SIAM J. NUMER. ANAL.
Vol. 57, No. 1, pp. 148–172




# ALGORITHMS FOR POSITIVE POLYNOMIAL APPROXIMATION*

MARTIN CAMPOS-PINTO†, FRÉDÉRIQUE CHARLES†, AND BRUNO DESPRÉS†

**Abstract.** We propose several algorithms for positive polynomial approximation. The main tool is a novel iterative method to compute nonnegative 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 nonnegative 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

**DOI.** 10.1137/17M1131891

## 1. Introduction.

Let $P_n$ denote the set of real polynomials of degree $\leq n$ over the interval $[0,1]$. The basic problem considered in this work concerns the interpolation of a Lipschitz function, $f \in W^{1,\infty}(0,1)$, by nonnegative polynomials in $P_n$, the set of which we denote by $P_n^+ = \{p_n \in P_n : p_n(x) \geq 0 \ \forall x \in [0,1]\}$. In most cases we will assume that the function is positive over the interval,

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

but nonnegative functions $f \geq 0$ will also be considered. Specifically, we consider the following formulation.

PROBLEM 1.1. *Given $f \in W^{1,\infty}(0,1)$ satisfying (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 $0 \leq i \leq n$ satisfies $p_n \in P_n^+$.*

Such a problem is of interest for pure numerical analysis purposes, but we also point out that being able to find a convenient characterization of positive polynomials is a central issue in several domains of scientific computing. A nonexhaustive list of references that reflect some of our own interests is [1, 12] for positive interpolation with cubic polynomials, [6] for automated testing, [9, 7] for characterization with sums of squares, [10, 11] on computer aided design with Bernstein and Bézier curves, [17, 13, 16] for nonnegative numerical approximation of hyperbolic equations, and [7, 8] and the references therein for comprehensive results 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 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

*Received by the editors May 25, 2017; accepted for publication (in revised form) November 16, 2018; published electronically January 15, 2019.
http://www.siam.org/journals/sinum/57-1/M113189.html

†1-Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France, and 2-CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France (campos@ljll.math.upmc.fr, charles@ljll.math.upmc.fr, despres@ljll.math.upmc.fr).

identified with the mesh size, as will be evidenced at the end of this article. So we believe that such a parameter $h$ is very natural in view of PDE discretizations.

The solutions to Problem 1.1 that we will propose can be cast under the following general form.

ALGORITHM 1.1. *Given* $f \in W^{1,\infty}(0,1)$ *satisfying* (1.1), *compute a sequence* $p_n^m \in P_n^+$ *for* $m = 0, 1, \ldots,$ *such that* $p_n^m \longrightarrow p_n \in P_n^+$ *and* $p_n$ *solves Problem* 1.1.

A byproduct of our analysis is a practical algorithm to determine whether a given polynomial is positive over $[0,1]$. Indeed in some cases positive polynomials are mandatory (as an ingredient in a given calculus), and such an algorithm provides a practical solution which can be interpreted as a new approximate certificate of positivity; see [9, 7].

ALGORITHM 1.2. *Given* $f \in P_n$, *consider the sequence* $(p_n^m)_{m\in\mathbb{N}}$ *computed by Algorithm* 1.1. *If* $p_n^m$ *tends to* $f$, *then* $f$ *is nonnegative over* $[0,1]$.

In practice it is also possible to use the following version of the above algorithms.

ALGORITHM 1.3. *Given* $f \in W^{1,\infty}(0,1)$ *satisfying* (1.1), *compute* $p_n^m \in P_n^+$ *the* $m$*th iterate of the sequence in Algorithm* 1.1 *for* $m$ *large enough. Replace* $f$ *by* $p_n^m$.

The interest in 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. And even if the theoretical convergence conditions are not realized, the polynomial $p_n^m$ can still be considered as a positive approximation of $f$.

Our objective here is to propose specific realizations of these abstract algorithms and to assess their convergence properties, both at the theoretical level (considering that Problem 1.1 may be applied to a localized function $f_h$ as described above) and at the numerical level.

**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. In our context, it is provided by the Lukács theorem; see, e.g., [15, sect. 1.21]. A recent proof in real algebra is available in [3].

THEOREM 1.1 (Lukács).

- *If* $n = 2p$, *then* $p_n \in P_n^+$ *if and only if there exists* $a_p \in P_p$ *and* $b_{p-1} \in P_{p-1}$ *such that* $p_n(x) = a_p(x)^2 + x(1-x)\, b_{p-1}(x)^2$.
- *If* $n = 2p+1$, *then* $p_n \in P_n^+$ *if and only if there exists* $a_p, b_p \in P_p$ *such that* $p_n(x) = x\, a_p(x)^2 + (1-x)\, b_p(x)^2$.

Our solutions to Problem 1.1 will be based on the possibility of determining the interpolation points in combination with a direct construction of the polynomials $a_p$ and $b_p$ (or $b_{p-1}$) appearing in the Lukács theorem. This construction involves two main technical ideas, namely, *oscillating polynomials* and *sliding interpolation points*. To illustrate these ideas we first consider the simplest case of $P_2^+$.

**Interpolation by positive quadratic polynomials.** According to the Lukács theorem we know that interpolating a given positive function $f$ by a polynomial

$p_2 \in P_2^+$ amounts to determining $a_1 \in P_1$ and $b_0 \in P_0 = \mathbb{R}$ such that

$$p_2(x) := a_1(x)^2 + x(1-x)b_0^2 \tag{1.2}$$

interpolates $f$ at three points in $[0,1]$. Observing that the weight $x(1-x)$ vanishes at the endpoints of the interval, we consider the case where 0 and 1 are two interpolation points. Then one has $a_1(0)^2 = p_2(0) = f(0) > 0$ and $a_1(1)^2 = p_2(1) = f(1) > 0$. There are two possibilities to finalize the construction of $a_1$.

• The first one uses an elementary idea from [3] that consists in imposing a change of sign in $a_1$ (say, $a_1(0) > 0$ and $a_1(1) < 0$), which is called an oscillating polynomial in the core of this paper. This choice gives $a_1(x) = \sqrt{f(0)}(1-x) - \sqrt{f(1)}x$. Since the polynomial $p_2 - a_1^2$ vanishes at the endpoints one can write $p_2 = a_1^2 + x(1-x)e$ with $e \in \mathbb{R}$. Now, a consequence of the fact that the sign of $a_1$ changes at the endpoints is that there exists $x_* \in (0,1)$ such that $a_1(x_*) = 0$. We can take this point as the third interpolation point: one has indeed $e = \frac{p_2(x_*)}{x_*(1-x_*)} = \frac{f(x_*)}{x_*(1-x_*)} > 0$ so that we can define $b_0 = \sqrt{e}$. This completes the construction, and also proves the Lukács theorem in the case $n = 2$. Notice that the inner interpolation point $x_*$ is determined self-consistently when solving the interpolation problem. As it is not known a priori but defined within the construction, we call it a sliding interpolation point.

• The other solution would be to design $a_1$ with the same sign (say, positive) at the endpoints. Then a simple counterexample shows that the construction might not work: take $f(x) = (1-2x)^2 = p_2(x)$, then $a_1(x) = \sqrt{f(0)}(1-x) + \sqrt{f(1)}x \equiv 1$ which leads to $b_0^2 = \frac{p_2(x) - a_1(x)^2}{x(1-x)} = -4$. In particular $b_0$ is a pure imaginary number which cannot yield a solution to the Lukács theorem.

In summary, we see that a generic solution to the positive interpolation problem in the case $n = 2$ is easily designed using the Lukács representation form. In this construction the use of oscillating polynomials appears as a convenient tool, and a necessary option as well.

**Main results.** Our contributions are twofold. On a practical level, we propose several algorithms that extend the above construction to arbitrary high degrees $n$. To do so we generalize the use of oscillating polynomials and we exploit their vanishing properties to define convenient sets of sliding interpolation points that allow us to design a solution to our positive interpolation problem in Lukács representation form. Iterative algorithms are then proposed to compute these interpolation points, under the form of simplified Newton–Raphson schemes. Our numerical tests demonstrate their efficiency and robustness with respect to several parameters. On a theoretical level, we establish the convergence of these fixed-point algorithms together with an a priori convergence estimate, under the assumption that the localization parameter $h$ introduced above is small enough. We also provide a convergence estimate for the resulting positive polynomial approximation, which is optimal in $h$ for a number of iterations that only depends on $n$. These results may be summarized as follows, where the uniform norm over $(0,1)$ is denoted by $\|\cdot\|$.

THEOREM 1.2. *Let $n \in \mathbb{N}$ and let $f \in W^{q,\infty}(0,1)$, $1 \le q \le n+1$, be a function that is positive over $[0,1]$ (hypothesis (1.1)). Denote $f_h(\cdot) = f(\cdot h)$ for $0 \le h \le 1$.*

1. *A simplified Newton–Raphson scheme allows us to compute a sequence of positive polynomials $p_n^m \in P_n^+$ for $m = 0, 1, \dots,$ to approximate $f_h$.*
2. *If $n = 2p+1$, the construction involves oscillating polynomials $\left(a_p^m, b_p^m\right) \in P_p^2$ and the odd order Lukàcs representation $p_n^m(x) = x a_p^m(x)^2 + (1-x) b_p^m(x)^2$.*

3. *There exists $h_0 > 0$ such that for all $0 \le h \le h_0$, the polynomials admit a limit $(a_p^\infty, b_{p-1}^\infty)$ and the approximation estimate*

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

*holds with a constant $C$ that may depend on $n$ and $f$, but not on $h$ nor on $m$.*
4. *If $n = 2p$, the same results hold with an even order Lukàcs representation $p_n^m(x) = a_p^m(x)^2 + x(1-x)b_{p-1}^m(x)^2$ involving oscillating polynomials $\left(a_p^m, b_{p-1}^m\right) \in P_p \times P_{p-1}$.*

Note that for smooth functions the $h$-$m$ convergence estimate (1.3) yields an optimal rate, namely, $h^{n+1}$, for a fixed number of iterations $m = p$. This is consistent with the a priori estimate [3] for the best approximation error by positive polynomials.

