Novel Range Functions via Taylor Expansions and Recursive Lagrange Interpolation with Application to Real Root Isolation

Kai Hormann
kai.hormann@usi.ch
Università della Svizzera italiana
Lugano, Switzerland

Lucas Kania
lucas.kania@usi.ch
Università della Svizzera italiana
Lugano, Switzerland

Chee Yap
yap@cs.nyu.edu
Courant Institute, NYU
New York, USA

ABSTRACT
Range functions are an important tool for interval computations, and they can be employed for the problem of root isolation. In this paper, we first introduce two new classes of range functions for real functions. They are based on the remainder form by Cornelius and Lohner [7] and provide different improvements for the remainder part of this form. On the one hand, we use centered Taylor expansions to derive a generalization of the classical Taylor form with higher than quadratic convergence. On the other hand, we propose a recursive interpolation procedure, in particular based on quadratic Lagrange interpo- lation, leading to recursive Lagrange forms with cubic and quartic convergence. We then use these forms for isolating the real roots of square-free polynomials with the algorithm Eval, a relatively recent algorithm that has been shown to be effective and practical. Finally, we compare the performance of our new range functions against the standard Taylor form. Range functions are often compared in isolation; in contrast, our holistic comparison is based on their performance in an application. Specifically, Eval can exploit features of our recursive Lagrange forms which are not found in range functions based on Taylor expansion. Experimentally, this yields at least a twofold speedup in Eval.

1 INTRODUCTION
This paper addresses two related computational problems: (P1) range functions and (P2) root isolation. Computing the range of functions is arguably the most basic task in interval computation [11, 19, 23]. Root isolation is also a fundamental task in the huge classical literature on root finding [17]. These two problems are connected by the fact that root isolation can be reduced to evaluating range functions. To see this, the next two subsections review the relevant literature on range functions and root isolation.

1.1 Range functions
We first consider problem (P1). Let \( f : \mathbb{R} \to \mathbb{R} \) be a real function. For any \( S \subseteq \mathbb{R} \), the range of \( f \) on \( S \) is the set \( f(S) := \{ f(x) : x \in S \} \) and we define the magnitude of \( S \) as \( |S| := \sup\{|s| : s \in S\} \). Let \( f_I \) denote the set of closed bounded intervals. For any \( I \in f_I \) with \( I = [a, b] \), the width, radius, and midpoint of \( I \) are given by \( \omega(I) := b - a \), \( r(I) := (b - a)/2 \), and \( m(I) := (a + b)/2 \), respectively. Note that \( |I| = \max\{|a|, |b|\} \). A range function (or inclusion function) for \( f \) is a function of the form

\[
\hat{f} : f_I \to f_I,
\]

where \( f(I) \subseteq \hat{f}(I) \) for all \( I \in f_I \). If \( \hat{f}(I) = f(I) \) for all \( I \), we call it the exact range function. Note that \( \hat{f} f \) is a generic name for a range function of \( f \); we use subscripts and/or superscripts to identify particular range functions: e.g., \( \hat{f}_0 f, \hat{f}_1 f \), or \( \hat{f}_2 f \). We can compare range functions using a natural “tightness partial order” on range functions of \( f \); we say that \( \hat{f}_0 f \) is as tight as \( \hat{f}_1 f \), denoted \( \hat{f}_0 f \leq \hat{f}_1 f \), if \( \hat{f}_1 f(I) \subseteq \hat{f}_0 f(I) \) for all \( I \). Generally, we prefer range functions that are as tight as possible, ideally the exact range function. But since tight range functions are inefficient (i.e., expensive to compute), we must choose a trade-off between tightness and efficiency. Comparative studies of range functions based on tightness or efficiency are often done in isolation, independent of any application. For example, see [8, 9, 31]. In this paper, we give a holistic or integrated comparison of range functions, namely comparisons in the context of an application (see Sec. 5).

