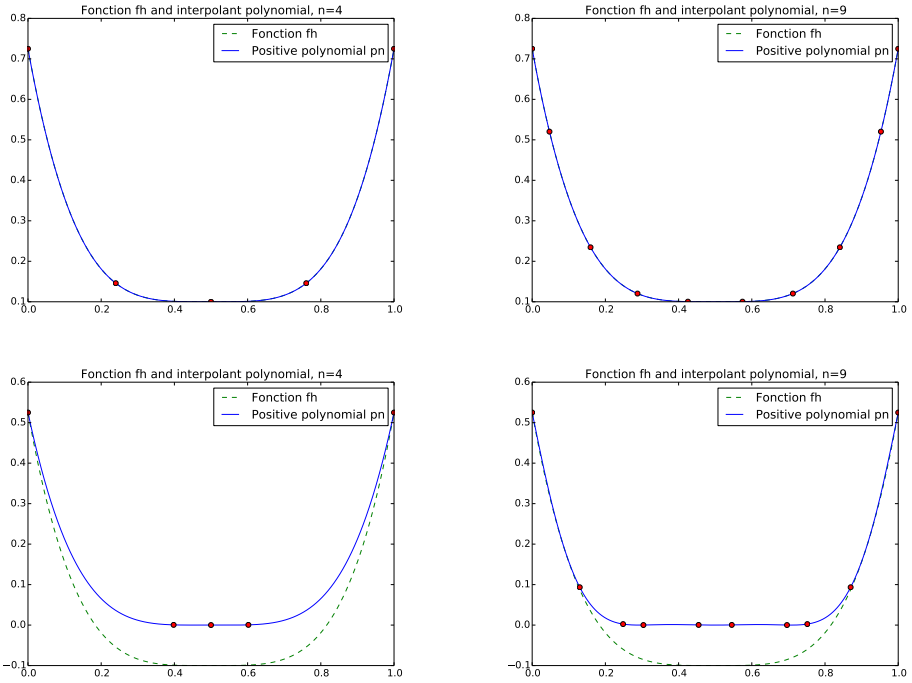

FIG. 5.3. *The objective polynomial can be either in P_n^+ (top) or not (bottom). In case it admits negative values, positive polynomials construct good approximation.*

equation

$$\begin{cases} \partial_t u + a\partial_x u = 0, & x \in \mathbb{R}, \quad t > 0, \quad a = 1, \\ u(x,0) = u_0(x), & x \in \mathbb{R}, \end{cases}$$

with a periodic initial data $u_0(x+1) = u_0(x)$ for all $x \in \mathbb{R}$. Assuming $u_0 \geq 0$ or $u_0 > 0$, one desires to design methods which respect these conditions. Two methods are tested, which are based on the approximate certificate of positivity described in section 5.3. That is the algorithms start by reconstructing a classical Lagrange interpolation from the available local data: this yields a local polynomial $p_n \in P_n$ with $n = 2p+1$; in a second stage we use the iteration loop (3.8) with $m = p$ steps; it yields a local polynomial $\widetilde{p}_n \in P_n^+$ which is a high order approximation of p_n. The first method is based on the semi-Lagrangian method [3] where the value at the foot of the characteristic is predicted by standard Lagrangian interpolation then modified so as to get a non negative value. The second one if essentially similar up to the fact that we solve the transport equation by computing fluxes equal to the integral over the length $\Delta l = a\Delta t$ of a polynomial $\widetilde{p}_n \in P_n^+$. The stability of the resulting scheme/algorithm is not possible to determine by theoretical means yet. We can only say that the stencil of the Lagrange interpolation is linearly stable in any L^r, $1 \leq r \leq \infty$, see [3], and that the guaranteed non negativity of the global method yields some non linear stability. Take the initial data is $u_0(x) = \cos(\pi x)^2 + 1 > 0$. The CFL constant is $a\frac{\Delta t}{\Delta x} = 0.5$. Therefore the foot of the characteristics is at the middle of the first left cell. In Table 5.5, we display the L^∞ error at time $T_{end} = 1$ for the semi-Lagrangian implementation, as a function of the numbers of cells.