**Organization.** Since the case $n = 2$ is almost trivial as seen in the discussion above, we address the first interesting case, namely, $n = 3$, in section 2. For this nontrivial case we propose a first fixed-point formulation of the positive interpolation problem and establish its well-posedness, along with rigorous convergence estimates for the associated iterative scheme. This approach is then extended to arbitrary high odd orders $n = 2p + 1$ in section 3, where we propose a more general method based on a simplified Newton–Raphson algorithm. Again this iterative scheme is proven to converge for small values of the localization parameter $h$, and a detailed proof is given for the $h$-$m$ convergence estimate (1.3). (For even degrees $n = 2p$ the same method applies but because of the different Lukács representation form some formulas take different expressions. They are given in the appendix.)

Section 4 then provides some details on the implementation and presents several numerical experiments which all show that the algorithms proposed in this work display good convergence and approximation properties. Interestingly, these results hold not only for small values of $h$ but also for larger ones, typically $h = 1$. Hence the range of valid parameters seems much larger than predicted by Theorem 1.2. In view of the possible use of such methods in a variety of practical problems, we believe that this is extremely important information. Finally, an elementary 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.

In this section we describe the treatment of the case $n = 3$ which is slightly simpler than the general case addressed in the next section. It is the first nontrivial extension of the quadratic case considered above, and it actually involves most of the important ideas that will be used later. As the proofs present no particular difficulties we only state the results. For the technical details we refer the reader to a preliminary version [2] of this article.

Using again the Lukács theorem we look for a solution to Problem 1.1 in the form

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

Here $a_1$ and $b_1$ are affine polynomials, therefore, they can be constructed by linear interpolation knowing two values. For $a_1$, one value is natural: indeed at $x = 1$ the identity reads $p_3(1) = a_1(1)^2$. So the main question is to obtain another interpolation point. Being optimistic, let us assume there exists $0 < \alpha < 1$ such that $b_1(\alpha) = 0$. Then one obtains $p_3(\alpha) = \alpha a_1(\alpha)^2$ and two possibilities for the reconstruction of $a_1$, depending on the sign of $a_1$ at $\alpha$ and 1. As seen in the introduction the best option is to take alternate signs in order to construct an oscillating polynomial: here,

$$a_1(x) = \sqrt{p_3(1)}\frac{(x-\alpha)}{1-\alpha} - \sqrt{\frac{p_3(\alpha)}{\alpha}}\frac{(1-x)}{1-\alpha}.$$

Similarly we may write $b_1(x) = \sqrt{\frac{p_3(\beta)}{1-\beta}}\frac{x}{\beta} - \sqrt{p_3(0)}\frac{(\beta-x)}{\beta}$ if there exists $\beta \in (0,1)$ such that $a_1(\beta) = 0$. Thus we have a series of relations and a couple of additional degrees of freedom. Elaborating on the idea that $p_3$ should interpolate $f$ at 0, 1 and at the sliding interpolation points $\alpha$ and $\beta$, we state a first result.

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

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

*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, $\alpha$, $\beta$, and 1.*

By looking at (2.1), the equation $a_1(\beta) = 0$ allows us to express $\beta$ as a function of $\alpha$ while the equation $b_1(\alpha) = 0$ gives $\alpha$ as a function of $\beta$. The relations (2.1) are then equivalent to the fixed-point problem $(\alpha, \beta) = K(\alpha, \beta)$ with $K(\alpha, \beta) = (\varphi(\beta), \psi(\alpha))$, where $\varphi$ and $\psi$ are two functions from $[0,1] \to [0,1]$ defined as

$$\varphi(\beta) = \frac{\beta\sqrt{(1-\beta)f(0)}}{\sqrt{(1-\beta)f(0)} + \sqrt{f(\beta)}} \quad \text{and} \quad \psi(\alpha) = \frac{\alpha\sqrt{\alpha f(1)} + \sqrt{f(\alpha)}}{\sqrt{\alpha f(1)} + \sqrt{f(\alpha)}}. \tag{2.2}$$

This fixed-point equation may also be recast as $T(\alpha) = 0$ with $T(\alpha) := \alpha - \varphi(\psi(\alpha))$.

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

Corollary 2.3. *There exist 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, $\alpha$, $\beta$, and 1.*

*Proof.* The proof is evident since there exists by continuity $0 < \alpha < 1$ such that $T(\alpha) = 0$. ∎

The next step is to construct a fixed-point algorithm with good properties. As explained above the convergence is better studied after a localization, so we consider $f_h(\cdot) = f(h\cdot)$ instead of $f$. The function $K$ then becomes $K_h(\alpha, \beta) = (\varphi_h(\beta), \psi_h(\alpha))$, where $\varphi_h$ and $\psi_h$ are defined as in (2.2), with $f$ replaced by $f_h$.

Algorithm 2.1. *Given $X^0 = (\alpha^0, \beta^0) \in (0,1)^2$, apply a fixed-point scheme to compute $X^{m+1} := K_h(X^m)$ for $m = 0, 1, \ldots$.*

Since $f$ is positive we verify that $K_h([0,1]^2) \subset [0,1]^2$, hence, the scheme is well-defined. The goal hereafter is to determine reasonable conditions for which the fixed-point converges. One can write $K_h(\alpha, \beta) = \big(\mathcal{K}\big(\beta, \sigma_h(\beta)\big), 1 - \mathcal{K}\big(1-\alpha, \tau_h(\alpha)\big)\big)$ 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 $\tau_h$ and $\sigma_h$ converge towards 1 uniformly on $[0,1]$. Hence $K_h$ converges uniformly on $[0,1]^2$ towards $K_0(\alpha, \beta) = (\mathcal{K}(\beta), 1 - \mathcal{K}(1-\alpha))$, where we have written for simplicity $\mathcal{K}(z) = \mathcal{K}(z, 1)$. The good news is that the function $K_h$ has good properties for $h$ small enough because its limit $K_0$ has good properties. Specifically (i) $K_0$ 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 a contraction on $F$ in the maximum norm; (iii) it has a unique fixed point in $(0,1)^2$, which is $\underline{X} = (\underline{\alpha}, \underline{\beta}) = \left(\frac{1}{4}, \frac{3}{4}\right) \in F$; (iv) the Jacobian matrix $\nabla K_0 = (\partial_j (K_0)_i)_{1 \le i,j \le 2}$ vanishes at $\underline{X}$. By continuity, we obtain the following properties for $K_h$.

LEMMA 2.4. *Let $f \in W^{1,\infty}(0,1)$ satisfy* (1.1). *Then there exists $h^* > 0$ and a constant $C^*$ such that for all $0 \le h \le h^*$,* (i) *$K_h$ leaves the domain $F$ invariant;* (ii) *$K_h$ is a contraction on $F$ in the maximum norm;* (iii) *the Jacobian matrix $\nabla K_h$ satisfies $\|\nabla K_h(X)\| \le C^*(h + \|X - \underline{X}\|)$ for $X \in F$;* (iv) *the derivative of $K_h$ with respect to $h$ satisfies $\|\partial_h K_h(X)\| \le C^*$ for $X \in F$.*

Since $h$ is expected to become small in the theory, a natural starting point is the optimal one for $h = 0$, that is, $X^0 := \underline{X} = \left(\frac{1}{4}, \frac{3}{4}\right)$. Obviously, it is independent of $h$.

PROPOSITION 2.5. *Let $f \in W^{1,\infty}(0,1)$ satisfy* (1.1). *There exists $h_0 > 0$ such that for all $0 \le h \le h_0$, the sequence $(X^m)_{m \ge 0}$ remains in the domain $F \subset (0,1)^2$ and converges to a fixed point of $K_h$ denoted $X_h^\infty \in F$. For all $m \ge 0$, one has the convergence estimate $\|X^\infty - X^m\| \le 2\left(\frac{h}{2h_0}\right)^{m+1}$.*

An important point is that it is not necessary to reach the convergence to get a positive polynomial *approximation* to $f_h$. Indeed from each iterate $(\alpha^m, \beta^m) = X^m$ we can define two affine polynomials, namely, $a_1^m(x) := \left(\frac{x-\alpha^m}{1-\alpha^m}\right)\sqrt{f(h)} - \left(\frac{x-1}{\alpha^m - 1}\right)\sqrt{\frac{f(h\alpha^m)}{\alpha^m}}$ and $b_1^m(x) := \left(\frac{x-\beta^m}{-\beta^m}\right)\sqrt{f(0)} - \left(\frac{x}{\beta^m}\right)\sqrt{\frac{f(h\beta^m)}{1-\beta^m}}$, and a cubic approximation to $f_h$ as

$$p_3^m(x) := x a_1^m(x)^2 + (1-x) b_1^m(x)^2 \quad \in \; P_3^+. \tag{2.3}$$

PROPOSITION 2.6. *Let $f \in W^{q,\infty}(0,1)$, $1 \le q \le 4$, satisfy* (1.1), *and let $h_0 > 0$ be given by Proposition* 2.5. *There exists a constant $C$ for which the cubic polynomial* (2.3) *satisfies $\|p_3^m - f_h\| \le C h^{\min(q, 2(m+1))}$ for all $0 \le h \le h_0$ and $m \ge 0$.*

This error estimate shows that only one iteration is actually needed for the optimal convergence in $h$. Indeed the rate is optimal for $q = 4$ and $2(m+1) = q$, that is, $m = 1$.