A more robust way to evaluate range functions is to look at “asymptotic tightness”. We say that \( \hat{f} f \) has order \( k \) convergence (for \( k \geq 1 \)) on \( I_0 \) if there exists a constant \( C_0 \) > 0 that depends on \( f \) and \( I_0 \) but not on \( I \), such that

\[
q(f(I), \hat{f}(I)) \leq C_0 \omega(I)^k
\]

for all \( I \subseteq I_0 \), where \( q([a, b], [a', b']) := \max\{|a - a'|, |b - b'|\} \) is the Hausdorff distance on intervals. If \( \hat{f} f \) has at least order 1 convergence, then we call \( \hat{f} f \) convergent. Note that for any sequence \( (I_i)_{i \geq 1} \) of intervals that converges monotonically to a point \( p \in I_0 \),
a convergent range function satisfies
\[ f(p) = \lim_{i \to \infty} \mathbf{f}(l_i). \]
Such a convergent range function is also called a box form of \( f \) [31].

When \( k = 2 \), we say \( \mathbf{f} \) has quadratic convergence.

Cornelius and Lohner [7] were the first to introduce techniques for higher than quadratic convergence. For any function \( g: \mathbb{R} \to \mathbb{R} \), they consider range functions of \( f \) of the form
\[ \mathbf{g}_f(l) := g(l) + \mathbf{R}_g(l), \]
where \( \mathbf{R}_g \) is the remainder function. They call \( g \) the exact part of this range function because its range must be computed exactly. This limits \( g \) to polynomials of small degree \( d \) (Cornelius and Lohner suggest \( d \leq 5 \)). The remainder part \( \mathbf{R}_g(l) \) need not be exact, but its width controls the overall Hausdorff distance, since [7, Theorem 4]
\[ q(f(l), \mathbf{g}_f(l)) \leq w(\mathbf{R}_g(l)). \]
It follows that the remainder form \( \mathbf{g}_f(l) \) has order \( k \) convergence, if \( w(\mathbf{R}_g(l)) \leq C_w(\mathbf{g})^k \).

Cornelius and Lohner show that this can be achieved by letting the exact part \( g \) be a Hermite interpolant of \( f \). In fact, if \( f \) is \( k \) times continuously differentiable, \( x_0, \ldots, x_l \in I \) are distinct interpolation nodes, \( p_0, \ldots, p_l \) are positive integers with \( \sum_{i=0}^l p_i = k \), and \( g \) is the unique polynomial of degree at most \( k - 1 \), such that
\[ g^{(j)}(x_i) = f^{(j)}(x_i), \quad j = 0, \ldots, p_i - 1, \quad i = 0, \ldots, l, \]
then the remainder function can be expressed for any \( x \in I \) as
\[ \mathbf{R}_g(x) = \frac{1}{k!} f^{(k)}(\xi_k) (x - x_i)^{p_i}, \]
for some \( \xi_k \in I \). We now define the remainder part as
\[ \mathbf{R}_g(l) := \frac{1}{k!} f^{(k)}(\xi) \sum_{i=0}^l (l - x_i)^{p_i}, \]
where \( f^{(k)}(\xi) \) is what Ratschek and Rokne [23, p. 23] call the natural interval extension of \( f^{(k)}(x) \). For example, if \( f(x) \) is a polynomial, we write it as an expression \( E(x) \) in the nested Horner form and define \( \mathbf{f}^{(k)}(l) := E(l) \). The remainder form \( \mathbf{g}_f(l) \) in (1) then has order \( k \) convergence, because \( |l - x_i| \leq w(l) \) and Lemma 1.6 in [23, p. 24] imply
\[ w(\mathbf{R}_g(l)) \leq 2w(\mathbf{R}_f(l)) \leq 2 \frac{|f^{(k)}(\xi)|}{k!} w(l)^k \leq \frac{2}{k!} \frac{|f^{(k)}(l_0)|}{k!} w(l)^k. \]

The simplest example of this approach is the convergent mean value form around \( x_0 \),
\[ \mathbf{f}_M(l) := f(x_0) + \mathbf{f}^*(l)(l - x_0), \]
which is obtained by letting \( f = 0 \) and \( p_0 = k = 1 \), so that \( g \) is the constant interpolant of \( f \) at \( x_0 \). This form has even quadratic convergence, if the range \( f^*(l) \) is approximated with a Lipschitz range function [23].

