Bivariate range functions with superior convergence order

Bingwei Zhang · Thomas Chen · Kai Hormann · Chee Yap

Abstract

Range functions are a fundamental tool for certified computations in geometric modelling, computer graphics, and robotics, but traditional range functions have only quadratic convergence order ($m = 2$). For "superior" convergence order (i.e., $m > 2$), we exploit the Cornelius–Lohner framework in order to introduce new bivariate range functions based on Taylor, Lagrange, and Hermite interpolation. In particular, we focus on practical range functions with cubic and quartic convergence order. We implemented them in Julia and provide experimental validation of their performance in terms of efficiency and efficacy.

Citation Info

Journal
Computer Aided Geometric Design

Volume
128, August 2026

Article
102569, 17 pages

Note
Proceedings of GMP

DOI
10.1016/j.cagd.2026.102569

1 Introduction

Many problems in geometric modelling, computer graphics, and robotics, such as surface-surface intersection, constructive solid geometry (CSG) operations, ray-tracing, and collision detection, can be solved robustly using interval analysis and range (or inclusion) functions [3, 22, 8]. The core idea of range functions is to calculate an interval that is guaranteed to enclose the range of a function f over any given domain B, where B is typically a box, simplex, or ball in Euclidean space [19]. Range functions provide various certified predicates for finding roots of f. One such predicate is the exclusion predicate of root finding: if the estimated range of f over B does *not* contain zero, then f has no roots in B. With good range functions, provably near-optimal complexity bounds can be achieved for isolating real and complex roots [20, 2]. Combined with subdivision, range functions lead to very efficient branch-and-bound algorithms for the problems mentioned earlier [15, 21]. In particular, *bivariate* range functions have been used for the isotopic approximation of curves [18, 13, 14], and they can be used for finding the roots of mappings from $\mathbb{R}^2$ to $\mathbb{R}^2$ [4].

Clearly, the efficiency of these algorithms depends on (1) the efficiency of evaluating the range functions and (2) the efficacy (or tightness) of the estimated ranges. A simple measure of tightness is the order of convergence. The classic approach to range functions for f is based on the Taylor expansion of f to some degree $n \geq 1$. Unfortunately, the order of convergence remains quadratic ($m = 2$), regardless of n [19]. Cornelius and Lohner [6] introduced the "CL framework" that finally achieves *superior* convergence orders (i.e., $m > 2$). While they can only reach convergence orders of up to $m \leq 6$, due to the underlying implicit computational model, Hormann, Yap, and Zhang [12, Theorems B and C] showed how to achieve convergence order m for any $m > 1$ by generalizing the CL framework.

Superior convergence orders have an efficiency cost. So in any application, one must choose a "sweet spot" convergence order that achieves practical trade-offs among efficiency, efficacy, and ease of implementation. Typically, the sweet spot is some small m, say $m < 6$. For example, for the EVAL root isolation algorithm [11], such considerations led to the sweet spot order of $m = 3$. Thus, in our current implementations, we focus on convergence orders up to $m \leq 4$. Although our techniques are in principle able to reach larger values of m, it is clear from our development below that their implementation will be much more involved.

In this paper, we focus on the bivariate setting and derive new range functions that advance the state of the art in terms of convergence order. Specifically, we introduce bivariate range functions with cubic and quartic convergence orders that are based on Taylor, Lagrange, and Hermite interpolation. The geometric modelling literature has a well-known "Bernstein range" for any polynomial f over a box B, namely the interval between the minimum and maximum Bernstein coefficients over B [9]. However, our experiments will not compare this Bernstein range function with our range functions. We offer two justifications for this omission. First, the Bernstein approach is fundamentally different from the interpolation approach of our methods, since it works only for polynomials and we know of no range functions based on the Bernstein approach with a superior convergence order. Second, the comprehensive study of Martin et al. [15] compared a set of 30 range functions for tracing planar curves, and they concluded that the Bernstein method performs

comparably to the centred form Taylor method, but the latter is easier to implement. This centred form Taylor method is just $\square^T_{2,n}$ in our experiments.

1.1 Range functions

For any bivariate real function $f:\mathbb{R}^2\to\mathbb{R}$ and any $S\subseteq\mathbb{R}^2$, the *range* of f on S is the set $f(S):=\{f(x):x\in S\}$. With $\square\mathbb{R}^2$ denoting the set of closed, axis-aligned rectangles, also called *boxes* and $\square\mathbb{R}$ denoting the set of closed intervals, we call $\square f:\square\mathbb{R}^2\to\square\mathbb{R}$ a *range function* for f, if $\square f(B)\supseteq f(B)$ for all $B\in\square\mathbb{R}^2$. For any box $B=[a_x,b_x]\times[a_y,b_y]\in\square\mathbb{R}^2$, the *midpoint*, *radius*, and (maximal) *width* of B are defined as $\boldsymbol{m}=(m_x,m_y):=(a_x+b_x,a_y+b_y)/2$, $\boldsymbol{r}=(r_x,r_y):=(b_x-a_x,b_y-a_y)/2$, and $w(B):=2\max(r_x,r_y)$. We further define the *width* and the *magnitude* of any $I=[a,b]\in\square\mathbb{R}$ as $w(I):=b-a$ and $|I|:=\max\{|a|,|b|\}$.

For any $m\geq 1$, we say that the range function $\square f$ exhibits *order m convergence* on $B_0\in\square\mathbb{R}^2$, if there exists a constant $C>0$ that may depend on f and B_0, such that

$$d_{\mathrm{H}}(f(B),\square f(B))\leq C\,w(B)^m \tag{1}$$

for all $B\subset B_0$, where $d_{\mathrm{H}}([a,b],[a',b']):=\max\{|a-a'|,|b-b'|\}$ denotes the Hausdorff distance between intervals.

For a rational function f, given in terms of a specific *expression* (or *formula*) $E(x,y)$, the simplest range function over $B=I\times J$ with $I,J\in\square\mathbb{R}$ is the *natural interval extension* [16, Section 3.3] $\square f(B):=E(I,J)$, obtained by replacing the variables with intervals and utilizing interval arithmetic during the evaluation of the expression E. If $\square f(B)$ is well-defined for all $B\subseteq B_0$, then $\square f$ is convergent over B_0 [16, Chapter 4], but the convergence is only of order 1 in general. Quadratic convergence is instead achieved by *centred forms*, for example, the celebrated *mean-value form* [19, Chapter 3].

The breakthrough to achieve range functions with superior convergence comes from the CL framework [6]. The idea is to split f into two parts $f(x)=g(x)+R_g(x)$ where g is called the *exact part* and $R_g:=f-g$ the *remainder part*, and for any interval I, we compute $g(I)$ exactly while $R_g(I)$ can be approximated by some range function $\square R_g(I)$.

Theorem 1 [6, Theorem 4]. If the function $f:\mathbb{R}\to\mathbb{R}$ is continuous over $I_0\in\square\mathbb{R}$ and can be written as $f(x)=g(x)+R_g(x)$ for any $x\in I_0$, then the range function

$$\square f(I):=g(I)+\square R_g(x) \tag{2}$$

satisfies

$$d_{\mathrm{H}}(f(I),\square f(I))\leq w(\square R_g(I))$$

for any $I\subseteq I_0$.

In particular, the convergence order of $\square f(I)$ is that of $\square R_g(I)$, and we know how to achieve superior convergence orders for $\square R_g$. The problem with (2) is that there are no[1] practical arithmetic models (standard models of Numerical Analysis, IEEE standard, bigNumber packages, etc.) for computing $g(I)$ exactly, except for very special g. For example, $g(x)=3x^2+1$ is okay, but $g(x)=3x^2+x$ is not. For that reason, Hormann, Yap, and Zhang [12] proposed allowing g to be approximated by a range function $\square g$.

Theorem 2 [12, Theorem B]. If the function $f:\mathbb{R}^2\to\mathbb{R}^2$ is continuous over $B_0\in\square\mathbb{R}^2$ and can be written as $f(\boldsymbol{x})=g(\boldsymbol{x})+R_g(\boldsymbol{x})$ for any $\boldsymbol{x}\in B_0$, then the range function

$$\square f(B):=\square g(B)+\square R_g(B) \tag{3}$$

satisfies

$$d_{\mathrm{H}}(f(B),\square f(B))\leq d_{\mathrm{H}}(g(B),\square g(B))+w(\square R_g(B))$$

for any $B\subseteq B_0$.

[1] Cornelius and Lohner [6] assumed that if we have an algebraic expression for $g(x)$, then we can compute its roots exactly. For example, if $g(x)$ is a polynomial of degree at most 4, then we can compute its roots. Although this is a common mathematical model, there are no practical exact implementations.

Note that while Hormann, Yap, and Zhang [12] proved this theorem for univariate functions, their proof carries over to the multivariate setting without major changes and is stated here for bivariate functions, as needed below. By (1), an immediate consequence of Theorem 2 is that the *generalized Cornelius–Lohner form* (3) has order-m convergence, provided we ensure

$$d_{\mathrm{H}}(g(B), \square g(B)) \leq w(\square R_g(B)) \leq C\, w(B)^m. \tag{4}$$

For a large class of functions, it is possible to compute range functions $\square g(B)$ that satisfy (4) [12]. For example, if $g(x)$ is a polynomial, then we need the ability to approximate the roots ξ of the derivative g' to any desired error $\delta > 0$. Then, for any $\varepsilon > 0$, we can determine $\delta = \delta(\varepsilon)$ so that $|\xi - \tilde{\xi}| \leq \delta$ implies $|g(\tilde{\xi}) - g(\xi)| \leq \varepsilon$. The values $g(\tilde{\xi})$ can then be used to determine $g(I)$ to within ε.

1.2 Contributions and overview

In this paper, we propose and discuss three novel bivariate range functions based on the generalized Cornelius–Lohner framework. The first (Section 2) stems from the Taylor expansion of f about the midpoint of the interval and generalizes the quadratically convergent centred Taylor forms [19, Section 3.5]. The second (Section 3) and third (Section 4) apply the idea of Cornelius and Lohner recursively and are inspired by the corresponding univariate constructions in [11] and [12], respectively. Our numerical experiments (Section 5) confirm the theoretically proven convergence orders of the proposed range functions and identify those with cubic convergence order ($m = 3$) as the sweet spot, that is, the best compromise between efficiency and efficacy. In the appendices, we provide exact expressions for the endpoints of $g(B)$ for low degree polynomials g in one variable (Appendix A) or two variables (Appendix B).

2 Range functions based on centred Taylor expansions

Abbreviating the higher order *partial derivatives* of $f : \mathbb{R}^2 \to \mathbb{R}$ by

$$f^{(i,j)} := D^{(i,j)} f = \frac{\partial^{\,i+j}}{\partial x^{\,i} \partial y^{\,j}} f,$$

and denoting the *k-th differential* of f by

$$d^k f := \left(h_x \frac{\partial}{\partial x} + h_y \frac{\partial}{\partial y} \right)^k f = \sum_{j=0}^{k} \binom{k}{j} h_x^{k-j} h_y^j f^{(k-j,j)},$$

we recall *Taylor's theorem* [7], by which any $(n+1)$-times continuously differentiable function f, where $n \in \mathbb{N}_0$, can be expressed as the *Taylor expansion*

$$f(x, y) = T_n(x, y) + R_n,$$

that is, as the sum of the *n-th degree Taylor polynomial* of f about (u, v),

$$T_n(x, y) = f(u, v) + \sum_{k=1}^{n} \frac{1}{k!} d^k f(u, v) \qquad \text{with} \qquad h_x = x - u, \quad h_y = y - v, \tag{5}$$

and the *remainder*

$$R_n = \frac{1}{(n+1)!} d^{n+1} f(u + \theta h_x, v + \theta h_y)$$

for some $\theta \in (0, 1)$ (see Figure 1).