## 3. Interpolation in $P_n^+$ with $n = 2p + 1$.

The objective of this section is to extend to arbitrary odd orders the approach described in the previous section for $n = 3$. (The extension to even orders poses no additional difficulties but the formulas need to be adapted to the specific Lukács representation form; see the appendix). One is essentially faced with two difficulties.

First, the exact calculation of the roots of a polynomial of arbitrary order is of course not possible. Therefore one cannot generalize in a straightforward manner the previous method which was based on the exact calculation of the root of affine polynomials. This difficulty is easily avoided by formulating the fixed-point problem as a set of polynomial equations that the roots must satisfy; see (3.7) below. The new algorithm (3.8) in section 3.1.1 is based on a Newton–Raphson algorithm which can be seen as a new iterative procedure to determine the sliding interpolation points. The second difficulty is perhaps more fundamental. Indeed one needs to start the Newton–Raphson algorithm from a well-chosen starting point, and one needs to prove that the Jacobian involved in the Newton–Raphson algorithm is a nonsingular matrix. It appears that it is possible to obtain an efficient solution that solves this second difficulty by using a suitable combination of Chebyshev polynomials which are natural oscillating polynomials. This is detailed in section 3.1.2.

The structure of the section follows the one of Theorem 1.2.

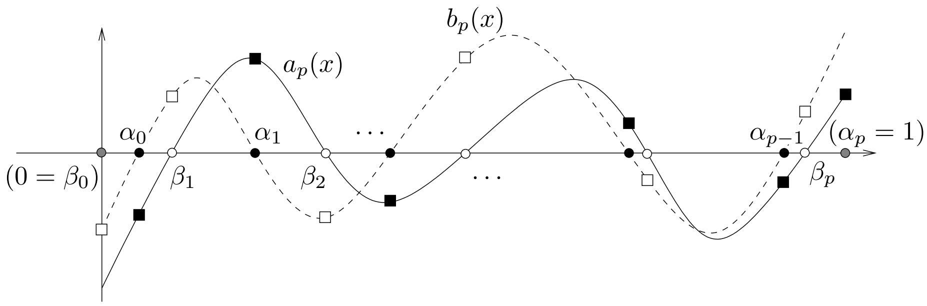


FIG. 3.1. *Sketch of two oscillating polynomials $a_{p,h}$ and $b_{p,h}$ in $P_p$ satisfying the criteria of Proposition* 3.1. *The black (resp., white) circles indicate the positions of the nodes $\alpha = (\alpha_0, \ldots, \alpha_{p-1})$ (resp., $\beta = (\beta_1, \ldots, \beta_p)$), where $b_{p,h}$ (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.*

**3.1. Construction of the iterative scheme: Items 1–2 of Theorem 1.2.**

**3.1.1. Formulating the positive interpolation as a root-finding problem.** Let $f \in W^{1,\infty}(0,1)$ be positive over $[0,1]$ (see (1.1)), and let $h \geq 0$. In order to recast the original interpolation problem as a root-finding one, we state sufficient criteria that extends the notion of oscillating polynomials developed in the previous section to arbitrary odd orders. An illustration is given in Figure 3.1.

PROPOSITION 3.1. *Assume there exist two polynomials $a_{p,h}, b_{p,h}$ in $P_p$, and $2p$ nodes $\alpha_0 < \cdots < \alpha_{p-1}$ and $\beta_1 < \cdots < \beta_p$ in $(0,1)$ with the following properties:*

(a) *the nodes are roots of the polynomials in the sense that*

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

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

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

*with fixed additional nodes $\beta_0 := 0$, $\alpha_p := 1$. Then the $n+1$ nodes are interlaced, $0 = \beta_0 < \alpha_0 < \beta_1 < \cdots < \beta_p < \alpha_p = 1$, and $p_n(x) := xa_{p,h}(x)^2 + (1-x)b_{p,h}(x)^2 \in P_n^+$ is the interpolation polynomial of $f_h = f(h\,\cdot)$ on these $n+1$ interlaced nodes.*

*Remark* 3.2. Clearly $p_n$ depends on $h$ by construction, but to simplify the notation we discard the index $h$. The reason why we keep this index in the polynomials $a_{p,h}$ and $b_{p,h}$ is to make a clear distinction with $\underline{a}_p$ and $\underline{b}_p$ which will be defined later.

*Proof.* From (3.2) the sign of $a_{p,h}(\alpha_i)$ is alternating, so its roots $\beta_i$ alternate with the $\alpha_i$. The same holds starting from the sign of $b_{p,h}(\beta_i)$. By construction we have $p_n(\alpha_i) = f(h\alpha_i)$ and $p_n(\beta_i) = f(h\beta_i)$ for $0 \leq i \leq p$. This yields $2p+2 = n+1$ interpolation points, so $p_n \in P_n$ is the interpolation polynomial at these points. ∎

The above criterion is made effective as follows. We denote ordered sets of $p$ distinct inner nodes by

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

For $(\alpha,\beta)=(\alpha_0,\dots,\alpha_{p-1};\,\beta_1,\dots,\beta_p)\in I_p^2$, we then let $a_{p,h}[\alpha]$ and $b_{p,h}[\beta]$ be the polynomials that satisfy the interpolation relations (3.2), namely,

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

with a fixed additional node $\alpha_p=1$, and

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

with a fixed additional node $\beta_0=0$. Next we define $\Theta_{p,h}: I_p^2\longrightarrow\mathbb{R}^{2p}$ by

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

Then the root relation (3.1) in Proposition 3.1 holds if and only if $(\alpha,\beta)\in I_p^2$ satisfies

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

Equation (3.7) is highly nonlinear. To handle it we rely on a Newton–Raphson method, and more precisely a simplified Newton–Raphson scheme which consists in freezing the Jacobian matrix at its starting value; see, e.g., [5]. As we consider a localized version of the problem we simplify the method even further and evaluate the Jacobian matrix at $h=0$.

Algorithm 3.1 (simplified Newton–Raphson scheme). *Given a starting point $X^0\in I_p^2$, compute*

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

*for $m=0,1,\dots$ 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 $X^0$,*

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

*Remark* 3.3. If $\alpha_i=\alpha_{i+1}$ or $\beta_i=\beta_{i+1}$, then $a_{p,h}$ or $b_{p,h}$ become infinite in (3.4)–(3.5). To avoid such degeneracies we introduce below a node separation operator.

Our goal is now to finish the construction of the method. This will be done in three steps. The first step will define a proper starting point $X^0\in I_p^2$, the second step will justify that $J_p(X^0)$ is a nonsingular matrix, and the third step will describe the node separation operator just motivated. The final method is Algorithm 3.2 below.

**3.1.2. Definition of the starting point $X^0$.** We desire the algorithm (3.8) to be exact for the simplest nontrivial case which is $f\equiv 1$ because it is difficult to expect any good property of (3.8) if this trivial case is not 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 $1\equiv x\underline{a}_p(x)^2+(1-x)\underline{b}_p(x)^2$.

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 $(T_p,U_{p-1})\in P_p\times P_{p-1}$,

$$T_p(\cos\theta)=\cos(p\theta)\quad\text{and}\quad U_{p-1}(\cos\theta)=\frac{\sin(p\theta)}{\sin\theta},\quad p\ge 0,$$

which satisfy $T_p(-x)=(-1)^pT_p(x)$ and $U_{p-1}(-x)=(-1)^{p-1}U_{p-1}(x)$; see, e.g., [14].

LEMMA 3.4. *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 $0 \le i \le p$.*

(ii) *Chebyshev form: the above polynomials read*

$$\begin{aligned} \underline{a}_p(x) &= T_p\left(2x-1\right) - 2(1-x)U_{p-1}\left(2x-1\right), \\ \underline{b}_p(x) &= T_p\left(2x-1\right) + 2xU_{p-1}\left(2x-1\right). \end{aligned} \tag{3.12}$$

(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*

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

*Remark* 3.5. Formula (3.15) below shows that $\underline{a}_p$ and $\underline{b}_p$ correspond more precisely to the (shifted) third and fourth kind Chebyshev polynomials [14, 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),

$$\begin{aligned} \hat{a}_p(x) &:= T_p\left(2x-1\right) - 2(1-x)U_{p-1}\left(2x-1\right), \\ \hat{b}_p(x) &:= T_p\left(2x-1\right) + 2xU_{p-1}\left(2x-1\right), \end{aligned}$$

satisfy (iv) and (v), then they coincide with $\underline{a}_p$ and $\underline{b}_p$, which would prove (ii) together with (iv) and (v). 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

$$\begin{aligned} \hat{a}_p(1-x) &= T_p\left(1-2x\right) - 2xU_{p-1}\left(1-2x\right) \\ &= (-1)^p T_p\left(2x-1\right) - 2x(-1)^{p-1}U_{p-1}\left(2x-1\right) = (-1)^p \hat{b}_p(x). \end{aligned}$$

For (iv), we let $\theta$ 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 $\beta_1, \ldots, \beta_p \in (0,1)$. The case of $\hat{b}_p$ follows from properties (i) and (iii). Turning to property (v), we compute $x\hat{a}_p(x)^2 + (1-x)\hat{b}_p(x)^2 = T_p(2x-1)^2 + 4\left(x - x^2\right) U_{p-1}(2x-1)^2$. Again with $\cos\theta = 2x-1$ we find that $\sin^2\theta = 1-(2x-1)^2 = 4(x-x^2)$, and then $T_p(2x-1)^2 + 4\left(x-x^2\right)U_{p-1}(2x-1)^2 = \cos^2(p\theta) + \sin^2\theta\left(\sin(p\theta)/\sin\theta\right)^2 = 1$. This shows (3.14) for the polynomials $\hat{a}_p$, $\hat{b}_p$ and hence completes our argument. ∎

DEFINITION 3.6 (starting point $X^0$). *Using the reference nodes* (3.10), *we define*

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

We next give elementary formulas which are useful for the practical implementation. By definition of the polynomials $\underline{a}_p$ and $\underline{b}_p$ we have (for $0 \le i \le p$)

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

The following result on the derivatives will also be useful in the subsequent analysis,

$$\underline{a}_p'(\alpha_i) = \frac{(-1)^{i+p+1}}{2\underline{\alpha}_i^{3/2}} \quad \text{and} \quad \underline{b}_p'(\beta_{i+1}) = \frac{(-1)^{i+p+1}}{2(1-\underline{\beta}_{i+1})^{3/2}} \quad \text{for } 0 \le i \le p-1. \tag{3.18}$$

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 $\underline{\beta}_i$.