h	$n=1$	$n=3$	$n=5$	$n=7$
20	0.195	0.0045767	0.00020966477	0.000016399842
40	0.109	0.0005707	0.00001083749	0.000000551818
80	0.058	0.0000725	0.00000039502	0.000000006955
160	0.029	0.0000091	0.00000001303.	0.000000000065
320	0.015	0.0000011	0.00000000041	$\varepsilon_{mach.}$
order	≈ 1	≈ 3	≈ 5	≈ 7

TABLE 5.5

h-convergence with respect to the polynomial degree n for the positive semi-Lagrangian scheme.

Same test problem with a conservative implementation (ENO-like reconstruction of the fluxes) yields the results in Table 5.6 with increase of the convergence order.

h	$n=1$	$n=3$	$n=5$	$n=7$
20	0.03791	0.000800990	0.000022346530	0.000001284228
40	0.00964	0.000052541	0.000000460477	0.000000023856
80	0.00242	0.000003355	0.000000008021	0.000000000168
160	0.00060	0.000000211	0.000000000132	$\varepsilon_{mach.}$
320	0.00015	0.000000013	$\varepsilon_{mach.}$	$\varepsilon_{mach.}$
order	≈ 2	≈ 4	≈ 6	≈ 8

TABLE 5.6

h-convergence wrt the polynomial degree n for the positive conservative semi-Lagrangian scheme. Gain of one convergence order with respect to the semi-Lagrangian scheme (Table 5.5).

REFERENCES

[1] S. Butt and K. W. Brodlie, Preserving positivity using piecewise cubic interpolation, Comput. & Graphics Vol. 17, No, 1, pp. 55-64, 1993.

[2] B. Després, Polynomials with bounds and numerical approximation, Numer. Algo., 1-31, 2017.

[3] B. Després, Uniform asymptotic stability of Strang's explicit compact schemes for linear advection, SIAM J. Numer. Anal. 47 (2009), no. 5, 3956-3976.

[4] H. Hong and D. Jakus, Testing Positiveness of Polynomials, J. of Aut. Reas. 21: 23-38, 1998.

[5] J.-B. Lasserre, Moments, Positive Polynomials and Their Applications, Imp. Col. Press, 2010.

[6] G.V. Milovanovic, D.S. Mitrinovic and T.M. Rassias, Topics in polynomials: extremal problems, inequalities, zeros. World Scientific Publishing Co., Inc., River Edge, NJ, 1994.

[7] V. Powers, Positive Polynomials and Sums of Squares: Theory and Practice, in Real Algebraic Geometry 2011, Conference Rennes University, Editors Basu and al.

[8] J.J. Risler, Mathematical methods for CAD. Camb. Univ. Press, Cambridge, 1992.

[9] J.J. Risler, Computer aided geometric design. Handbook of numerical analysis, Vol. V, 715-818, Handb. Numer. Anal., V, North-Holland, Amsterdam, 1997.

[10] J.-W. Schmidt and W. Hess, Positivity of cubic polynomials on intervals and positive spline interpolation, BIT Numerical Mathematics June 1988, Volume 28, Issue 2, 340-352.

[11] C.W. Shu, Bound-preserving high order accurate schemes , Notes of the Canadian Mathematical Society (CMS Notes), v45 (2013), pp.24-25.

[12] Olver and al editors, NIST Handbook of Mathematical Functions, Camb. Univ. Press, 2010.

[13] G. Szegö, Orthogonal polynomials, American Mathematical Society, Providence, R.I., 1939.

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

[15] J.-H. Zhang and Z.-X.Yao, Optimized explicit finite-difference schemes for spatial derivatives using maximum norm, Journal of Computational Physics 250 (2013) 511-526.