Now let $B_0 \in \square\mathbb{R}^2$ and $w_0 = w(B_0)$. For any $B \subseteq B_0$ and $m, n \in \mathbb{N}$ with $m \leq n$, let us consider the Taylor expansion of f about the midpoint $\boldsymbol{m}$ of B and split off the first m terms from the Taylor polynomial T_{n-1}. For any $\boldsymbol{x} = (x, y) \in B$, we then have

$$f(\boldsymbol{x}) = T_{m-1}(\boldsymbol{x}) + \sum_{k=m}^{n-1} \frac{1}{k!} d^k f(\boldsymbol{m}) + \frac{1}{n!} d^n f(\xi_{\boldsymbol{x}})$$

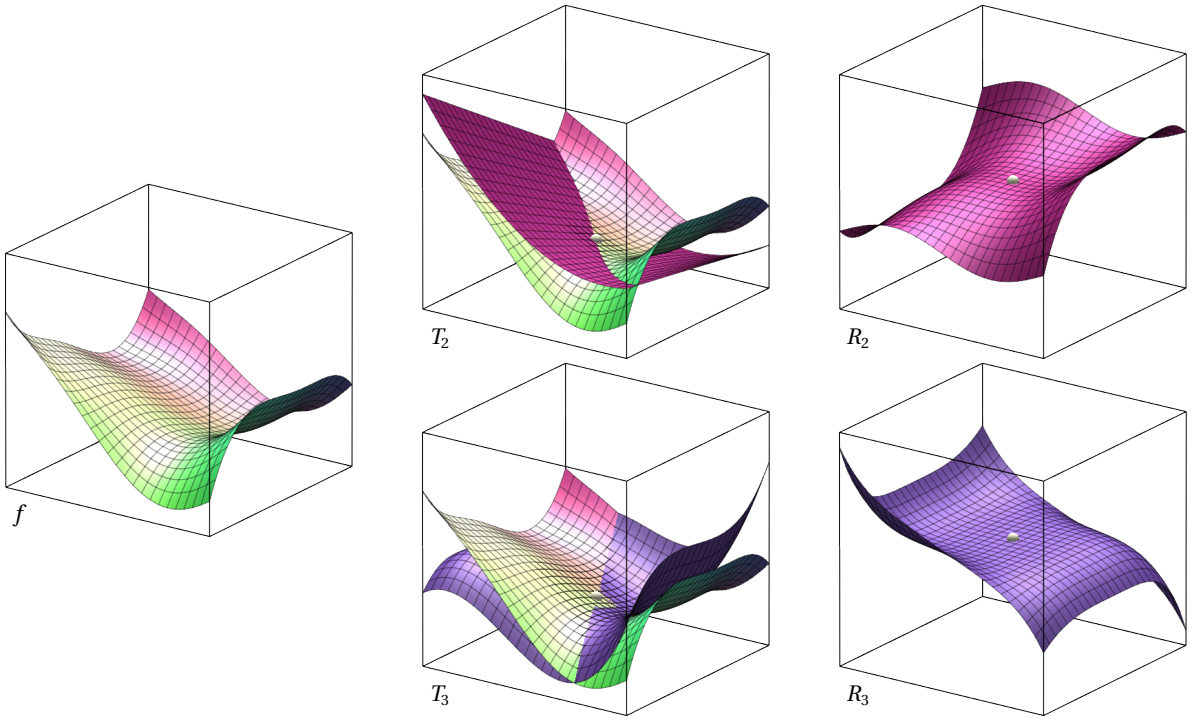


Figure 1: Graph of the function $f(x,y) = 12xy^4 + 12x^2 + 6xy + 4\sin(\pi x) + 3\sin(\pi y) - 15$ over $[-1,1]^2$ (left), its quadratic and cubic Taylor polynomials about the midpoint $(0,0)$ (centre), and the corresponding remainders (right). The vertical range of all plot boxes is $[-30,30]$.

for some $\boldsymbol{\xi}_{\boldsymbol{x}} \in B$ that depends on $\boldsymbol{x}$. Assume for simplicity that B is a *square* with $r := r_x = r_y$. Since

$$h_x = x - m_x \in r[-1,1], \qquad h_y = y - m_y \in r[-1,1],$$

we can estimate the exact range of f over B by the *Taylor form of order m and level n*,

$$\square^{\mathrm{T}}_{m,n} f(B) := T_{m-1}(B) + r^m[-1,1]S_{m,n}, \qquad S_{m,n} := \sum_{k=m}^{n} s_k r^{k-m}, \tag{6}$$

where the s_k are defined as

$$s_k := \frac{1}{k!}\sum_{j=0}^{k}\binom{k}{j}|f^{(k-j,j)}(\boldsymbol{m})|, \quad k = m,\ldots,n-1, \qquad s_n := \frac{1}{n!}\sum_{j=0}^{n}\binom{n}{j}|\square f^{(n-j,j)}(B)|$$

and $\square f^{(n-j,j)}$, $j = 0,\ldots,n$ are bounded range functions over B_0.

Theorem 3. *The Taylor form $\square^{\mathrm{T}}_{m,n} f$ in (6) has order m convergence.*

Proof. Since $B \subseteq B_0$, we have $r = w(B)/2 \le w_0/2$ and

$$s_k \le \frac{1}{k!}\sum_{j=0}^{k}\binom{k}{j}|f^{(k-j,j)}(B_0)| =: \bar{s}_k, \quad k = m,\ldots,n-1.$$

Moreover, since the range functions $\square f^{(n-j,j)}$ are bounded over B_0 there exist constants C_j, $j = 0,\ldots,n$, such that

$$s_n \le \frac{1}{n!}\sum_{j=0}^{n}\binom{n}{j}C_j =: \bar{s}_n.$$

Using these bounds, we conclude

$$S_{m,n} \le C', \qquad C' := \sum_{k=0}^{n-m}\Big(\frac{w_0}{2}\Big)^k \bar{s}_{k+m},$$

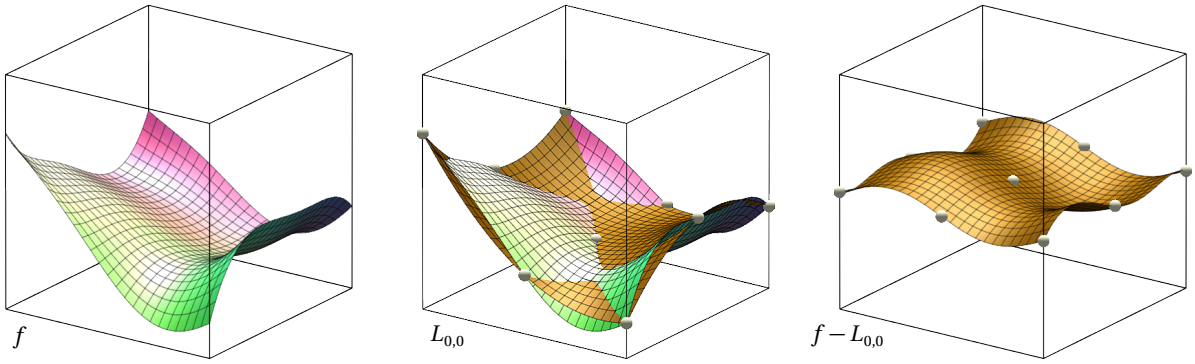


Figure 2: Same function as in Figure 1 (left), its biquadratic Lagrange interpolant (centre), and the remainder (right).

where the constant C' depends only on f and B_0.

Noticing that $\square^{\mathrm{T}}_{m,n} f$ is a special case of the generalized Cornelius–Lohner form (3) with $\square g = g = T_{m-1}$ and $\square R_g = r^m[-1,1]S_{m,n}$, the order m convergence of $\square^{\mathrm{T}}_{m,n} f$ then follows from Theorem 2 and (4), because

$$w(\square R_g(B)) \le 2|\square R_g(B)| = 2r^m S_{m,n} \le C\, w(B)^m$$

with $C := 2^{1-m} C'$. □

Note that $\square^{\mathrm{T}}_{1,n} f$ is called "Taylor form of *order n*" by Ratschek and Rokne [19, Definition 3.4], but "n-th order" does not imply "order n convergence" in their terminology [19, Section 2.4]. Instead, we prefer referring to this parameter as the *level* of the Taylor form. The quadratic convergence of these Taylor forms (including the mean-value form $\square^{\mathrm{T}}_{1,1} f$), which was proven by Ratschek and Rokne [19, Theorem 3.4], is also asserted by Theorem 3, because the exact range of the linear Taylor polynomial T_1 (see Appendix B.1) is

$$T_1(B) = f(\boldsymbol{m}) + r[-1,1]\big(|f^{(1,0)}(\boldsymbol{m})| + |f^{(0,1)}(\boldsymbol{m})|\big) = T_0(B) + r[-1,1]s_1,$$

hence $\square^{\mathrm{T}}_{1,n} f = \square^{\mathrm{T}}_{2,n} f$.

The advantage of our generalized Taylor forms is that they can achieve any convergence order in theory. In practice, however, one will probably make do with cubic or quartic convergence, because the exact ranges $T_2(B)$ and $T_3(B)$ of the quadratic and cubic Taylor polynomials can be determined by simple algorithms (see Appendices B.2 and B.4).

If f is a polynomial of degree d, so that all partial derivatives of order $k > d$ vanish, then we call $\square^{\mathrm{T}}_m f(B) := \square^{\mathrm{T}}_{m,d+1} f(B)$ with $s_{d+1} = 0$ the *maximal Taylor form of order m*. This form depends only on the radius r of the square B and the $(d+1)(d+2)/2 \in O(d^2)$ partial derivatives of f of orders $0, 1, \ldots, d$, evaluated at $\boldsymbol{m}$.

3 Recursive Lagrange form

Another approach to construct range functions with cubic convergence is based on repeated Lagrange interpolation, an idea that was explored in the univariate setting by Hormann, Kania, and Yap [11]. To derive the bivariate version of their recursive Lagrange form, we recall a result by Mößner and Reif [17, Theorem 2.1]. It asserts that the error between f and the biquadratic polynomial $L_{0,0}$ (see Figure 2) that interpolates f at the corners, the edge-midpoints, and the midpoint of a square B with radius $\boldsymbol{r} = (r, r)$, that is, at the regular 3×3 grid $G(B) = \{m_x - r, m_x, m_x + r\} \times \{m_y - r, m_y, m_y + r\}$ (see Appendix B.3), is bounded from above by

$$E^{\mathrm{L}}_{0,0} := \Omega_{\mathrm{L}} \cdot \big(|f^{(3,0)}(B)| + |f^{(0,3)}(B)|\big) + \Omega^2_{\mathrm{L}} \cdot |f^{(3,3)}(B)|, \qquad \Omega_{\mathrm{L}} := \frac{\sqrt{3}}{27} r^3,$$

where the constant Ω_{L} depends only on r. That is,

$$|f(\boldsymbol{x}) - L_{0,0}(\boldsymbol{x})| \le E^{\mathrm{L}}_{0,0},$$

for all $\boldsymbol{x} \in B$, hence

$$f(B) \subseteq L_{0,0}(B) + [-1,1]E^{\mathrm{L}}_{0,0}. \tag{7}$$

To further bound $E^{\mathrm{L}}_{0,0}$, we simply apply the same idea to the functions $f^{(3,0)}$ and $f^{(0,3)}$. That is, we determine the biquadratic polynomials $L_{1,0}$ and $L_{0,1}$ that interpolate $f^{(3,0)}$ and $f^{(0,3)}$, respectively, at $G(B)$, and conclude that

$$|f^{(3,0)}(B)| \le |L_{1,0}(B)| + E^{\mathrm{L}}_{1,0}, \qquad |f^{(0,3)}(B)| \le |L_{0,1}(B)| + E^{\mathrm{L}}_{0,1},$$

where

$$E^{\mathrm{L}}_{1,0} := \Omega_{\mathrm{L}} \cdot \big(|f^{(6,0)}(B)| + |f^{(3,3)}(B)|\big) + \Omega_{\mathrm{L}}^2 \cdot |f^{(6,3)}(B)|,$$
$$E^{\mathrm{L}}_{0,1} := \Omega_{\mathrm{L}} \cdot \big(|f^{(3,3)}(B)| + |f^{(0,6)}(B)|\big) + \Omega_{\mathrm{L}}^2 \cdot |f^{(3,6)}(B)|.$$