**3.1.3. Study of the reference Jacobian matrix $J_p(X^0)$.** The main result of this section is the fact that the reference Jacobian matrix has a very simple structure and is nonsingular. We also provide explicit formulas.

LEMMA 3.7. *The reference Jacobian matrix defined by* (3.9) *has the form*

$$J_p(X^0) = \begin{pmatrix} \nabla_\alpha b_{p,0}[\beta](\alpha) & \nabla_\beta b_{p,0}[\beta](\alpha) \\ \nabla_\alpha a_{p,0}[\alpha](\beta) & \nabla_\beta a_{p,0}[\alpha](\beta) \end{pmatrix}\Bigg|_{(\alpha,\beta)=X^0} = \sqrt{f(0)}\begin{pmatrix} D_\alpha & 0 \\ 0 & D_\beta \end{pmatrix}, \tag{3.19}$$

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

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

*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 $a_{p,0} = \sqrt{f(0)}\underline{a}_p$ and $b_{p,0} = \sqrt{f(0)}\underline{b}_p$. The other properties are proved as follows.

• First, the diagonal form of the diagonal blocks of $J_p(X^0)$ is clear, since for $j \neq i$, $a_{p,0}[\alpha](\beta_i)$ does not depend on $\beta_j$ (and similarly $b_{p,0}[\beta](\alpha_i)$ does not depend on $\alpha_j$). The values of the nonzero entries then follow by direct computation, using the formulas $\partial_x\{T_p(2x-1)\} = 2pU_{p-1}(2x-1)$ and $\partial_x\{U_{p-1}(2x-1)\} =$

$((2x-1)U_{p-1}(2x-1)-pT_p(2x-1))/(4x(1-x))$. It is also easy to see that these values are nonzero: from Lemma 3.4 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 $\underline{\alpha}_i$.

- Second, we need to study how $a_{p,0}[\alpha]$ defined by (3.4), namely,

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

varies as a function of the inner nodes $\alpha=(\alpha_0,\dots,\alpha_{p-1})$ close to $\underline{\alpha}$ (and similarly for $b_{p,0}[\beta]$). We focus on the dependency with respect to $\alpha_0$ of the above quantity, which means that we freeze all the other variables and study $q(x,\alpha_0):=a_{p,0}[(\alpha_0,\underline{\alpha}_1,\dots,\underline{\alpha}_{p-1})](x)-a_{p,0}[(\underline{\alpha}_0,\underline{\alpha}_1,\dots,\underline{\alpha}_{p-1})](x)$. Observe that since $h=0$ in the reference Jacobian we can assume that $f=1$. One has then the property that $q(\underline{\alpha}_i,\alpha_0)=(-1)^{i+p}\sqrt{\frac{1}{\underline{\alpha}_i}}-(-1)^{i+p}\sqrt{\frac{1}{\underline{\alpha}_i}}=0$ for $i=1,\dots,p$. Since $q(\cdot,\alpha_0)\in P_p$, this yields $q(x,\alpha_0)=\lambda(\alpha_0)\underline{\Pi}_p(x)$ with $\underline{\Pi}_p(x):=\prod_{1\le i\le p}(x-\underline{\alpha}_i)$. Here $\lambda$ can be obtained by identification at an arbitrary point. We take this point equal to $\alpha_0$, which gives

$$\lambda(\alpha_0)=\frac{a_{p,0}[\alpha_0,\underline{\alpha}_1,\dots,\underline{\alpha}_{p-1}](\alpha_0)-a_{p,0}[\underline{\alpha}](\alpha_0)}{\underline{\Pi}_p(\alpha_0)}=\frac{1}{\underline{\Pi}_p(\alpha_0)}\left(\frac{(-1)^p}{\sqrt{\alpha_0}}-\underline{a}_p(\alpha_0)\right). \tag{3.20}$$

By differentiating the representation $a_{p,0}[\alpha_0,\underline{\alpha}_1,\dots,\underline{\alpha}_{p-1}](x)=\underline{a}_p(x)+\lambda(\alpha_0)\underline{\Pi}_p(x)$ with respect to $\alpha_0$, we find $\frac{\partial}{\partial\alpha_0}a_{p,0}[\alpha_0,\underline{\alpha}_1,\dots,\underline{\alpha}_{p-1}](x)=\lambda'(\alpha_0)\underline{\Pi}_p(x)$ and using (3.20) we write

$$\lambda'(\alpha_0)=\frac{\partial}{\partial\alpha_0}\left[\frac{1}{\underline{\Pi}_p(\alpha_0)}\right]\left[\frac{(-1)^p}{\alpha_0^{1/2}}-\underline{a}_p(\alpha_0)\right]+\frac{1}{\underline{\Pi}_p(\alpha_0)}\left[\frac{1}{2}\frac{(-1)^{p+1}}{\alpha_0^{3/2}}-\underline{a}_p'(\alpha_0)\right].$$

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

**3.1.4. Node separation and construction of the final algorithm.** 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 not, the whole construction falls 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 $p\ge 2$ we define a separation set

$$I_{p,\varepsilon}=\{(x_1,\dots x_p)\in[\varepsilon,1-\varepsilon]^p:\varepsilon\le x_i-x_{i-1}\quad\text{for}\quad 2\le i\le p\} \tag{3.21}$$

for some given $\varepsilon>0$. We will require that $\varepsilon\le\frac{1}{2(p+1)}$ which also guarantees that $I_{p,\varepsilon}$ is nonempty. We next introduce a node separation operator to map each iterate $X^m$ in the set $I_{p,\varepsilon}$. This operator is constructed in two steps.

- 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 $0\le\alpha_0^*\le\dots\le\alpha_{p-1}^*\le 1$.

- 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, $\widetilde{\Delta_i} \geq \frac{2\varepsilon}{1+2(p+1)\varepsilon} \geq \varepsilon$ since $\varepsilon \leq \frac{1}{2(p+1)}$. On the other hand, we still have $\sum_{i=0}^p \widetilde{\Delta_i} = 1$ and we can define

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

By construction the resulting vector $\tilde{\alpha} = (\tilde{\alpha}_0, \dots, \tilde{\alpha}_{p-1})$ is indeed in $I_{p,\varepsilon}$.

DEFINITION 3.8. *For $X = (\alpha, \beta) \in \mathbb{R}^{2p}$, we define $S_{p,\varepsilon} X = (\tilde{\alpha}, \tilde{\beta}) \in (I_{p,\varepsilon})^2$ by applying the above construction to $\alpha$ and $\beta$ separately.*

If the nodes are far away from each other we verify that this process has no effect,

$$X \in (I_{p,2\varepsilon})^2 \quad \Longrightarrow \quad S_{p,\varepsilon} X = X. \tag{3.22}$$

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

$$X^{m+1} = S_{p,\varepsilon} G_h(X^m), \tag{3.23}$$

*where $G_h$ defined in* (3.8) *involves the $2p \times 2p$ diagonal Jacobian matrix $J_p(X^0)$; see Lemma* 3.7.

In practice, $\varepsilon$ 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). We will consider that it has a fixed value depending only on $n$, such that the following condition holds:

$$\varepsilon \leq \min\left(\frac{1}{2(p+1)}, \min_{1 \leq i \leq 2p+1}\left(\frac{\underline{\gamma}_i - \underline{\gamma}_{i-1}}{3}\right)\right), \tag{3.24}$$

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

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

In fact, (3.24) guarantees a stronger property that we will also use, which is that the $n-1$ inner nodes $\underline{\gamma} = (\underline{\gamma}_1, \dots, \underline{\gamma}_{2p})$ are in the set $I_{2p,3\varepsilon}$. In particular we have

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

**3.2. Convergence estimates: Item 3 of Theorem 1.2.** Algorithm 3.2 corresponds to items 1 and 2 of Theorem 1.2. A preparatory result for the proof of item 3 is the convergence of the sliding points: it is established in section 3.2.1 below. Then we shall prove item 3, namely, the positive polynomial approximation error estimate, in section 3.2.2.

**3.2.1. Convergence of the sliding interpolation points.** The present analysis is a generalization of the case $p = 1$ corresponding to $n = 3$. 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 regular Newton–Raphson algorithm. It is therefore not a surprise that, for small $h > 0$, our simplified algorithm inherits the very strong contraction properties of regular Newton–Raphson algorithms: the result will be a geometric convergence rate evidenced in (3.30). Again the proof relies on a technical lemma that generalizes Lemma 2.4.

LEMMA 3.9. *Let $f \in W^{1,\infty}(0,1)$ satisfy* (1.1) *and $\varepsilon > 0$ be such that* (3.24) *holds. Then there exists a constant $C^*_{p,\varepsilon}$ independent of $h \in [0,1]$, such that*

(i) *the Jacobian matrix $\nabla G_h$ satisfies*

$$\|\nabla G_h(X)\| \le C^*_{p,\varepsilon}(h + \|X - X^0\|), \qquad X \in (I_{p,\varepsilon})^2; \tag{3.27}$$

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

$$\|\partial_h G_h(X)\| \le C^*_{p,\varepsilon}, \qquad X \in (I_{p,\varepsilon})^2. \tag{3.28}$$

*Proof.* Using (3.23) we compute

$$\nabla G_h(X) = I - \nabla\Theta_{p,0}(X^0)^{-1}\nabla\Theta_{p,h}(X) \tag{3.29}$$

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 $\Phi$ (resp., $\Psi$) involves values of $f$ (resp., $f'$) and is Lipschitz (resp., bounded) on $[0,1] \times (I_{p,\varepsilon})^2$, with Lipschitz constant (resp., $L^\infty$ norm) depending on $f$, $p$, and $\varepsilon$, but not on $h$. From (3.29) 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. ∎

PROPOSITION 3.10 (convergence of the sliding points). *Let $f \in W^{1,\infty}(0,1)$ satisfy* (1.1) *and $\varepsilon = \varepsilon(n) > 0$ be such that* (3.24) *holds. Then there exists $h_0 > 0$ such that for all $0 \le h \le h_0$ the following properties hold.*

(i) *The algorithms* (3.8) *and* (3.23) *with starting point* (3.16) *compute the same iterates $(X^m)_{m\ge 0}$ which belong to the ball $B(X^0,\varepsilon)$, a subset of $(I_{p,2\varepsilon})^2 \subset (0,1)^{2p}$. In particular, the separation operator $S_{p,\varepsilon}$ is not active.*

(ii) *The sequence $(X^m)_{m\ge 0}$ converges towards a fixed point of $G_h$ in the ball $B(X^0,\varepsilon)$, denoted $X^\infty_h = (\alpha^\infty_0, \dots, \alpha^\infty_{p-1}; \beta^\infty_1, \dots \beta^\infty_p)$.*

(iii) *The error estimate holds for all $m \ge 0$,*

$$\|X^\infty_h - X^m\| \le 2\Big(\frac{h}{2h_0}\Big)^{m+1}. \tag{3.30}$$

*Remark* 3.11. In the iterative scheme the nodes $\alpha^m$ and $\beta^m$ must each be in a separation set such as $I_{p,\varepsilon}$, but it is not required that the $n+1$ nodes $\beta_0, \alpha_0, \dots, \alpha_p, \beta_p$ are all bounded away from each other. However, such a property is needed to define an interpolation polynomial on these $n+1$ nodes, and thanks to (3.26) it holds true for every iterate in the ball $B(X^0,\varepsilon)$. In particular the fixed point $X_h^\infty$ consists of interlaced nodes bounded away from each other and from the end nodes, in the sense that $(\alpha_0^\infty,\beta_1^\infty,\dots,\alpha_{p-1}^\infty,\beta_p^\infty)\in I_{2p,\varepsilon}$.

*Proof.* Let $h_0 := (2C^*_{p,\varepsilon})^{-1}\min\big(\varepsilon,(1+2C^*_{p,\varepsilon})^{-1}\big)$ with $C^*_{p,\varepsilon}$ the constant from Lemma 3.9, and let $h\le h_0$. The first bound gives $2C^*_{p,\varepsilon}h\le\varepsilon$ and, using (3.25),

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

For $X\in B(X^0,2C^*_{p,\varepsilon}h)$ we can further write, using (3.27)–(3.28),

$$\begin{aligned}\|G_h(X)-X^0\| &\le \|G_h(X)-G_h(X^0)\|+\|G_h(X^0)-G_0(X^0)\|\\ &\le C^*_{p,\varepsilon}(h+\|X-X^0\|)\|X-X^0\|+C^*_{p,\varepsilon}h\\ &\le 2C^*_{p,\varepsilon}h\big(C^*_{p,\varepsilon}(h+2C^*_{p,\varepsilon}h)+\tfrac12\big)\le 2C^*_{p,\varepsilon}h.\end{aligned}$$

Hence $G_h(B(X^0,2C^*_{p,\varepsilon}h))\subset B(X^0,2C^*_{p,\varepsilon}h)$ and all the iterates $X^m$ are in $B(X^0,\varepsilon)$ and also in $(I_{p,2\varepsilon})^2$. Using (3.22) this shows that the operator $S_{p,\varepsilon}$ has no effect. Since $I_{p,2\varepsilon}\subset I_{p,\varepsilon}$, estimate (3.27) holds and gives $\|\nabla G_h(X)\|\le\frac{h}{2h_0}$. In particular $G_h$ is a contraction on $B(X^0,2C^*_{p,\varepsilon}h)$ and the fixed point theorem of Picard applies: $G_h$ has a unique fixed point $X_h^\infty$ in the ball. Writing $e_h^m=\|X^m-X_h^\infty\|$ we have $e_h^{m+1}=\|G_h(X^m)-G_h(X_h^\infty)\|\le C^*_{p,\varepsilon}(h+2C^*_{p,\varepsilon}h)e_h^m\le\frac{h}{2h_0}e_h^m$ so that $e_h^m\le\big(\frac{h}{2h_0}\big)^m e_h^0$. Estimate (3.30) follows by noticing that $e_h^0=\|X^0-X_h^\infty\|\le 2C^*_{p,\varepsilon}h\le\frac{h}{h_0}$ . ∎

An easy corollary of the above convergence result is a local version of the Lukács theorem.

Corollary 3.12. *Assume $f\in P_n^+$ and $f>0$ on $[0,1]$. Then there exists $h_0>0$ 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 $(\alpha^\infty,\beta^\infty)=X_h^\infty$ 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 $0=\beta_0^\infty<\alpha_0^\infty<\cdots<\beta_p^\infty<\alpha_p^\infty=1$; see Remark 3.11. Since it is an equality between polynomials of degree $\le n$, it yields the claim. ∎

**3.2.2. Convergence of the approximation polynomials.** Let us denote by $(\alpha^m,\beta^m):=X^m$ the $m$th approximation of $(\alpha^\infty,\beta^\infty):=X_h^\infty$ in the iterative Newton scheme (3.23) with starting point (3.16). Since $X^m\in(I_{p,\varepsilon})^2$ thanks to the separation 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

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

where the inner nodes have been again completed by $\beta_0^m=0$ and $\alpha_p^m=1$. Let then

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

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 item 3 of Theorem 1.2 for odd degrees.

PROPOSITION 3.13 (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 Proposition* 3.10*. 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 follows by inspecting the values of $p_n^m$ on 0, $\alpha^m$, $\beta^m$, 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 (3.23) involves a node separation we have $(\alpha^m, \beta^m) \in (I_{p,\varepsilon})^2$ for all $m$, and using also that $f$ is 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 for the doubled rate of convergence. The same bound holds for $|p_n^m(\beta_i^m) - f_h(\beta_i^m)|$.

Let us then 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 $\varepsilon$ according to (3.31) and (3.26); see Remark 3.11. 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 (3.30) this concludes the proof. ∎

## 4. Numerical illustrations.

In this section we first provide some implementation details. Then we present the results of various numerical experiments.

### 4.1. Implementation details.

The practical implementation of the algorithms described above requires elementary modifications, so as to run smoothly even if the hypotheses of Theorem 1.2 are not entirely satisfied. We describe these modifications for the simplified Newton–Raphson algorithm (3.8)

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

In practice, the modified loop writes

$$X^{m+1} = S_{p,\varepsilon}\widehat{G}_h(X^m), \quad \text{where} \quad \widehat{G}_h(X) = X - J_{p,\varepsilon}(X^0)^{-1}\Theta_{p,h,\varepsilon}(X). \tag{4.1}$$

Three ingredients contribute to get a nonsingular algorithm up to a small truncation error of order $\varepsilon > 0$.

• The first one, already introduced in Definition 3.8, is the separation operator $S_{p,\varepsilon}$.

• The second one is a new Jacobian replacing $J_p(X^0)$ (see (3.19)),

$$J_{p,\varepsilon}(X^0) := \sqrt{f_\varepsilon^m} \begin{pmatrix} D_\alpha & 0 \\ 0 & D_\beta \end{pmatrix} \qquad \text{(with matrices } D_{\alpha,\beta} \text{ defined in Lemma 3.7)},$$

where $f_\varepsilon^m$ stands for a maximal value of $f$ evaluated at iteration $m$. A convenient choice is $f_\varepsilon^m := \max\left(\max_{z\in X^m} f(z), \varepsilon\right) \ge \varepsilon > 0$ where we recall that the coordinates of $X^m$ are the nodes $\alpha_i^m$ and $\beta_i^m$.

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

$$a_{p,h,\varepsilon}[\alpha](x) = \sum_{0\le i\le p} (-1)^{i+p} \sqrt{\frac{\max\left(f(h\alpha_i), \varepsilon\right)}{\alpha_i}} \prod_{0\le j\ne i\le p} \frac{x-\alpha_j}{\alpha_i-\alpha_j} \tag{4.2}$$

and

$$b_{p,h,\varepsilon}[\beta](x) = \sum_{0\le i\le p} (-1)^{i+p} \sqrt{\frac{\max\left(f(h\beta_i), \varepsilon\right)}{1-\beta_i}} \prod_{0\le j\ne i\le p} \frac{x-\beta_j}{\beta_i-\beta_j}. \tag{4.3}$$

The new function $\Theta_{p,h,\varepsilon} : I_p^2 \longrightarrow \mathbb{R}^{2p}$ is then

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

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

**4.2. Interpolation.** The application to interpolation problems is presented.

**4.2.1. Cubic nodes.** We illustrate in Table 4.1 the convergence of the fixed-point Algorithm 2.1 which computes the interpolation nodes for the cubic case. We consider the function

$$f(x) = 0.5 + |x-0.5| \text{ for } x < 0.5, \quad f(x) = 0.5 + \frac{1}{2}|x-0.5| \text{ for } 0.5 \le x. \tag{4.4}$$

The numerical values of the nodes $X^m = (\alpha^m, \beta^m)$ are given as a function of the iteration index $m$. One observes very fast convergence of the sliding interpolation points to their limit value. Since $h = 1$ in this numerical simulation, we note that the convergence behavior is in some sense better than the one predicted by Proposition 2.5 which was restricted to small enough values of $h$.