Consequently,

$$E^{\mathrm{L}}_{0,0} \le \Omega_{\mathrm{L}} \cdot \big(|L_{1,0}(B)| + |L_{0,1}(B)|\big) + \Omega_{\mathrm{L}}^2 \cdot \big(|f^{(6,0)}(B)| + 3|f^{(3,3)}(B)| + |f^{(0,6)}(B)|\big) + \Omega_{\mathrm{L}}^3 \cdot \big(|f^{(6,3)}(B)| + |f^{(3,6)}(B)|\big).$$

If f has bounded partial derivatives up to order $3n+3$, then we may repeat this procedure to obtain the upper bound

$$E^{\mathrm{L}}_{0,0} \le \sum_{k=1}^{n-1} \Omega_{\mathrm{L}}^k \sum_{j=0}^{k} \begin{bmatrix} k \\ j \end{bmatrix} |L_{k-j,j}(B)| + \Omega_{\mathrm{L}}^n \sum_{j=0}^{n} \begin{bmatrix} n \\ j \end{bmatrix} |f^{(3(n-j),3j)}(B)| + \Omega_{\mathrm{L}}^{n+1} \sum_{j=1}^{n} \begin{bmatrix} n-1 \\ j-1 \end{bmatrix} |f^{(3(n+1-j),3j)}(B)|, \tag{8}$$

where $L_{i,j}$ denotes the biquadratic polynomial that interpolates $f^{(3i,3j)}$ at the grid $G(B)$. The coefficients

$$\begin{bmatrix} n \\ k \end{bmatrix} := D(n-k,k) = \sum_{i=0}^{k} \binom{k}{i} \binom{n-k}{i} 2^i, \qquad 0 \le k \le n, \tag{9}$$

are derived from the *Delannoy numbers* $D(m,n)$. For $k=0$ and $k=n$, (9) implies $\left[\begin{smallmatrix} n \\ 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} n \\ n \end{smallmatrix}\right] = 1$. For other values of k, these coefficients satisfy a recurrence relation that is similar to the one for binomial coefficients,

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n-1 \\ k-1 \end{bmatrix} + \begin{bmatrix} n-1 \\ k \end{bmatrix} + \begin{bmatrix} n-2 \\ k-1 \end{bmatrix}, \qquad 0 < k < n.$$

Arranging them in successive rows for $n = 0, 1, \ldots$ gives the *Tribonacci triangle* [1]

$$\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & & 1 & & & & & \\
 & & & & 1 & & 3 & & 1 & & & & \\
 & & & 1 & & 5 & & 5 & & 1 & & & \\
 & & 1 & & 7 & & 13 & & 7 & & 1 & & \\
 & 1 & & 9 & & 25 & & 25 & & 9 & & 1 & \\
\iddots & & & & & & \vdots & & & & & & \ddots
\end{array}$$

which is similar to *Pascal's triangle*.

Finally, we express each biquadratic Lagrange interpolant as $L_{i,j} = T^{\mathrm{L}}_{i,j} + R^{\mathrm{L}}_{i,j}$, that is, as the sum of the second degree Taylor polynomial of $L_{i,j}$ about $\boldsymbol{m}$ (cf. Equation (5)),

$$T^{\mathrm{L}}_{i,j}(\boldsymbol{x}) = L_{i,j}(\boldsymbol{m}) + h_x L^{(1,0)}_{i,j}(\boldsymbol{m}) + h_y L^{(0,1)}_{i,j}(\boldsymbol{m}) + \frac{1}{2} h_x^2 L^{(2,0)}_{i,j}(\boldsymbol{m}) + h_x h_y L^{(1,1)}_{i,j}(\boldsymbol{m}) + \frac{1}{2} h_y^2 L^{(0,2)}_{i,j}(\boldsymbol{m}), \tag{10}$$

where $\boldsymbol{x} = (x,y) \in B$, $h_x = x - m_x$, and $h_y = y - m_y$, and the remainder

$$R^{\mathrm{L}}_{i,j}(\boldsymbol{x}) = \frac{1}{2} h_x^2 h_y L^{(2,1)}_{i,j}(\boldsymbol{m}) + \frac{1}{2} h_x h_y^2 L^{(1,2)}_{i,j}(\boldsymbol{m}) + \frac{1}{4} h_x^2 h_y^2 L^{(2,2)}_{i,j}(\boldsymbol{m}). \tag{11}$$

Note that simple algorithms for computing the exact ranges $T^{\mathrm{L}}_{i,j}(B)$ and $R^{\mathrm{L}}_{i,j}(B)$ exist (see Appendices B.2 and B.3). It then follows from (7) and (8) that $f(B)$ can be estimated by the *recursive Lagrange form of order* 3 *and level* n,

$$\square^{\mathrm{L}}_{3,n} f(B) := T^{\mathrm{L}}_{0,0}(B) + R^{\mathrm{L}}_{0,0}(B) + [-1,1] U_{3,n}, \qquad U_{3,n} := \sum_{k=1}^{n+1} u_k \Omega_{\mathrm{L}}^k, \tag{12}$$

where the u_k are defined as

$$u_k := \sum_{j=0}^{k} \begin{bmatrix} k \\ j \end{bmatrix} |T^{\mathrm{L}}_{k-j,j}(B) + R^{\mathrm{L}}_{k-j,j}(B)|, \quad k = 1, \ldots, n-1,$$

and

$$u_n := \sum_{j=0}^{n} \begin{bmatrix} n \\ j \end{bmatrix} |\square f^{(3(n-j),3j)}(B)|, \qquad u_{n+1} := \sum_{j=1}^{n} \begin{bmatrix} n-1 \\ j-1 \end{bmatrix} |\square f^{(3(n+1-j),3j)}(B)|,$$

and $\square f^{(3i,3j)}$ for $i+j \in \{n, n+1\}$ are bounded range functions over B_0.

Theorem 4. *The recursive Lagrange form $\square^{\mathrm{L}}_{3,n} f$ in (12) has cubic convergence.*

Proof. By Theorem 5.1 in [17], there exist constants $C_{k,l}$, such that the partial derivatives of the error between f and its biquadratic Lagrange interpolant $L_{0,0}$ are bounded from above as

$$|f^{(k,l)}(\boldsymbol{x}) - L^{(k,l)}_{0,0}(\boldsymbol{x})| \le C_{k,l}\big[(2r)^{3-k}|f^{(3,l)}(B)| + (2r)^{3-l}|f^{(k,3)}(B)| + (2r)^{6-k-l}|f^{(3,3)}(B)|\big],$$

for all $\boldsymbol{x} \in B$ and any $0 \le k, l \le 2$. Since $B \subseteq B_0$ and $r = w(B)/2 \le w_0/2$, this implies

$$|L^{(k,l)}_{0,0}(\boldsymbol{m})| \le C'_{k,l},$$

where the constant

$$C'_{k,l} := |f^{(k,l)}(B_0)| + C_{k,l}\big[w_0^{3-k}|f^{(3,l)}(B_0)| + w_0^{3-l}|f^{(k,3)}(B_0)| + w_0^{6-k-l}|f^{(3,3)}(B_0)|\big]$$

depends only on f and B_0. By (11), we then have

$$|R^{\mathrm{L}}_{0,0}(B)| \le C' w(B)^3, \qquad C' := \frac{1}{16}(C'_{2,1} + C'_{1,2}) + \frac{1}{64} w(B) C'_{2,2}.$$

Similarly, we can bound the partial derivatives of the error between $f^{(3i,3j)}$ and $L_{i,j}$ and conclude from (10) and (11) that $|T^{\mathrm{L}}_{i,j}(B)|$ and $|R^{\mathrm{L}}_{i,j}(B)|$ are bounded from above by constants that depend only on f and B_0. Since the range functions $\square f^{(3i,3j)}$ are also bounded over B_0, there exist constants $\bar{u}_k$ that depend only on f and B_0, such that $u_k \le \bar{u}_k$ for $k = 1, \ldots, n+1$. Recalling that $\Omega_{\mathrm{L}} = \sqrt{3}/27 \cdot r^3$, we conclude that

$$U_{3,n} \le C'' w(B)^3, \qquad C'' := \frac{\sqrt{3}}{216} \sum_{k=0}^{n} \left(\frac{\sqrt{3}}{216} w_0^3\right)^k \bar{u}_{k+1}.$$

Noticing that $\square^{\mathrm{L}}_{3,n} f$ is a special case of the generalized Cornelius–Lohner form (3) with $\square g = g = T^{\mathrm{L}}_{0,0}$ and $\square R_g = R^{\mathrm{L}}_{0,0}(B) + [-1,1]U_{3,n}$, the cubic convergence of $\square^{\mathrm{L}}_{3,n} f$ then follows from Theorem 2 and (4), because

$$w(\square R_g(B)) \le 2|\square R_g(B)| \le 2|R^{\mathrm{L}}_{0,0}(B)| + 2U_{3,n} \le 2(C' + C'')w(B)^3.$$

□

If f is a polynomial of degree d, then we call $\square^{\mathrm{L}}_3 f(B) := \square^{\mathrm{L}}_{3,n+1} f(B)$ with $n = \lfloor d/3 \rfloor$ and $u_{n+1} = u_{n+2} = 0$ the *maximal Lagrange form of order* 3. This form depends only on the radius r of the square B and the $(n+1)(n+2)/2 \in O(d^2)$ partial derivatives $f^{(3i,3j)}$ of f with $i, j \ge 0$ and $i + j \le n$, evaluated at the 9 nodes of the grid $G(B)$, which is comparable to the $(d+1)(d+2)/2$ evaluations at the midpoint of B that are needed for the maximal Taylor forms $\square^{\mathrm{T}}_m f(B)$.

4 Recursive Hermite form

A range function with quartic convergence can be obtained similarly by deriving the bivariate version of the univariate recursive Hermite form in [12]. By Theorem 2.1 in [17], the error between f and the bicubic polynomial $H_{0,0}$ (see Figure 3) that interpolates f, $f^{(1,0)}$, $f^{(0,1)}$, and $f^{(1,1)}$ at the corners of the square B with radius $\boldsymbol{r} = (r, r)$, is bounded from above by

$$E^{\mathrm{H}}_{0,0} := \Omega_{\mathrm{H}} \cdot \big(|f^{(4,0)}(B)| + |f^{(0,4)}(B)|\big) + \Omega^2_{\mathrm{H}} \cdot |f^{(4,4)}(B)|, \qquad \Omega_{\mathrm{H}} := \frac{1}{24} r^4.$$

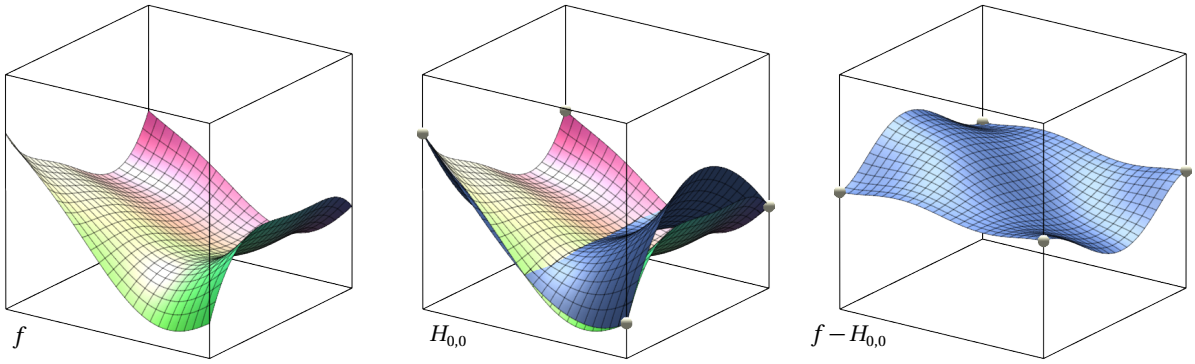


Figure 3: Same function as in Figure 1 (left), its bicubic Hermite interpolant (centre), and the remainder (right).