The next series of numerical results illustrate the convergence estimate of Proposition 2.6 for the positive cubic approximation of a given function. For different values of $h$, we consider the functions

$$f_h(x) = \frac{1}{1-hx}, \quad 0 \le x \le 1, \quad h = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \cdots \tag{4.5}$$

The results are given in Table 4.2. 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 $m$. In agreement with Proposition 2.6, we find that for $m = 0$, the accuracy is second order and for $m = 1$ and beyond, it is fourth order. It cannot be greater than fourth order since this is the optimal rate for the approximation with cubic polynomials. We also note

Table 4.1
*Algorithm* 2.1: *Convergence of the sliding interpolation points for the function* (4.4).

| $m$ | $\alpha^m$ | $\beta^m$ |
|---|---|---|
| 0 | 0.250000, | 0.750000 |
| 1 | 0.290569 | 0.747017 |
| 2 | 0.290678 | 0.747013 |
| ... | 0.290678 | 0.747013 |

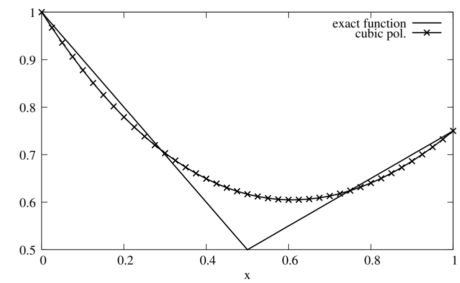


Table 4.2
*Relative* $L^\infty$ *errors between the function* (4.5) *and its approximated cubic interpolant, with the reduction factors. The observed convergence order is in accordance with Proposition* 2.6. *The last column with* $m = 2$ *shows no improvement with respect to* $m = 1$, *as expected.*

| $h$ | $m = 0$ | | $m = 1$ | | $m = 2$ |
|---|---|---|---|---|---|
| 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 | | $\approx 2$ | | $\approx 4$ | |

that our approximations with positive polynomials are slightly less accurate than the one with Chebyshev interpolation. This observation can be related to the theoretical bound $\inf_{p_n\in P_n^+} \|f-p_n\| \leq 2\inf_{q_n\in P_n} \|f-q_n\|$ that can be derived from Theorem 1.2 in [3]. A comparison of our results with additional simulations using other approximation methods (not shown here) indicates that this factor 2 is asymptotically satisfied.

**4.2.2. 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 (A.6)). The initial point is (3.16), and the algorithm is always well defined since the Jacobian $J_p(X^0)$ is a nonsingular matrix according to Lemma 3.7. The sliding nodes are well separated provided $h$ is small enough, as explained in section 3.1.4 and Proposition 3.10. However we have observed in many numerical experiments excellent convergence properties even for $h = 1$.

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

$$R(x) = \frac{1}{1 + 25(2x-1)^2} \tag{4.6}$$

by positive polynomials of degree $n = 7k$ with $k = 1, 2, 3, 4$. The function $R$ and its positive interpolation polynomials are shown in Figure 4.1: the converged position of the interpolation points is displayed with bullets. The number of iterations is always the same: $m = 20$. One observes stability and convergence of the interpolation points as $n$ increases, either for even or odd degrees.

The second series of experiments illustrates the Lukács theorem. We take a function $f \in P_{17}^+$ which is now polynomial,

$$f(x) = 10^5 x^{10}(1-x)^7 + 0.01 \tag{4.7}$$

and very close to zero at the boundaries. We use the general order Newton–Raphson

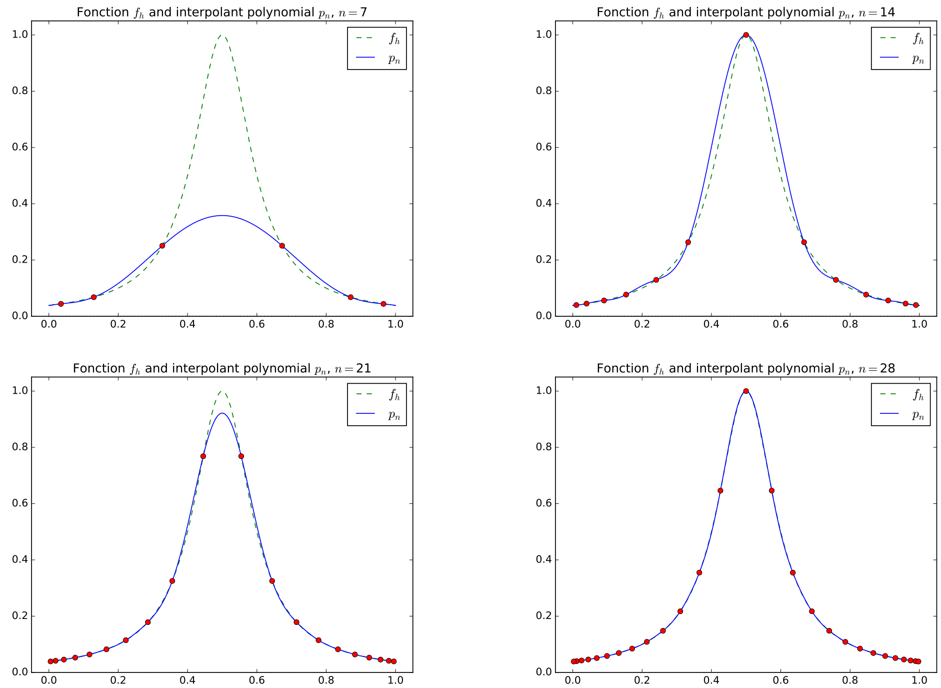


FIG. 4.1. *Degree $n = 7, 14, 21, 28$. The (Runge) function (4.6) is the dashed line. The positive interpolations are the continuous lines. Interpolation points are represented by bullets.*

TABLE 4.3
*Convergence of the interpolation polynomials to their polynomial $f$ (4.7).*

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

algorithm (3.8) until convergence. The results, in terms of normalized $L^\infty$ error, are given in Table 4.3. 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 nonnegative, which is a property of the method.

The convergence order of estimate (1.3) in Theorem 1.2, which deals with the nonnegative polynomial approximation of high order, is illustrated in Table 4.4, using the same method as in Table 4.2. The objective function is $f_h$ with $f$ given in (4.7). Here we consider odd degrees $n = 2p+1$, 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, $n+1 = 2p+2 = 2(m+1) \Longleftrightarrow m = p$.

Figure 4.2 illustrates for $n = 15$ (that is, odd degres with $p = 7$) the evolution of the nodes $\alpha = (\alpha_0, \ldots, \alpha_{p-1})$ and $\beta = (\beta_1, \ldots, \beta_p)$ and polynomials $a_p$ and $b_p$ after $m = 15$ iterations.

**4.2.3. Optimality with respect to the polynomial degree.** The approximation by positive interpolation polynomials is optimal in terms of polynomial degree. This can be visualized by comparison with another trivial positive approximation

TABLE 4.4
*Relative $L^\infty$ errors between the function $f_h(\cdot) = f(h\cdot)$ with $f$ provided in (4.7) 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, $n+1 = 2(m+1)$.*

| $h$ | $p=m=0$ | $p=m=1$ | $p=m=2$ | $p=m=3$ |
|---|---|---|---|---|
| 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 | $\approx 2$ | $\approx 4$ | $\approx 6$ | $\approx 8$ |

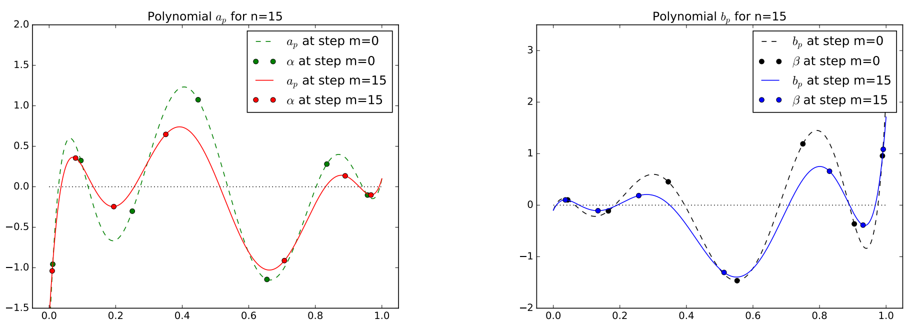


FIG. 4.2. *Evolution of $\alpha$, $\beta$, and polynomials $a_p$, $b_p$ for the function $f$ given by* (4.7).

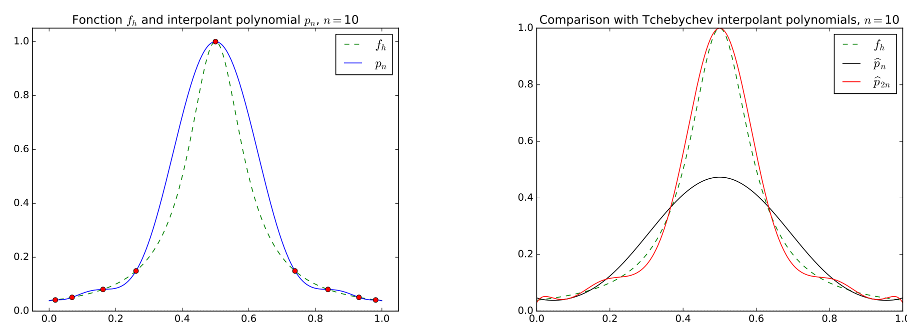


FIG. 4.3. *Approximation of $f_h = R$ by two polynomials of degree $n = 10$. Left: $p_n$ in continuous line is the positive polynomial obtained by iteration. Right: $\widehat{p}_n$ and $\widehat{p}_{2n}$ are obtained by squaring a standard Chebyshev interpolation (see* (4.8)*), using* 6 *and* 11 *interpolation points, respectively.*

which is written

$$\widehat{p}_n = \left(\mathcal{I}_p\left(\sqrt{f}\right)\right)^2, \quad n = 2p, \tag{4.8}$$

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 4.3 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. It seems however slighty less accurate than $\widehat{p}_{2n}$ which use 11 interpolation points (but is of degree 20). Here the target is again the Runge function $R$; see (4.6).

**4.2.4. Sensitivity with respect to the minimum.** The theoretical estimates in this work are based on the hypothesis that $f$ is positive over $[0, 1]$. It is therefore

TABLE 4.5
*Minimal distance at convergence ($m \gg 1$) between sliding interpolation points. The second, third, and fourth columns correspond to a maximal polynomial degree $n = 2p + 1$.*

| $\lambda$ | $p = 3$ | $p = 4$ | $p = 5$ |
|---|---|---|---|
| 1 | 0.310049 | 0.167179 | 0.110775 |
| 1e-1 | 0.221470 | 0.118254 | 0.093653 |
| 1e-2 | 0.128890 | 0.068488 | 0.070997 |
| 1e-3 | 0.069168 | 0.035922 | 0.050335 |
| 1e-4 | 0.036235 | 0.017926 | 0.033526 |
| 1e-5 | 0.018976 | 0.008733 | 0.020959 |
| 1e-5 | 0.010022 | 0.004213 | 0.012481 |
| 1e-6 | 0.005350 | 0.002028 | 0.007236 |
| 1e-7 | 0.002887 | 0.000978 | 0.004143 |

natural to assess the sensitivity of the algorithms with respect to $\min_{[0,1]} f > 0$. To do so we consider

$$f_\lambda = f + \lambda, \quad f(x) = 10^5 x^{10}(1-x)^7, \qquad \lambda > 0,$$

and we vary the parameter $\lambda$. The difficulty lies in the fact that the limit case $\lambda = 0$ cannot be directly captured by the method proposed in section 3. To understand this behavior, it is sufficient to start from the Lukács theorem written as

$$f(x) = x a_p(x)^2 + (1-x) b_p(x)^2, \quad p \geq 8.$$

A basic reasoning shows that $x^5$ factorizes both $a_p$ and $b_p$. Therefore $a_p$ and $b_p$ have the root $x_0 = 0$ with multiplicity 5. It is impossible to have at the same time an oscillating representation of $a_p$ (resp., $b_p$) as the one sketched in Figure 3.1 with $p$ distinct roots in the interval $[0,1]$. Since our representation was based on distinct roots, it means that some degeneracy of the algorithms is expected as $\lambda \to 0^+$. To assess this issue, the numerical tests below explore the sensitivity of the algorithms for decreasing values of $\lambda > 0$.

We record the minimal distance between the sliding interpolation points (for different values of $\lambda$ and $p$) at approximate convergence, that is, for large $m$,

$$\delta(\lambda, p) \approx d_-(X_\lambda^{m+1}, X_\lambda^m), \qquad m \gg 1,$$

where $d_-(X,Y) = \min_{1 \leq i \neq j \leq p} |X_i - Y_j|$ for $X = (x_1, \ldots, x_p)$ and $Y = (y_1, \ldots, y_p)$. The results in Table 4.5 show that the minimal distance scales approximately as

$$\delta(\lambda, p) \approx C(p, f)\lambda.$$

It is linear with respect to $\lambda$ and the constant depends a priori on the function which is approximated. It indicates that some roots of $a_p$ and $b_p$ have a tendency to coincide for small $\lambda > 0$. What we have always observed is that the methods developed in this work are very stable even for very small offsets (i.e., very small values of $\lambda > 0$).

**4.3. 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 section 4.4 below.

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

$$q_\lambda(x) = 10(x - 1/2)^4 + \lambda \tag{4.9}$$

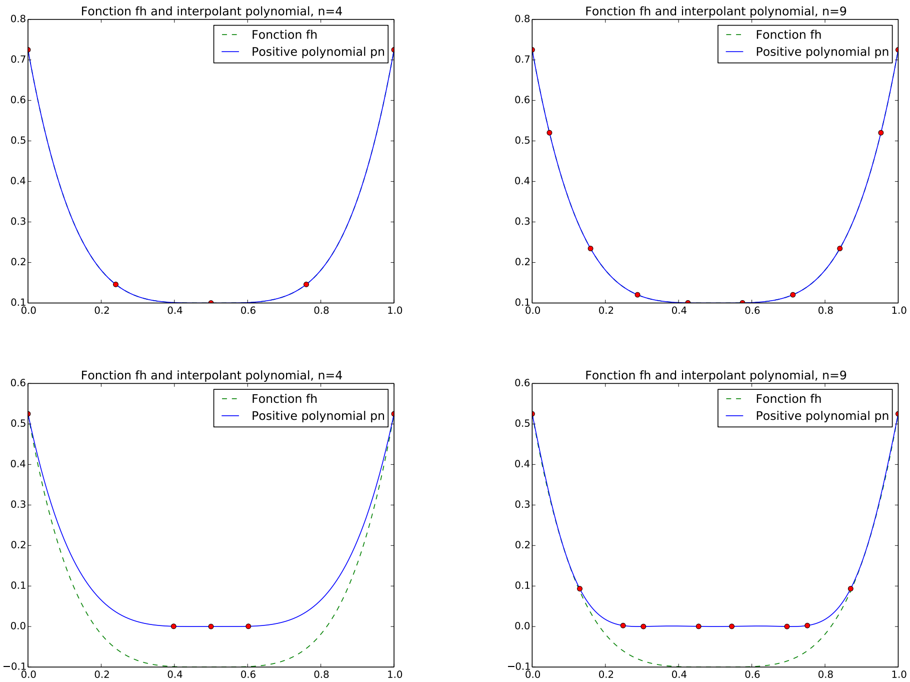


FIG. 4.4. *The objective polynomial $f_h$ can be either in $P_n^+$ (top) or not (bottom). In the case where it takes negative values, positive polynomials construct good approximations to* $\max(f_h, 0)$.

and consider different values of $\lambda$. For $\lambda \geq 0$, one clearly has $q_\lambda \in P_4^+$ and, on the other hand, for $\lambda < 0$, then $q_\lambda \notin P_4^+$. For such a polynomial it is evident to determine whether $q_\lambda(x) > 0$: 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 (4.1) 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 $q_\lambda$,
- or $q_\lambda \notin P_4^+$, then $p_4^m \in P_4^+$ can be used as a nonnegative polynomial surrogate to the objective function $q_\lambda$.

This is illustrated in Figure 4.4 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 4.4 are extremely accurate. On the other hand for $\lambda = -0.1 < 0$, the two bottom figures in Figure 4.4 show that positive polynomials have the ability to capture a very good (polynomial) nonnegative 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).

**4.4. Numerical approximation of the advection equation.** This section can be considered as an ultimate justification for the introduction of the parameter $h$ in our various approximation results, such as the main one, Theorem 1.2. This parameter is now proportional to the mesh size $\Delta x > 0$ used to discretize partial

TABLE 4.6
*h-convergence with respect to the polynomial degree n for the positive semi-Lagrangian scheme.*

| $h$ | $n=1$ | $n=3$ | $n=5$ | $n=7$ |
|---|---|---|---|---|
| 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 | $\approx 1$ | $\approx 3$ | $\approx 5$ | $\approx 7$ |

TABLE 4.7
*h-convergence with respect to the polynomial degree n for the positive conservative semi-Lagrangian scheme. A gain of one convergence order is observed, compared to the results reported in Table* 4.6.

| $h$ | $n=1$ | $n=3$ | $n=5$ | $n=7$ |
|---|---|---|---|---|
| 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 | $\approx 2$ | $\approx 4$ | $\approx 6$ | $\approx 8$ |

differential equations. We consider the discretization of the advection equation

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

with 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 4.3. Thus, 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 [4] where the value at the foot of the characteristic is predicted by standard Lagrangian interpolation then modified so as to get a nonnegative value. The second one is 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 yet possible to determine by theoretical means. We can only say that the stencil of the Lagrange interpolation is linearly stable in any $L^r$, $1 \leq r \leq \infty$ (see [4]), and that the guaranteed nonnegativity of the global method yields some nonlinear stability. Take as initial data $u_0(x) = \cos(\pi x)^2 + 1 > 0$. 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 4.6 we display the $L^\infty$ error at time $T_{end} = 1$ for the semi-Lagrangian implementation, as a function of the numbers of cells.

The same test problem with a conservative implementation (ENO-like reconstruction of the fluxes) yields the results in Table 4.7 with an increased convergence order.

**Appendix A. Item 4 of Theorem 1.2: The even case $n = 2p$.** For even degrees, the results are the same as for odd degrees, except 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).

**A.1. Positive interpolation as a root-finding problem.** For even degrees $n$ the sufficient criterion of Proposition 3.1 generalizes without difficulty.