With the same arguments as in Section 3 and under the assumption that f has bounded partial derivatives up to order $4n+4$, we can bound $E^{\mathrm{H}}_{0,0}$ from above by

$$E^{\mathrm{H}}_{0,0} \le \sum_{k=1}^{n-1} \Omega^k \sum_{j=0}^{k} \begin{bmatrix} k \\ j \end{bmatrix} |H_{k-j,j}(B)| + \Omega^n \sum_{j=0}^{n} \begin{bmatrix} n \\ j \end{bmatrix} |f^{(4(n-j),4j)}(B)| + \Omega^{n+1} \sum_{j=1}^{n} \begin{bmatrix} n-1 \\ j-1 \end{bmatrix} |f^{(4(n+1-j),4j)}(B)|, \tag{13}$$

where $H_{i,j}$ denotes the bicubic polynomial that interpolates $f^{(4i,4j)}$, $f^{(4i+1,4j)}$, $f^{(4i,4j+1)}$, and $f^{(4i+1,4j+1)}$ at the corners of B.

As in Section 3, it remains to split each bicubic Hermite interpolant into the third degree Taylor polynomial of $H_{i,j}$ about $\boldsymbol{m}$ and the remainder, $H_{i,j} = T^{\mathrm{H}}_{i,j} + R^{\mathrm{H}}_{i,j}$, noting that the exact ranges $T^{\mathrm{H}}_{i,j}(B)$ and $R^{\mathrm{H}}_{i,j}(B)$ can be computed with simple procedures (see Appendices B.4 and B.5). We can then estimate $f(B)$ by the *recursive Hermite form of order* 4 *and level* n,

$$\square^{\mathrm{H}}_{4,n} f(B) := T^{\mathrm{H}}_{0,0}(B) + R^{\mathrm{H}}_{0,0}(B) + [-1,1] V_{4,n}, \qquad V_{4,n} := \sum_{k=1}^{n+1} v_k \Omega^k_{\mathrm{H}}, \tag{14}$$

where the v_k are defined as

$$v_k := \sum_{j=0}^{k} \begin{bmatrix} k \\ j \end{bmatrix} |T^{\mathrm{H}}_{k-j,j}(B) + R^{\mathrm{H}}_{k-j,j}(B)|, \quad k = 1, \ldots, n-1,$$

and

$$v_n := \sum_{j=0}^{n} \begin{bmatrix} n \\ j \end{bmatrix} |\square f^{(4(n-j),4j)}(B)|, \qquad v_{n+1} := \sum_{j=1}^{n} \begin{bmatrix} n-1 \\ j-1 \end{bmatrix} |\square f^{(4(n+1-j),4j)}(B)|,$$

and $\square f^{(4i,4j)}$ for $i+j \in \{n, n+1\}$ are bounded range functions over B_0.

Theorem 5. *The recursive Hermite form* $\square^{\mathrm{H}}_{4,n} f$ *in* (14) *has quartic convergence.*

Proof. By Theorem 5.1 in [17], there exist constants $C_{k,l}$, such that the partial derivatives of the error between f and its bicubic Hermite interpolant $H_{0,0}$ are bounded from above as

$$|f^{(k,l)}(\boldsymbol{x}) - H^{(k,l)}_{0,0}(\boldsymbol{x})| \le C_{k,l}\big[(2r)^{4-k}|f^{(4,l)}(B)| + (2r)^{4-l}|f^{(k,4)}(B)| + (2r)^{8-k-l}|f^{(4,4)}(B)|\big],$$

for any $0 \le k, l \le 3$, hence

$$|H^{(k,l)}_{0,0}(\boldsymbol{m})| \le C'_{k,l},$$

Analogously to the proof of Theorem 4, we conclude that there exist constants C' and C'', such that

$$|R^{\mathrm{H}}_{0,0}(B)| \le C' w(B)^4, \qquad V_{4,n} \le C'' w(B)^4,$$

and by noticing that $\square^{\mathrm{H}}_{4,n} f$ is a special case of the generalized Cornelius–Lohner form (3) with $\square g = g = T^{\mathrm{H}}_{0,0}$ and $\square R_g = R^{\mathrm{H}}_{0,0}(B) + [-1,1] V_{4,n}$, the quartic convergence of $\square^{\mathrm{H}}_{4,n} f$ then follows from Theorem 2 and (4), because

$$w(\square R_g(B)) \le 2(C' + C'') w(B)^4.$$

□

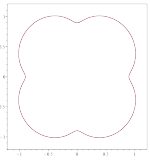
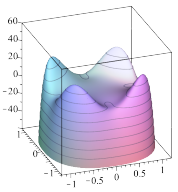
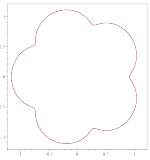
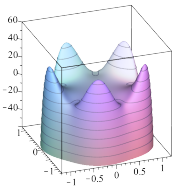
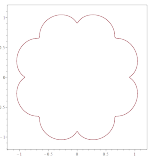
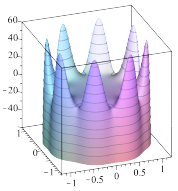
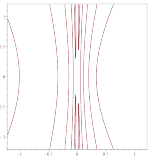
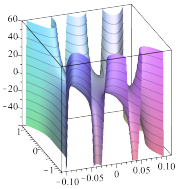
name	zero level set	graph	degree	$f(x,y)$
clover-4			10	$-50x^{10}-(249y^2-57)x^8-(498y^4-227y^2-1)x^6$ $-(498y^6-341y^4-3y^2+16)x^4$ $-(249y^8-227y^6-3y^4-102y^2-1)x^2$ $-50y^{10}+57y^8+y^6-16y^4+y^2+1$
clover-5			12	$-71x^{12}-(424y^2-79)x^{10}-(1059y^4-396y^2-1)x^8$ $-(1412y^6-793y^4-4y^2-1)x^6-20x^5$ $-(1059y^8-793y^6-6y^4-3y^2-1)x^4+202y^2x^3$ $-(424y^{10}-396y^8-4y^6-3y^4-2y^2-1)x^2$ $-101y^4x-71y^{12}+79y^{10}+y^8+y^6+y^4+y^2+1$
clover-8			18	$-156x^{18}-(1406y^2-170)x^{16}-(5625y^4-1363y^2-1)x^{14}$ $-(13125y^6-4769y^4-7y^2-1)x^{12}-(19688y^8-9538y^6-21y^4-6y^2-1)x^{10}$ $-(19688y^{10}-11922y^8-35y^6-15y^4-5y^2+30)x^8$ $-(13125y^{12}-9538y^{10}-35y^8-21y^6-11y^4-879y^2-1)x^6$ $-(5625y^{14}-4769y^{12}-21y^{10}-15y^8-11y^6+2181y^4-4y^2-1)x^4$ $-(1406y^{16}-1363y^{14}-7y^{12}-6y^{10}-5y^8-879y^6-4y^4-3y^2-1)x^2$ $-156y^{18}+170y^{16}+y^{14}+y^{12}+y^{10}-30y^8+y^6+y^4+y^2+1$
grass			12	$1+\prod_{k=1}^{6}\left((1-4^k)x^2+y^2-2x+1\right)$

Table 1: Overview of the four polynomial test functions used in our numerical experiments.

If f is a polynomial of degree d, then we call $\square_4^{\mathrm{H}} f(B) := \square_{4,n+1}^{\mathrm{H}} f(B)$ with $n = \lfloor d/4 \rfloor$ and $v_{n+1} = v_{n+2} = 0$ the *maximal Hermite form of order* 4. This form depends only on the radius r of the square B and the $(n+1)(n+2)/2 \in O(d^2)$ partial derivatives $f^{(4i+k,4j+l)}$ of f with $i, j \geq 0$ and $i+j \leq n$ and $0 \leq k, l \leq 1$, evaluated at the 4 corners of B, which is comparable to the number of evaluations needed for the maximal Taylor forms $\square_m^{\mathrm{T}} f(B)$ and the maximal Lagrange forms $\square_m^{\mathrm{L}} f(B)$.

5 Experiments

In this section, we validate the practicality and usefulness of our new range functions. We do this by reporting on the raw performance of evaluating each range function on individual boxes. We evaluate these range functions in terms of efficiency and efficacy as noted in the introduction. Our code, data, and Makefile experiments can be downloaded from our Core Library webpage [5].

Our experimental platform is a Windows 11 laptop equipped with a 2.2 GHz Intel Core i7-14650HX processor and 32 GB of RAM. All numerical experiments are implemented in Julia using standard IEEE 754 double-precision floating-point arithmetic (Float64), and timings are obtained using Julia's built-in benchmarking tools. Of independent interest is that our bivariate functions are implemented as Straight-Line Programs (SLPs) or codelists in the sense of Automatic Differentiation [10], and we perform symbolic differentiation of codelists (in contrast to the standard numeric differentiation) to improve performance. We will report the details of this implementation in a separate paper.

5.1 Test functions

Table 1 lists four bivariate polynomials for which the results reported below capture the typical behaviour observed in all our experiments. For each test function $f(x,y)$ we show the curve $f(x,y) = 0$ in the domain $B_0 := [-1.2, 1.2]^2$, as well as the graph of f, to give some intuition about the characteristics of these polynomials.

	$r = 0.1$		$r = 0.01$	
	range	d_{H}	range	d_{H}
f	$[-1.3586, -0.9646]$	—	$[-1.07788547, -1.05241970]$	—
$\square_2^{\mathrm{T}} f$	$[-1.4303, -0.6978]$	0.2667	$[-1.07824745, -1.04988220]$	0.00253750
$\square_3^{\mathrm{T}} f$	$[-1.3976, -0.8436]$	0.1209	$[-1.07792045, -1.05238265]$	0.00003705
$\square_3^{\mathrm{L}} f$	$[-1.3688, -0.8688]$	0.0958	$[-1.07789250, -1.05241267]$	0.00000703
$\square_4^{\mathrm{T}} f$	$[-1.3630, -0.9397]$	0.0249	$[-1.07788591, -1.05241719]$	0.00000252
$\square_4^{\mathrm{H}} f$	$[-1.3621, -0.9508]$	0.0138	$[-1.07788571, -1.05241821]$	0.00000149

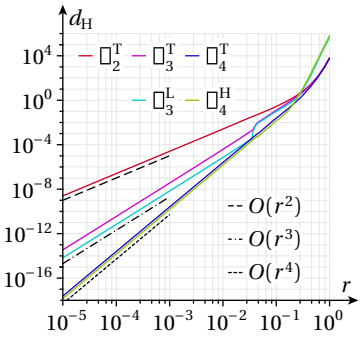


Figure 4: Concrete (rounded) values (left) and log-log plot (right) of the Hausdorff distance d_{H} between the exact range $f(B_r)$ of the test function "clover-4" over a square B_r with midpoint $(0.1, 0.2)$ and radius $r \in [0.00001, 1]$ and the approximate ranges $\square_m^{\mathrm{X}} f(B_r)$ given by maximal Taylor, Lagrange, and Hermite forms.