PROPOSITION A.1. *Let $f \in W^{1,\infty}(0,1)$ satisfy (1.1), and let $h \geq 0$. Assume there exist two polynomials $a_{p,h} \in P_p$, $b_{p-1,h} \in P_{p-1}$ and $2p-1$ nodes $0 < \alpha_1 < \cdots < \alpha_{p-1} < 1$ and $0 < \beta_1 < \cdots < \beta_p < 1$ in $(0,1)$, with the following properties:*

(a) *the nodes are roots of the polynomials in the sense that*

$$a_{p,h}(\beta_i) = 0 \quad \textit{for} \quad 1 \leq i \leq p, \qquad b_{p-1,h}(\alpha_i) = 0 \quad \textit{for} \quad 1 \leq i \leq p-1; \tag{A.1}$$

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

$$\begin{cases} a_{p,h}(\alpha_i) = (-1)^{i+p}\sqrt{f(h\alpha_i)} & \textit{for} \quad 0 \leq i \leq p, \\ b_{p-1,h}(\beta_i) = (-1)^{i+p}\sqrt{\frac{f(h\beta_i)}{\beta_i(1-\beta_i)}} & \textit{for} \quad 1 \leq i \leq p \end{cases} \tag{A.2}$$

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

**A.2. Simplified Newton algorithms.** Algorithms (3.8) and (3.23) are naturally adapted. 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 (A.2), i.e.,

$$a_{p,h}[\alpha](x) = \sum_{0 \leq i \leq p} (-1)^{i+p}\sqrt{f(h\alpha_i)} \prod_{0 \leq j \neq i \leq p} \frac{x-\alpha_j}{\alpha_i - \alpha_j} \tag{A.3}$$

and

$$b_{p-1,h}[\beta](x) = \sum_{1 \leq i \leq p} (-1)^{i+p}\sqrt{\frac{f(h\beta_i)}{\beta_i(1-\beta_i)}} \prod_{1 \leq j \neq i \leq p} \frac{x-\beta_j}{\beta_i - \beta_j}. \tag{A.4}$$

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

$$\Gamma_{p,h}(\alpha,\beta) = \left(b_{p-1,h}[\beta](\alpha_1),\ldots,b_{p-1,h}[\beta](\alpha_{p-1}), a_{p,h}[\alpha](\beta_1),\ldots,a_{p,h}[\alpha](\beta_p)\right). \tag{A.5}$$

As in section 3 the sufficient criterion of Proposition A.1 applies as soon as $(\alpha,\beta) \in I_{p-1} \times I_p$ satisfies $\Gamma_{p,h}(\alpha,\beta) = 0$ and we introduce, starting from $X^0 \in I_{p-1} \times I_p$, the following simplified Newton–Raphson algorithms

$$X^{m+1} := G_h(X^m) \quad \text{with} \quad G_h(X) = X - [\nabla\Gamma_{p,0}(X^0)]^{-1}\Gamma_{p,h}(X^m), \tag{A.6}$$

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 guaranteed node separation then is recast as $X^{m+1} = S_{p,\varepsilon}G_h(X^m)$ with a conveniently defined operator $S_{p,\varepsilon}$. 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 $2 \times 2$ block form

$$\nabla\Gamma_{p,0}(X^0) = \begin{pmatrix} \nabla_\alpha b_{p-1,h}[\beta](\alpha) & \nabla_\beta b_{p-1,h}[\beta](\alpha) \\ \nabla_\alpha a_{p,h}[\alpha](\beta) & \nabla_\beta a_{p,h}[\alpha](\beta) \end{pmatrix}\Bigg|_{(\alpha,\beta)=X^0}. \tag{A.7}$$

**A.3. Definition of the starting point $X^0$.** Again, to define the starting point $X^0$ we rely on the Chebyshev polynomials $(T_p, U_{p-1}) \in P_p \times P_{p-1}$. We seek two polynomials $\underline{a}_p, \underline{b}_{p-1} \in P_p \times P_{p-1}$ such that $\underline{a}_p(x)^2 + x(1-x)\underline{b}_{p-1}(x)^2 = 1$ for all $x \in [0,1]$. The following lemma can be proved in the same way as Lemma 3.4.

LEMMA A.2. *Given $p \in \mathbb{N}$ let*

$$\underline{\alpha}_i := \frac{1}{2}\left[1 - \cos\left(\frac{i\pi}{p}\right)\right], \ i = 0, \dots, p, \quad \underline{\beta}_i := \frac{1}{2}\left[1 - \cos\left(\frac{(2i-1)\pi}{2p}\right)\right], \ i = 1, \dots, p. \tag{A.8}$$

*and let $\underline{a}_p$ and $\underline{b}_{p-1}$ be the polynomials defined according to* (A.3)–(A.4) *with a constant function $f = 1$. We have the following properties.*

(i) *Interlacing of the nodes: we have*

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

(ii) *Chebyshev form: the above polynomials read*

$$\underline{a}_p(x) = T_p\left(2x-1\right), \qquad \underline{b}_{p-1}(x) = 2U_{p-1}\left(2x-1\right). \tag{A.9}$$

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

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

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

$$\underline{a}_p(x)^2 + x(1-x)\underline{b}_{p-1}(x)^2 = 1. \tag{A.10}$$

For the starting point of the algorithm (A.6) we then set

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

using the reference nodes (A.8) for the even case. By differentiating (A.10) and using the values of the polynomials on these nodes we find that $\underline{a}'_p(\underline{\alpha}_i) = 0$ for $i = 0, \dots, p$ and $\underline{b}'_{p-1}(\underline{\beta}_i) = (-1)^{i+p+1}\frac{(1-2\underline{\beta}_i)}{2[\underline{\beta}_i(1-\underline{\beta}_i)]^{3/2}}$ for $i = 1, \dots, p$. Reasoning as in the proof of Lemma 3.7 we can then verify that the reference Jacobian matrix has a very simple structure and is nonsingular. We also provide explicit formulas.

LEMMA A.3. *The reference Jacobian matrix* (A.7) *has the form*

$$\nabla\Gamma_{p,0}(X^0) = \begin{pmatrix} \nabla_\alpha b_{p-1,0}[\beta](\alpha) & \nabla_\beta b_{p-1,0}[\beta](\alpha) \\ \nabla_\alpha a_{p,0}[\alpha](\beta) & \nabla_\beta a_{p,0}[\alpha](\beta) \end{pmatrix}\Bigg|_{(\alpha,\beta)=X^0} = \sqrt{f(0)}\begin{pmatrix} D_\alpha & 0 \\ 0 & D_\beta \end{pmatrix}$$

*with $D_\alpha = \operatorname{diag}\left(\underline{b}'_{p-1}(\underline{\alpha}_i) : i = 1, \dots, p-1\right)$ and $D_\beta = \operatorname{diag}\left(\underline{a}'_p(\underline{\beta}_i) : i = 1, \dots, p\right)$, two diagonal matrices given by $\underline{a}'_p(\underline{\beta}_i) = \frac{p(-1)^{p+i}}{\sqrt{\underline{\beta}_i(1-\underline{\beta}_i)}} \neq 0$ and $\underline{b}'_{p-1}(\underline{\alpha}_i) = \frac{2p(-1)^{p+i+1}}{\underline{\alpha}_i(1-\underline{\alpha}_i)} \neq 0$ .*

The convergence estimates for even degrees $n = 2p$ take the same form as for odd degrees, which allows us to state Theorem 1.2 as a general result.

REFERENCES

[1] S. Butt and K. W. Brodlie, *Preserving positivity using piecewise cubic interpolation*, Comput. Graph., 17 (1993), pp. 55–64.

[2] F. Charles, M. Campos Pinto, and B. Després, *Algorithms for Positive Polynomial Approximation*, preprint, http://hal.upmc.fr/hal-01527763 (2017).

[3] B. Després, *Polynomials with bounds and numerical approximation*, Numer. Algorithms, (2017), pp. 1–31.

[4] B. Després, *Uniform asymptotic stability of Strang's explicit compact schemes for linear advection*, SIAM J. Numer. Anal., 47 (2009), pp. 3956–3976.

[5] P. Deuflhard, *Newton Methods for Nonlinear Problems*, Springer Ser. Comput. Math. 35, Springer, Berlin, 2011.

[6] H. Hong and D. Jakus, *Testing positiveness of polynomials*, J. Automat. Reason., 21 (1998), pp. 23–38.

[7] J.-B. Lasserre, *Moments, Positive Polynomials and Their Applications*, Imperial College Press, London, 2010.

[8] G. V. Milovanovic, D. S. Mitrinovic, and T. M. Rassias, *Topics in Polynomials: Extremal Problems, Inequalities, Zeros*, World Scientific, River Edge, NJ, 1994.

[9] V. Powers, *Positive Polynomials and Sums of Squares: Theory and Practice*, http://www.mathcs.emory.edu/~vicki/preprint/PsdSosSurvey.pdf (2015).

[10] J. J. Risler, *Mathematical methods for CAD*, Cambridge University Press, Cambridge, 1992.

[11] J. J. Risler, *Computer aided geometric design*, in Handbook of Numerical Anaysis, Vol. V, North-Holland, Amsterdam, 1997, pp. 715–818.

[12] J.-W. Schmidt and W. Hess, *Positivity of cubic polynomials on intervals and positive spline interpolation*, BIT, 28 (1988), pp. 340–352.

[13] C. W. Shu, *Bound-preserving high order accurate schemes*, CMS Notes, 45 (2013), pp. 24–25.

[14] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, eds., NIST Handbook of Mathematical Functions, Cambridge University Press, Cambridge, 2010.

[15] G. Szegö, *Orthogonal Polynomials*, American Mathematical Society, Providence, RI, 1939.

[16] E. F. Toro, *Riemann Solvers and Numerical Methods in Fluid Dynamics, A Practical Introduction*, Springer, Berlin, 1997.

[17] J.-H. Zhang and Z.-X.Yao, *Optimized explicit finite-difference schemes for spatial derivatives using maximum norm*, J. Comput. Phys., 250 (2013), pp. 511–526.