	$r = 0.005$		$r = 0.0005$	
	range	d_{H}	range	d_{H}
f	$[-61.874, -46.411]$	—	$[-60.5351611, -59.2710915]$	—
$\square_2^{\mathrm{T}} f$	$[-73.566, -46.367]$	11.6914	$[-60.6614110, -59.2708307]$	0.12624978
$\square_3^{\mathrm{T}} f$	$[-62.737, -46.391]$	0.8625	$[-60.5351831, -59.2710780]$	0.00002195
$\square_3^{\mathrm{L}} f$	$[-62.639, -45.980]$	0.7648	$[-60.5355311, -59.2707216]$	0.00036989
$\square_4^{\mathrm{T}} f$	$[-61.926, -46.404]$	0.0516	$[-60.5351702, -59.2710910]$	0.00000904
$\square_4^{\mathrm{H}} f$	$[-61.947, -46.360]$	0.0728	$[-60.5351657, -59.2710865]$	0.00000503

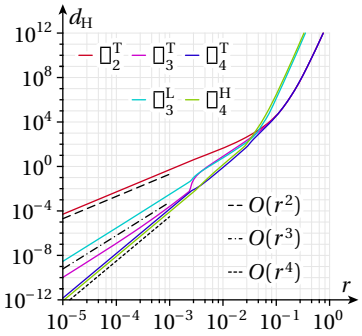


Figure 5: Data and plot for a similar experiment as in Figure 4, using the "grass" test function and a square with midpoint $(0.1, 0.1)$.

5.2 Convergence orders

Figures 4 and 5 report and compare the tightness of the five range functions

$$\square_2^{\mathrm{T}} f, \quad \square_3^{\mathrm{T}} f, \quad \square_4^{\mathrm{T}} f, \quad \square_3^{\mathrm{L}} f, \quad \square_4^{\mathrm{H}} f, \tag{15}$$

where f is the function "clover-4" (Figure 4) or "grass" (Figure 5) from Table 1 and $\square_m^{\mathrm{X}} f$ (for $\mathrm{X} \in \{\mathrm{T}, \mathrm{L}, \mathrm{H}\}$) is based on the maximal X-form of order m. We evaluate it over squares of varying radii around a specific midpoint. The plots confirm the asymptotic convergence orders of the Taylor forms of orders $m = 2, 3, 4$, the Lagrange form of order 3, and the Hermite form of order 4, as stated in Theorems 3, 4, and 5, respectively. We further observe in Figure 4, both from the plot and the data, that $\square_3^{\mathrm{L}}$ is consistently tighter than $\square_3^{\mathrm{T}}$ for $r \leq 0.275$ and significantly tighter for $r \leq 0.035$. Similarly, $\square_4^{\mathrm{H}}$ is tighter than $\square_4^{\mathrm{T}}$ for $r \leq 0.02$. While the convergence orders were confirmed in all our tests, a different relative behaviour between $\square_3^{\mathrm{L}}$ and $\square_3^{\mathrm{T}}$ can be seen in Figure 5. This reflects the fact that the Hausdorff distance d_{H} between the exact and the approximate range depends not only on the convergence *order*, but also on the convergence *constant*, which in turn depends on the test function f. Here, we can deduce from the plots that the convergence constant of $\square_3^{\mathrm{T}}$ for "clover-4" is bigger than that of $\square_3^{\mathrm{L}}$ for "clover-4", but that this relation is inverted for "grass".

5.3 Efficiency and efficacy

In Table 2, we evaluate the five range functions of (15) on the set of 1024 boxes obtained by subdividing the square $B_0 = [-1.2, 1.2]^2$ uniformly into a 32×32 grid. Let $\mathrm{TotalTime}_m^{\mathrm{X}}$ be the sum of the times for evaluating $\square_m^{\mathrm{X}} f(B)$ for each of these 1024 boxes. Similarly, $\mathrm{TotalWidth}_m^{\mathrm{X}}$ is the sum of the widths of the ranges for these 1024 boxes. Viewing the quadratically convergent $\square_2^{\mathrm{T}}$ as the *base line* method, we define the *speedup ratio* (i.e., efficiency ratio) of method $\square_m^{\mathrm{X}}$ as $\frac{\mathrm{TotalTime}_2^{\mathrm{T}}}{\mathrm{TotalTime}_m^{\mathrm{X}}}$. Similarly, the *efficacy ratio* of method $\square_m^{\mathrm{X}}$ is $\frac{\mathrm{TotalWidth}_2^{\mathrm{T}}}{\mathrm{TotalWidth}_m^{\mathrm{X}}}$. Note that we define each ratio so that the method $\square_m^{\mathrm{X}}$ is better than the base line, if and only if the ratio is greater

test function	range function	time (ms)	speedup	efficacy	memory (MB)
clover-4	$□_2^T$ (baseline)	195.98	1	1	141.70
	$□_3^T$	197.92	0.99	1.1978	140.89
	$□_4^T$	233.58	0.84	1.1991	158.68
	$□_3^L$	368.19	0.53	1.1950	251.10
	$□_3^L$ (shared)	179.67	1.09		122.66
	$□_4^H$	546.54	0.36	1.1997	748.78
	$□_4^H$ (shared)	306.17	0.64		584.38
clover-5	$□_2^T$ (baseline)	333.99	1	1	217.26
	$□_3^T$	312.35	1.07	1.2223	216.45
	$□_4^T$	320.70	1.04	1.2229	235.58
	$□_3^L$	592.08	0.56	1.2195	398.05
	$□_3^L$ (shared)	288.54	1.16		193.40
	$□_4^H$	865.50	0.39	1.2240	1039.23
	$□_4^H$ (shared)	457.68	0.73		801.42
clover-8	$□_2^T$ (baseline)	846.72	1	1	560.39
	$□_3^T$	857.81	0.99	1.2986	559.57
	$□_4^T$	848.14	1.00	1.2990	584.61
	$□_3^L$	1514.08	0.56	1.2941	975.60
	$□_3^L$ (shared)	755.13	1.12		466.93
	$□_4^H$	1928.57	0.44	1.3014	2222.16
	$□_4^H$ (shared)	988.37	0.86		1625.63
grass	$□_2^T$ (baseline)	838.99	1	1	492.87
	$□_3^T$	901.25	0.93	1.1993	492.06
	$□_4^T$	804.90	1.04	1.2014	511.20
	$□_3^L$	1385.02	0.61	1.1890	807.83
	$□_3^L$ (shared)	662.46	1.27		381.26
	$□_4^H$	1608.16	0.52	1.2008	1469.37
	$□_4^H$ (shared)	669.50	1.25		916.84

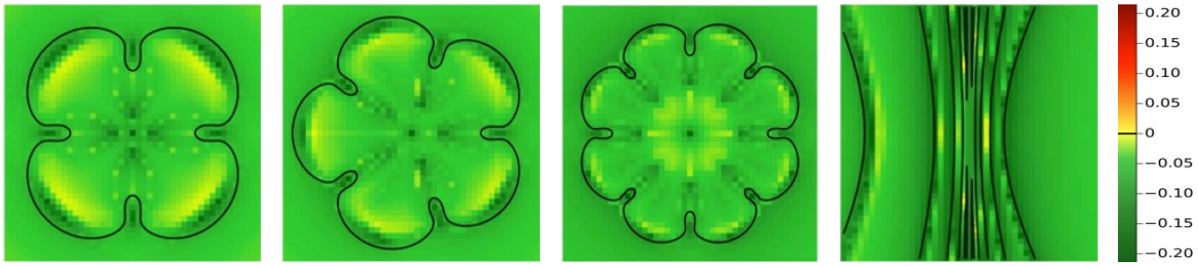
Table 2: Performance comparison of the range functions in (15) for the 1024 boxes obtained by uniformly subdividing the square $[-1.2, 1.2]^2$. For each test function, we highlight in green the best values under *time*, *speedup*, *efficacy*, and *memory* usage.

than 1. The column "time (ms)" gives the average time in milliseconds over 10 runs for each method and test function, as measured by Julia's benchmarking tool.

One of the attractions of Lagrange and Hermite ranges is that in subdivision algorithms, their evaluations can *share* computations [11, 12]. In this case the evaluation of the test function and its partial derivatives at the corners and edge midpoints of the 1024 boxes leads to amortized time and space complexity that is not reflected in the timing of individual evaluations. So we added a second row in Table 2 for these two methods to show the improved speedup and memory usage when sharing is turned on.

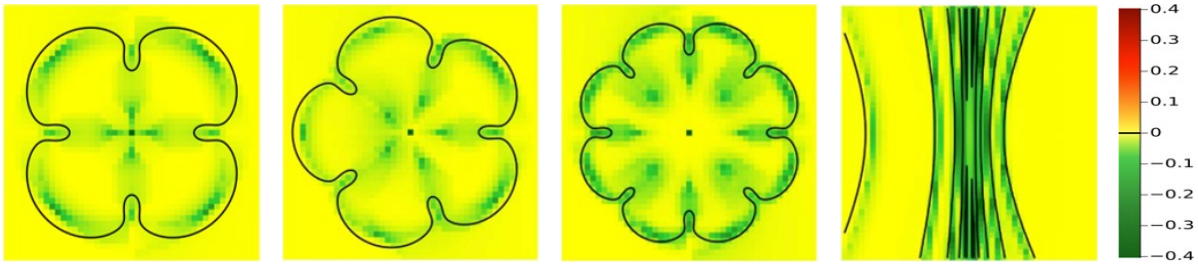
From the data in Table 2 we observe that the efficacy (tightness) is greater than 1 for all the methods with superior convergence order, as expected. In fact, efficacy ratios lie in the range $[1.19, 1.30]$. At least for $□_3^T$ and $□_4^T$, this improvement does not come at a significant loss in efficiency, since the speedups are only marginally less than 1 in some cases and often even greater than 1 (i.e., they are a strict improvement on the base line). Likewise, the memory usage is very similar for all three Taylor forms. Instead, for $□_3^L$ and $□_4^H$ (without sharing), the tradeoff is clearly visible, both in the speedup ratios, which are in the ranges $[0.52, 0.61]$ and $[0.36, 0.52]$, respectively, and the memory usage. But speedup is roughly double (and memory roughly halved), if the data is shared.

We draw two general conclusions (within the limits of our test functions): (1) it appears that the efficacy



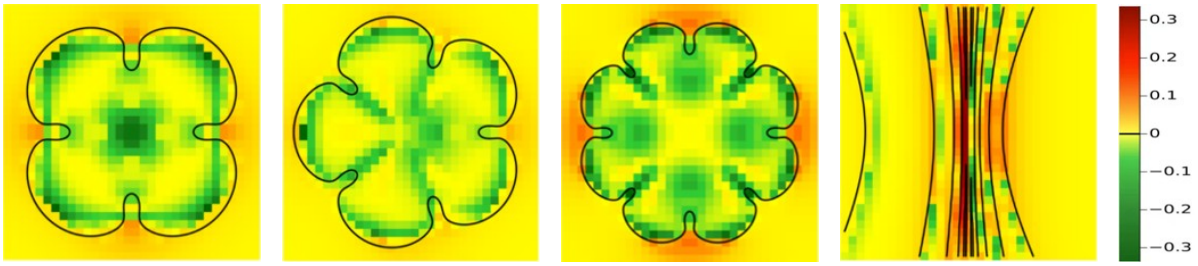
clover-4 clover-5 clover-8 grass

Figure 6: Efficacy comparison between $\square_3^{\mathrm{T}}$ and $\square_2^{\mathrm{T}}$.



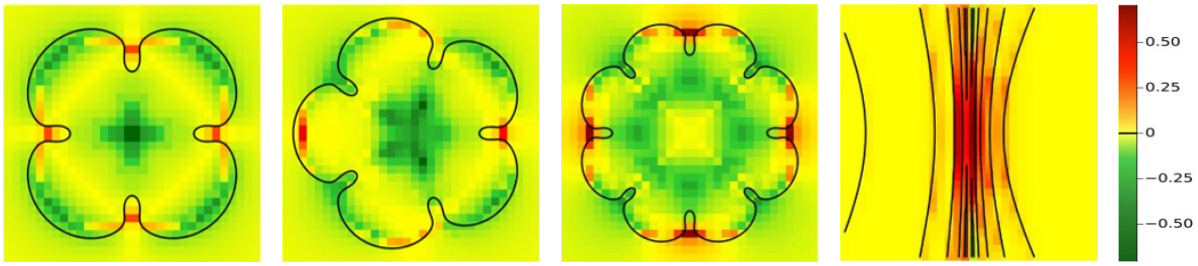
clover-4 clover-5 clover-8 grass

Figure 7: Efficacy comparison between $\square_4^{\mathrm{T}}$ and $\square_3^{\mathrm{T}}$.



clover-4 clover-5 clover-8 grass

Figure 8: Efficacy comparison between $\square_3^{\mathrm{L}}$ and $\square_3^{\mathrm{T}}$.



clover-4 clover-5 clover-8 grass

Figure 9: Efficacy comparison between $\square_4^{\mathrm{H}}$ and $\square_4^{\mathrm{T}}$.

grows with the convergence order m, but the improvement from $m=3$ to $m=4$ is not significant; (2) the efficiency (both in terms of runtime and memory usage) of the higher order Taylor forms $\square_3^{\mathrm{T}}$ and $\square_4^{\mathrm{T}}$ is similar to the base line $\square_2^{\mathrm{T}}$, but the Lagrange form $\square_3^{\mathrm{L}}$ with data sharing is the most efficient method.

5.4 Pairwise efficacy comparison over a landscape

While the previous table uses an aggregate $\mathrm{TotalWidth}_m^{\mathrm{X}}$ to measure the efficacy of $\square_m^{\mathrm{X}}$ against the base line, let us now compare the relative efficacy of two range functions $\square_m^{\mathrm{X}} f$ and $\square_n^{\mathrm{Y}} f$ over the entire landscape of

1024 boxes as follows. For each box B in the 32×32 grid of $B_0 = [-1.2, 1.2]^2$, we compute the value

$$W^{\mathrm{X,Y}}_{m,n}(B) := \log_{10} \frac{w\big(\square^{\mathrm{X}}_m f(B)\big)}{w\big(\square^{\mathrm{Y}}_n f(B)\big)}.$$

Thus, the first range $\square^{\mathrm{X}}_m f$ is tighter than the second $\square^{\mathrm{Y}}_n f$, if and only if $W^{\mathrm{X,Y}}_{m,n} < 0$. We can now visualize their relative tightness over the entire landscape of 1024 boxes by mapping the values $W^{\mathrm{X,Y}}_{m,n}$ to a colour scale that ranges from dark green (for the smallest value) to dark red (for the largest value). In general, green hues indicate boxes for which $\square^{\mathrm{X}}_m f$ is tighter, while boxes for which $\square^{\mathrm{Y}}_n f$ is tighter are in red hues, and yellow hues are used for boxes where both ranges are similar. For example, a mostly green landscape as in Figure 6 indicates that the first function is generally tighter than the second.

We consider four such pairwise comparisons of methods (over all four test functions), which confirm the observations from Table 2. Figure 6 shows that $\square^{\mathrm{T}}_3$ gives significantly tighter ranges than the baseline $\square^{\mathrm{T}}_2$, essentially for all 1024 boxes. The absence of red hues in Figure 7 further shows that $\square^{\mathrm{T}}_4$ is generally tighter than $\square^{\mathrm{T}}_3$, but the predominantly yellow landscape also indicates that the average improvement is not very substantial. However, the green hues along the outlines of the clovers and grass suggest that $\square^{\mathrm{T}}_4$ is tighter near the zero set of f and would be more effective in tracing these curves. Figure 8 shows that the two third-order methods $\square^{\mathrm{L}}_3$ and $\square^{\mathrm{T}}_3$ can both outperform each other in terms of tightness and neither can be declared a clear winner in general. The outcome seems to depend a lot on the local shape of the function over each box. As we can see in Figure 9, a similar verdict holds for the two fourth-order methods $\square^{\mathrm{H}}_4$ and $\square^{\mathrm{T}}_4$.

5.5 Experimental limitations

Although Theorem 2 provides a complete CL theory that can be implemented in practical arithmetic models (say, bigFloat libraries), our implementation adopts the "standard expedient" of computing with the IEEE arithmetic model (Julia's Float64). At present, our SLP implementation only supports polynomial functions. Nevertheless, the underlying theory developed in this paper applies to general analytic functions. More precisely, to compute $\square f(B) = \square g(B) + \square R_g(B)$, we compute $\square g(B)$ (and similarly for $\square R_g(B)$) as follows: If $g(B) = [a, b]$ and $g(x, y)$ is a low degree polynomial, then we have exact algebraic expressions for a and b (see the Appendices). Then a valid range function would be $\square g(B) = [\tilde{a} - \varepsilon, \tilde{b} + \varepsilon]$ where $\tilde{a}, \tilde{b}$ are IEEE evaluations of the expressions for a, b and $|a - \tilde{a}|, |b - \tilde{b}| \leq \varepsilon$. Although ε could be determined for any g, we took the shortcut of ignoring this ε-correction. Let us suppose that all these errors are less than some ε_0. Is this amount of error still "correct"? This question can only be answered in specific applications of these range functions.

6 Conclusion

This paper initiates the study of range functions with superior convergence order for bivariate functions. Our experimental study shows the practicality of these functions for cubic and quartic convergence orders. Similar to the univariate setting [11, 12], the sweet spot, that is, the best trade-off between efficiency and efficacy, and ease of implementation, seems to be the cubic convergence order, since fourth-order methods give only marginally tighter ranges, are generally slower, and more complicated to implement (see Appendices B.4 and B.5).

We conclude with some directions for future work:

- Clearly, the ideas presented in this paper extend to higher convergence orders and to multivariate functions. In particular, Lagrange interpolation at the nodes of a regular $m \times m$ grid with $m > 3$ can be used to define recursive Lagrange forms with order m convergence, and similarly for Hermite interpolation of higher order partial derivatives at the corners of the box. However, the practicality of such extensions may not be obvious and ought to be explored.
- This paper reports the raw performance of our range functions on individual boxes. A more holistic evaluation would be to compare them in the context of an application. As in [11, 12], our Lagrange and Hermite forms are expected to outshine Taylor forms in subdivision applications such as curve tracing [18, 13, 14]. We plan to write follow-up papers, in which we will also include details about our SLP implementation. Moreover, we will extend our framework so as to handle more general functions beyond polynomials.

- The concept of levels may also be exploited in applications. For the root isolation application, Hormann, Yap, and Zhang [12] showed empirically that for several classes of polynomials, for a given convergence order m, there is an optimal level n ($m \le n \le d$) with optimal speedup.

References

[1] K. Alladi and V. E. Hoggatt, Jr. On tribonacci numbers and related functions. *The Fibonacci Quaterly*, 15(1):42–45, Feb. 1977.

[2] R. Becker, M. Sagraloff, V. Sharma, and C. Yap. A near-optimal subdivision algorithm for complex root isolation based on Pellet test and Newton iteration. *Journal of Symbolic Computation*, 86:51–96, May–June 2018.

[3] A. Bowyer, J. Berchtold, D. Eisenthal, I. Voiculescu, and K. Wise. Interval methods in geometric modeling. In R. Martin and W. Wang, editors, *Proceedings of Geometric Modeling and Processing 2000*, pages 321–327. IEEE Computer Society, Hong Kong, Apr. 2000.

[4] J.-S. Cheng, J. Wen, and B. Zhang. Certified numerical real root isolation for bivariate nonlinear systems. *Journal of Symbolic Computation*, 114:149–171, Jan.–Feb. 2023.

[5] Range function project in corelib, since 2019. Download of source, documentation and data: `https://cs.nyu.edu/exact/core_pages/svn-core.html`. The username and password are both 'guest'. You can navigate to the range function project (`../corelib2/trunk/progs/eval`) or go directly to `https://subversive.cims.nyu.edu/exact/corelib2/trunk/progs/eval`.

[6] H. Cornelius and R. Lohner. Computing the range of values of real functions with accuracy higher than second order. *Computing*, 33(3–4):331–347, Sept. 1984.

[7] R. Courant and F. John. *Introduction to Calculus and Analysis*, volume II. Springer, New York, 1989. ISBN 978-1-4613-8960-6.

[8] T. Duff. Interval arithmetic recursive subdivision for implicit functions and constructive solid geometry. *ACM SIGGRAPH Computer Graphics*, 26(2):131–138, July 1992.

[9] G. Farin. *Curves and Surfaces for CAGD: A Practical Guide*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann, San Francisco, 5th edition, 2001. ISBN 978-1-55860-737-8.

[10] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Frontiers in Applied Mathematics. SIAM, Philadelphia, 2nd edition, 2008. ISBN 978-0-89871-659-7.

[11] K. Hormann, L. Kania, and C. Yap. Novel range functions via Taylor expansions and recursive Lagrange interpolation with application to real root isolation. In *Proceedings of the 2021 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC'21, pages 193–200, Saint Petersburg, Russia, July 2021. ACM, New York. [PDF]

[12] K. Hormann, C. Yap, and Y. S. Zhang. Range functions of any convergence order and their amortized complexity analysis. In F. Boulier, M. England, I. Kotsireas, T. M. Sadykov, and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 14139 of *Lecture Notes in Computer Science*, pages 162–182. Springer, Cham, 2023. [PDF]

[13] L. Lin and C. Yap. Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach. *Discrete & Computational Geometry*, 45(4):760–795, June 2011.

[14] L. Lin, C. Yap, and J. Yu. Non-local isotopic approximation of nonsingular surfaces. *Computer-Aided Design*, 45(2):451–462, Feb. 2013.

[15] R. Martin, H. Shou, I. Voiculescu, A. Bowyer, and G. Wang. Comparison of interval methods for plotting algebraic curves. *Computer Aided Geometric Design*, 19(7):553–587, 2002.

[16] R. E. Moore. *Methods and Applications of Interval Analysis*. Number 2 in Studies in Applied and Numerical Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1979. ISBN 978-0-89871-161-5.

[17] B. Mößner and U. Reif. Error bounds for polynomial tensor product interpolation. *Computing*, 86(2–3):185–197, Oct. 2009.

[18] S. Plantinga and G. Vegter. Isotopic approximation of implicit curves and surfaces. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP'04, pages 245–254. ACM, New York, 2004.

[19] H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions*. Ellis Horwood Series in Mathematics and its Applications. Ellis Horwood Limited, Chichester, 1984. ISBN 978-0-85312-703-1.

[20] V. Sharma and C. K. Yap. Near optimal tree size bounds on a simple real root isolation algorithm. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, ISSAC'12, pages 319–326, Grenoble, France, July 2012. ACM, New York.

[21] F. Sichetti, E. Puppo, Z. Huang, M. Attene, D. Zorin, and D. Panozzo. MiSo: A DSL for robust and efficient solve and minimize problems. *ACM Transactions on Graphics*, 44(4):Article 120, 18 pages, July 2025.

[22] J. M. Snyder. Interval analysis for computer graphics. *ACM SIGGRAPH Computer Graphics*, 26(2):121–130, July 1992.

A Exact ranges of univariate polynomials with low degree

We recall the following results from [11]. Let $I = [a, b]$ be a real interval with *radius* $r = (b-a)/2$ and *midpoint* $m = (a+b)/2$. Given a polynomial p over I, let

$$\alpha_0 := \min\{p(a), p(b)\}, \qquad \beta_0 := \max\{p(a), p(b)\}$$

be the minimal and the maximal value of p at the endpoints of I.

A.1 Linear polynomials

The exact range of the linear polynomial

$$p(x) = c_0 + c_1(x-m)$$

is $p(I) = [\alpha_0, \beta_0]$, because p is monotonic. The range can also be expressed as

$$p(I) = c_0 + r[-1, 1]c_1.$$

A.2 Quadratic polynomials

To determine the exact range $p(I) = [\alpha, \beta]$ of the quadratic polynomial

$$p(x) = c_0 + c_1(x-m) + c_2(x-m)^2,$$

we first set $\alpha := \alpha_0$, $\beta := \beta_0$. Next, we observe that the extremum of p occurs at

$$x^* = m - \frac{c_1}{2c_2},$$

which is inside I, if and only if $|c_1| < 2|c_2|r$. If $x^* \notin I$, then p is monotonic on I and our initial assignment of α and β gives the correct range. Otherwise, we check the sign of c_2 to see if p has a minimum ($c_2 > 0$) or a maximum ($c_2 < 0$) at x^* and accordingly replace α or β with

$$p(x^*) = c_0 - \frac{c_1^2}{4c_2}.$$

Note that the special case of p being linear (or constant) with $c_2 = 0$ is handled correctly by this procedure.

A.3 Cubic polynomials

To find the exact range $p(I) = [\alpha, \beta]$ of the cubic polynomial

$$p(x) = c_0 + c_1(x-m) + c_2(x-m)^2 + c_3(x-m)^3,$$

we distinguish between the case $c_3 = 0$, in which we use the procedure in Appendix A.2, and the case $c_3 \neq 0$, in which we first set $\alpha := \alpha_0$ and $\beta := \beta_0$. If the discriminant $\Delta = c_2^2 - c_1 c_3$ of the quadratic equation $p'(x) = 0$ is non-positive, then p is monotonic on I and the initial assignment of α and β gives the correct range. Otherwise, p has a local maximum at x^+ and a local minimum at x^-, where

$$x^\pm = m - \frac{c_2 \pm \sqrt{\Delta}}{3c_3}.$$

It remains to check if these stationary points lie inside I and to adjust α and β accordingly.

B Exact ranges of bivariate polynomials with low degree

Let $B = [a_x, b_x] \times [a_y, b_y]$ be a real, rectangular *box* with *midpoint* $(m_x, m_y) = (a_x + b_x, a_y + b_y)/2$ and *radii* $(r_x, r_y) = (b_x - a_x, b_y - a_y)/2$. Given a bivariate polynomial p over B, let

$$\alpha_0 := \min\{p(a_x, a_y), p(a_x, b_y), p(b_x, a_y), p(b_x, b_y)\},$$
$$\beta_0 := \max\{p(a_x, a_y), p(a_x, b_y), p(b_x, a_y), p(b_x, b_y)\}$$

be the minimal and the maximal value of p at the corners of B.

B.1 Linear polynomials

The exact range of the linear polynomial

$$p(x,y) = c_{0,0} + c_{1,0}(x-m_x) + c_{0,1}(y-m_y)$$

is $p(B) = [\alpha_0, \beta_0]$, because p is monotonic in x and in y. The range can also be expressed as

$$p(B) = c_{0,0} + [-1,1](r_x|c_{1,0}| + r_y|c_{0,1}|),$$

and if B is a *square* with radius $r = r_x = r_y$, then this simplifies to

$$p(B) = c_{0,0} + r[-1,1](|c_{1,0}| + |c_{0,1}|).$$

For example, if p is the linear Taylor polynomial T_1 of f about the midpoint $\boldsymbol{m} = (m_x, m_y)$ of B, then the coefficients to be used in the procedure above are

$$c_{0,0} = f(\boldsymbol{m}), \qquad c_{1,0} = f^{(1,0)}(\boldsymbol{m}), \qquad c_{0,1} = f^{(0,1)}(\boldsymbol{m}).$$

B.2 Quadratic polynomials

To determine the exact range $p(B) = [\alpha, \beta]$ of the quadratic polynomial

$$\begin{aligned} p(x,y) = {} & c_{0,0} + c_{1,0}(x-m_x) + c_{0,1}(y-m_y) \\ & + c_{2,0}(x-m_x)^2 + c_{1,1}(x-m_x)(y-m_y) + c_{0,2}(y-m_y)^2, \end{aligned}$$

we initially set $\alpha := \alpha_0$, $\beta := \beta_0$ and then search for extrema on the boundary of B. To this end, we consider the four univariate quadratic polynomials

$$\begin{aligned} p_1(x) &= p(x, a_y), & x &\in I_1 = [a_x, b_x], \\ p_2(x) &= p(x, b_y), & x &\in I_2 = [a_x, b_x], \\ p_3(y) &= p(a_x, y), & y &\in I_3 = [a_y, b_y], \\ p_4(y) &= p(b_x, y), & y &\in I_4 = [a_y, b_y], \end{aligned}$$

analyse them as described in Appendix A.2, and update α and β accordingly.

To further check for extrema inside B, we compute the *discriminant* of p,

$$D_p = \begin{vmatrix} p_{xx} & p_{xy} \\ p_{yx} & p_{yy} \end{vmatrix} = 4c_{2,0}c_{0,2} - c_{1,1}^2.$$

If $D_p < 0$, then the unique stationary point of p is a saddle point, and if $D_p = 0$, then p either has no stationary points or there exists a line of infinitely many stationary points with a common critical value. In both cases we can be sure that p obtains its minimal and maximal value on the boundary of B, and the exact range $p(B)$ is correctly determined by the previous boundary analysis.

If $D_p > 0$, then p has a unique stationary point at

$$(x^*, y^*) = (m_x, m_y) - \frac{(2c_{1,0}c_{0,2} - c_{0,1}c_{1,1}, 2c_{0,1}c_{2,0} - c_{1,0}c_{1,1})}{D_p},$$

which is inside B, if and only if

$$|2c_{1,0}c_{0,2} - c_{0,1}c_{1,1}| < D_p r_x \qquad \text{and} \qquad |2c_{0,1}c_{2,0} - c_{1,0}c_{1,1}| < D_p r_y.$$

In this case, we check the sign of $c_{2,0}$ to see if p has a minimum ($c_{2,0} > 0$) or a maximum ($c_{2,0} < 0$) at (x^*, y^*) and accordingly replace α or β with

$$p(x^*, y^*) = c_{0,0} - \frac{c_{1,0}^2 c_{0,2} - c_{1,0}c_{0,1}c_{1,1} + c_{0,1}^2 c_{2,0}}{D_p}.$$

Note that the special case of p being linear (or constant) with $c_{2,0} = c_{1,1} = c_{0,2} = 0$ is handled correctly by this procedure.

For example, if p is the quadratic Taylor polynomial T_2 of f about the midpoint $\boldsymbol{m} = (m_x, m_y)$ of B, then the coefficients to be used in the procedure above are

$$\begin{aligned} c_{0,0} &= f(\boldsymbol{m}), & c_{1,0} &= f^{(1,0)}(\boldsymbol{m}), & c_{0,1} &= f^{(0,1)}(\boldsymbol{m}), \\ c_{2,0} &= \tfrac{1}{2} f^{(2,0)}(\boldsymbol{m}), & c_{1,1} &= f^{(1,1)}(\boldsymbol{m}), & c_{0,2} &= \tfrac{1}{2} f^{(0,2)}(\boldsymbol{m}). \end{aligned}$$

B.3 Biquadratic polynomials

The exact range of the biquadratic polynomial

$$\begin{aligned} p(x,y) = {} & c_{0,0} + c_{1,0}(x-m_x) + c_{0,1}(y-m_y) \\ & + c_{2,0}(x-m_x)^2 + c_{1,1}(x-m_x)(y-m_y) + c_{0,2}(y-m_y)^2 \\ & + c_{2,1}(x-m_x)^2(y-m_y) + c_{1,2}(x-m_x)(y-m_y)^2 + c_{2,2}(x-m_x)^2(y-m_y)^2 \end{aligned}$$

can only be computed numerically, since determining the stationary points of p requires finding the roots of a polynomial of degree five. Instead, we propose to split p into the quadratic Taylor polynomial q of p about $\boldsymbol{m}$ and the remainder $r = p - q$, that is,

$$r(x,y) = c_{2,1}(x-m_x)^2(y-m_y) + c_{1,2}(x-m_x)(y-m_y)^2 + c_{2,2}(x-m_x)^2(y-m_y)^2,$$

and to approximate the range of p by $q(B) + r(B) \supseteq p(B)$. While the range of q can be computed as described in Appendix B.2, a careful analysis of r (finding the stationary points, checking the discriminant, etc.) reveals that it does not have isolated local extrema. The range of r can hence be found by analyzing r over the boundary of B, as explained in the first step of the procedure in Appendix B.2.

For example, if p is the biquadratic Lagrange interpolant of f at $\{a_x, m_x, b_x\} \times \{a_y, m_y, b_y\}$, then the coefficients of p are

$$c_{0,0} = f_{1,1}, \qquad c_{1,0} = \frac{f_{2,1} - f_{0,1}}{2r_x}, \qquad c_{0,1} = \frac{f_{1,2} - f_{1,0}}{2r_y},$$

$$c_{2,0} = \frac{f_{2,1} - 2f_{1,1} + f_{0,1}}{2r_x^2}, \qquad c_{1,1} = \frac{f_{2,2} - f_{0,2} - f_{2,0} + f_{0,0}}{4r_x r_y}, \qquad c_{0,2} = \frac{f_{1,2} - 2f_{1,1} + f_{1,0}}{2r_y^2},$$

$$c_{2,1} = \frac{f_{2,2} - 2f_{1,2} + f_{0,2} - f_{2,0} + 2f_{1,0} - f_{0,0}}{4r_x^2 r_y}, \qquad c_{1,2} = \frac{f_{2,2} - 2f_{2,1} + f_{2,0} - f_{0,2} + 2f_{0,1} - f_{0,0}}{4r_x r_y^2}$$

$$c_{2,2} = \frac{f_{2,2} - 2f_{1,2} + f_{0,2} - 2f_{2,1} + 4f_{1,1} - 2f_{0,1} + f_{2,0} - 2f_{1,0} + f_{0,0}}{4r_x^2 r_y^2},$$

where

$$f_{i,j} = f(m_x + (i-1)r_x, m_y + (j-1)r_y), \qquad 0 \le i, j \le 2.$$

B.4 Cubic polynomials

To determine the exact range $p(B) = [\alpha, \beta]$ of the cubic polynomial

$$\begin{aligned} p(x,y) = {} & c_{0,0} + c_{1,0}(x-m_x) + c_{0,1}(y-m_y) \\ & + c_{2,0}(x-m_x)^2 + c_{1,1}(x-m_x)(y-m_y) + c_{0,2}(y-m_y)^2 \\ & + c_{3,0}(x-m_x)^3 + c_{2,1}(x-m_x)^2(y-m_y) + c_{1,2}(x-m_x)(y-m_y)^2 + c_{0,3}(y-m_x)^3, \end{aligned}$$

we initially set $\alpha := \alpha_0$, $\beta := \beta_0$ and then search for extrema on the boundary of B, adjusting α and β accordingly, as in Appendix B.2, but taking into account that the four polynomials are cubic and must be analysed with the procedure in Appendix A.3. To further check for extrema inside B, we have to determine the common roots of the first order partial derivatives of p. To this end, we express the latter as polynomials in x,

$$A(x) := p^{(1,0)}(x + m_x, y + m_y) = a_0 x^2 + a_1 x + a_2, \qquad B(x) := p^{(0,1)}(x + m_x, y + m_y) = b_0 x^2 + b_1 x + b_2,$$

with coefficients in y,

$$\begin{aligned} a_0 &:= 3c_{3,0}, & a_1 &:= 2c_{2,0} + 2c_{2,1}y, & a_2 &:= c_{1,0} + c_{1,1}y + c_{1,2}y^2, \\ b_0 &:= c_{2,1}, & b_1 &:= c_{1,1} + 2c_{1,2}y, & b_2 &:= c_{0,1} + 2c_{0,2}y + 3c_{0,3}y^2, \end{aligned}$$

and consider the determinant of the Sylvester matrix,

$$\begin{vmatrix} a_0 & a_1 & a_2 & 0 \\ 0 & a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 & 0 \\ 0 & b_0 & b_1 & b_2 \end{vmatrix} = (a_0 b_2 - a_2 b_0)^2 - (a_0 b_1 - a_1 b_0)(a_1 b_2 - a_2 b_1).$$

The roots of this quartic polynomial (in y) are the y-coordinates of the stationary points of p, and they can be found, for example, using *Ferrari's method*. To find the corresponding x-coordinates, we need to substitute each of the (up to four) roots into the coefficients of $A(x)$ and solve for x. While A certainly vanishes at the resulting (up to eight) pairs of coordinates, we must further check that $B(x)$ also evaluates to zero, which reduces the number of stationary points of p to at most four, as guaranteed by Bézout's theorem. For those stationary points that lie inside B, we finally use the second derivative test to determine if they are local extrema and update α and β accordingly.

For example, if p is the cubic Taylor polynomial T_3 of f about the midpoint $\boldsymbol{m} = (m_x, m_y)$ of B, then the coefficients to be used in the procedure above are

$$\begin{aligned} c_{0,0} &= f(\boldsymbol{m}), & c_{1,0} &= f^{(1,0)}(\boldsymbol{m}), & c_{0,1} &= f^{(0,1)}(\boldsymbol{m}), \\ c_{2,0} &= \tfrac{1}{2} f^{(2,0)}(\boldsymbol{m}), & c_{1,1} &= f^{(1,1)}(\boldsymbol{m}), & c_{0,2} &= \tfrac{1}{2} f^{(0,2)}(\boldsymbol{m}). \\ c_{3,0} &= \tfrac{1}{6} f^{(3,0)}(\boldsymbol{m}), & c_{2,1} &= \tfrac{1}{2} f^{(2,1)}(\boldsymbol{m}), & c_{1,2} &= \tfrac{1}{2} f^{(1,2)}(\boldsymbol{m}), & c_{0,3} &= \tfrac{1}{6} f^{(0,3)}(\boldsymbol{m}). \end{aligned}$$

B.5 Bicubic polynomials

For a bicubic polynomial

$$\begin{aligned} p(x,y) = {} & c_{0,0} + c_{1,0}(x-m_x) + c_{0,1}(y-m_y) \\ & + c_{2,0}(x-m_x)^2 + c_{1,1}(x-m_x)(y-m_y) + c_{0,2}(y-m_y)^2 \\ & + c_{3,0}(x-m_x)^3 + c_{2,1}(x-m_x)^2(y-m_y) + c_{1,2}(x-m_x)(y-m_y)^2 + c_{0,3}(y-m_x)^3 \\ & + c_{3,1}(x-m_x)^3(y-m_y) + c_{2,2}(x-m_x)^2(y-m_y)^2 + c_{1,3}(x-m_x)(y-m_x)^3 \\ & + c_{3,2}(x-m_x)^3(y-m_y)^2 + c_{2,3}(x-m_x)^2(y-m_y)^3 + c_{3,3}(x-m_x)^3(y-m_y)^3, \end{aligned}$$

we proceed as for biquadratic polynomials in Appendix B.3. We split p into the cubic Taylor polynomial q of p about $\boldsymbol{m}$ and the remainder $r = p - q$, that is,

$$\begin{aligned} r(x,y) = {} & c_{3,1}(x-m_x)^3(y-m_y) + c_{2,2}(x-m_x)^2(y-m_y)^2 + c_{1,3}(x-m_x)(y-m_x)^3 \\ & + c_{3,2}(x-m_x)^3(y-m_y)^2 + c_{2,3}(x-m_x)^2(y-m_y)^3 + c_{3,3}(x-m_x)^3(y-m_y)^3, \end{aligned}$$

and approximate the range of p by $q(B) + r(B) \supseteq p(B)$. While the range of q can be computed as described in Appendix B.4, a slight variation of this procedure can be used to determine the range of r. The restriction of r to the boundary of B gives four cubic polynomials, whose extrema are determined in the same way. To find the extrema of r inside B, we observe that the first order partial derivatives of r are

$$r^{(1,0)}(x+m_x, y+m_y) = y\,A(x), \qquad r^{(0,1)}(x+m_x, y+m_y) = x\,B(x),$$

where

$$A(x) := a_0 x^2 + a_1 x + a_2, \qquad B(x) := b_0 x^2 + b_1 x + b_2$$

are polynomials in x with coefficients in y,

$$\begin{aligned} a_0 &:= 3c_{3,1} + 3c_{3,2} y + 3c_{3,3} y^2, & a_1 &:= 2c_{2,2} y + 2c_{2,3} y^2, & a_2 &:= c_{1,3} y^2, \\ b_0 &:= c_{3,1} + 2c_{3,2} y + 3c_{3,3} y^2, & b_1 &:= 2c_{2,2} y + 3c_{2,3} y^2, & b_2 &:= 3c_{1,3} y^2. \end{aligned}$$

The determinant of the corresponding Sylvester matrix is $y^4 D(y)$, where D is a quartic polynomial, leading to at most four stationary points of r, which need to be further checked. The additional stationary point of r at (m_x, m_y) can be ignored, because $r(m_x, m_y) = 0$, and this value is already contained in the ranges of the boundary polynomials, because $r(x, m_y) = r(m_x, y) = 0$ for any $x, y \in \mathbb{R}$.

For example, if p is the bicubic Hermite interpolant of f at the corners of B, then the coefficients of p are

$$
\begin{aligned}
c_{0,0} &= \frac{+f_{0,0}+f_{0,1}+f_{1,0}+f_{1,1}}{4} + \frac{+f^x_{0,0}+f^x_{0,1}-f^x_{1,0}-f^x_{1,1}}{8r_x^{-1}} + \frac{+f^y_{0,0}-f^y_{0,1}+f^y_{1,0}-f^y_{1,1}}{8r_y^{-1}} + \frac{+f^{xy}_{0,0}-f^{xy}_{0,1}-f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_x^{-1}r_y^{-1}},\\
c_{1,0} &= \frac{-f_{0,0}-f_{0,1}+f_{1,0}+f_{1,1}}{8r_x/3} + \frac{-f^x_{0,0}-f^x_{0,1}-f^x_{1,0}-f^x_{1,1}}{8} + \frac{-f^y_{0,0}+f^y_{0,1}+f^y_{1,0}-f^y_{1,1}}{16r_x r_y^{-1}/3} + \frac{-f^{xy}_{0,0}+f^{xy}_{0,1}-f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_y^{-1}},\\
c_{0,1} &= \frac{-f_{0,0}+f_{0,1}-f_{1,0}+f_{1,1}}{8r_y/3} + \frac{-f^x_{0,0}+f^x_{0,1}+f^x_{1,0}-f^x_{1,1}}{16r_y r_x^{-1}/3} + \frac{-f^y_{0,0}-f^y_{0,1}-f^y_{1,0}-f^y_{1,1}}{8} + \frac{-f^{xy}_{0,0}-f^{xy}_{0,1}+f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_x^{-1}},\\
c_{2,0} &= \frac{-f^x_{0,0}-f^x_{0,1}+f^x_{1,0}+f^x_{1,1}}{8r_x} + \frac{-f^{xy}_{0,0}+f^{xy}_{0,1}+f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_x r_y^{-1}},\\
c_{1,1} &= \frac{+f_{0,0}-f_{0,1}-f_{1,0}+f_{1,1}}{16r_x r_y/9} + \frac{+f^x_{0,0}-f^x_{0,1}+f^x_{1,0}-f^x_{1,1}}{16r_y/3} + \frac{+f^y_{0,0}+f^y_{0,1}-f^y_{1,0}-f^y_{1,1}}{16r_x/3} + \frac{+f^{xy}_{0,0}+f^{xy}_{0,1}+f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_x^{-1}},\\
c_{0,2} &= \frac{-f^y_{0,0}+f^y_{0,1}-f^y_{1,0}+f^y_{1,1}}{8r_y} + \frac{-f^{xy}_{0,0}+f^{xy}_{0,1}+f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_y r_x^{-1}},\\
c_{3,0} &= \frac{+f_{0,0}+f_{0,1}-f_{1,0}-f_{1,1}}{8r_x^3} + \frac{+f^x_{0,0}+f^x_{0,1}+f^x_{1,0}+f^x_{1,1}}{8r_x^2} + \frac{+f^y_{0,0}-f^y_{0,1}-f^y_{1,0}+f^y_{1,1}}{16r_x^3 r_y^{-1}} + \frac{+f^{xy}_{0,0}-f^{xy}_{0,1}+f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_x^2 r_y^{-1}},\\
c_{2,1} &= \frac{+f^x_{0,0}-f^x_{0,1}-f^x_{1,0}+f^x_{1,1}}{16r_x r_y/3} + \frac{+f^{xy}_{0,0}+f^{xy}_{0,1}-f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_x},\\
c_{1,2} &= \frac{+f^y_{0,0}-f^y_{0,1}-f^y_{1,0}+f^y_{1,1}}{16r_x r_y/3} + \frac{+f^{xy}_{0,0}-f^{xy}_{0,1}+f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_y},\\
c_{0,3} &= \frac{+f_{0,0}-f_{0,1}+f_{1,0}-f_{1,1}}{8r_y^3} + \frac{+f^x_{0,0}-f^x_{0,1}-f^x_{1,0}+f^x_{1,1}}{16r_y^3 r_x^{-1}} + \frac{+f^y_{0,0}+f^y_{0,1}+f^y_{1,0}+f^y_{1,1}}{8r_y^2} + \frac{+f^{xy}_{0,0}+f^{xy}_{0,1}-f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_y^2 r_x^{-1}},\\
c_{3,1} &= \frac{-f_{0,0}+f_{0,1}+f_{1,0}-f_{1,1}}{16r_x^3 r_y/3} + \frac{-f^x_{0,0}+f^x_{0,1}-f^x_{1,0}+f^x_{1,1}}{16r_x^2 r_y/3} + \frac{-f^y_{0,0}-f^y_{0,1}+f^y_{1,0}+f^y_{1,1}}{16r_x^3} + \frac{-f^{xy}_{0,0}-f^{xy}_{0,1}-f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_x^2},\\
c_{2,2} &= \frac{+f^{xy}_{0,0}-f^{xy}_{0,1}-f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_x r_y},\\
c_{1,3} &= \frac{-f_{0,0}+f_{0,1}+f_{1,0}-f_{1,1}}{16r_x r_y^3/3} + \frac{-f^x_{0,0}+f^x_{0,1}-f^x_{1,0}+f^x_{1,1}}{16r_y^3} + \frac{-f^y_{0,0}-f^y_{0,1}+f^y_{1,0}+f^y_{1,1}}{16r_x r_y^2/3} + \frac{-f^{xy}_{0,0}-f^{xy}_{0,1}-f^{xy}_{1,0}-f^{xy}_{1,1}}{16r_y^2},\\
c_{3,2} &= \frac{-f^y_{0,0}+f^y_{0,1}+f^y_{1,0}-f^y_{1,1}}{16r_x^3 r_y} + \frac{-f^{xy}_{0,0}+f^{xy}_{0,1}-f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_x^2 r_y},\\
c_{2,3} &= \frac{-f^x_{0,0}+f^x_{0,1}+f^x_{1,0}-f^x_{1,1}}{16r_x r_y^3} + \frac{-f^{xy}_{0,0}-f^{xy}_{0,1}+f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_x r_y^2},\\
c_{3,3} &= \frac{+f_{0,0}-f_{0,1}-f_{1,0}+f_{1,1}}{16r_x^3 r_y^3} + \frac{+f^x_{0,0}-f^x_{0,1}+f^x_{1,0}-f^x_{1,1}}{16r_x^2 r_y^3} + \frac{+f^y_{0,0}+f^y_{0,1}-f^y_{1,0}-f^y_{1,1}}{16r_x^3 r_y^2} + \frac{+f^{xy}_{0,0}+f^{xy}_{0,1}+f^{xy}_{1,0}+f^{xy}_{1,1}}{16r_x^2 r_y^2},
\end{aligned}
$$

where

$$
\begin{aligned}
f_{i,j} &= f(m_x+(2i-1)r_x, m_y+(2j-1)r_y),\\
f^x_{i,j} &= f^{(1,0)}(m_x+(2i-1)r_x, m_y+(2j-1)r_y),\\
f^y_{i,j} &= f^{(0,1)}(m_x+(2i-1)r_x, m_y+(2j-1)r_y),\\
f^{xy}_{i,j} &= f^{(1,1)}(m_x+(2i-1)r_x, m_y+(2j-1)r_y),
\end{aligned}
$$

for $0 \le i, j \le 1$.