Cornelius and Lohner further point out that it is also possible to define the exact part as
\[ \hat{g}(x) := g(x) + \frac{1}{k!} \sum_{i=0}^l (x - x_i)^{p_i}. \]
for some \( y \in f^{(k)}(l) \subset \mathbf{f}^{(k)}(l) \) and the remainder part (cf. (4)) as
\[ \mathbf{R}_g(l) := \frac{1}{k!} \sum_{i=0}^l (f^{(k)}(l) - y)(l - x_i)^{p_i}. \]

If \( f^{(k)} \) is Lipschitz continuous, then this gives one extra order of convergence, because \( |f^{(k)}(l) - y| \leq w(\mathbf{f}^{(k)}(l)) \leq C w(l) \) for all \( l \subset I_0 \) and some constant \( C_w > 0 \) that depends on \( I_0 \) but not on \( f \) [7, Theorem 2]. In this variant, \( \hat{g} \) is of degree \( k \) and the condition that distinguishes \( \hat{g} \) from \( g \) is that \( g^{(k)} = 0 \). Evaluating \( \hat{g}(l) \) exactly is of course more costly than evaluating \( g^{(k)}(l) \), because \( \hat{g} \) has a higher degree than \( g \). Note that we can also get this extra order of convergence by adding one Hermite interpolation condition to the definition of \( g \). The evaluation of the exact part would then be as costly as the evaluation of \( \hat{g}(l) \), but the remainder part would depend on \( f^{(k+1)} \), while the remainder part in (6) depends on \( f^{(k)} \), a fact that we shall exploit in Sec. 3.2.

The Cornelius–Lohner framework appears to suggest that convergence is limited by the exact part alone, without attaching much interest to the remainder part. In this paper, we suggest the contrary: for any function \( f \) with exact part \( g \), the remainder part \( \mathbf{R}_g(l) \) in (1) can vary. Despite having the same order of convergence, their actual performance in an application like root isolation can diverge significantly.

In this paper, we propose two new ideas for defining such improved remainder parts. The first relies on expressing the remainder function (3) in centered form (Sec. 2.1), the second approximates \( f^{(k)}(l) \) in (4) using again the remainder form in (1), thus applying the idea of Cornelius and Lohner recursively (Sec. 3).

### 1.2 Real root isolation and Eval

We next turn to (P2). Consider again a real function \( f: \mathbb{R} \to \mathbb{R} \). The zero set of \( f \) on \( S \subset \mathbb{R} \) is \( \text{Zero}_f(S) := \{ x \in S : f(x) = 0 \} \), and \( \#(S) \) denotes\(^1\) the cardinality of \( \text{Zero}_f(S) \). An isolator for \( f \) is an interval \( I \) such that \( f(I) = 0 \), and we say that \( I \) isolates the unique zero of \( f \) in \( I \). The root isolation problem can then be formalized as follows:

Given \( f \) and an interval \( I_0 \subset \mathbb{R} \), compute a set \( Z \) of isolators for \( f \), such that each \( \zeta \in \text{Zero}_f(I_0) \) is isolated by some \( I \in Z \). Assuming \( f \) to be nice, in the sense that \( f \) is continuously differentiable and the zeros of \( f \) in \( I_0 \) are simple (i.e., \( f'(\zeta) \neq 0 \)), we can reduce problem (P2) to (P1) using a procedure that we call Eval (see Algo. 1).

Note that the numerical computation of Eval is reduced to evaluating two range functions, one for \( f \) (line 5) and one for its derivative \( f' \) (line 6). Moreover, Eval uses two queues to hold intervals, an active queue \( Q \) and an output queue \( Z \). The intervals \( I \) are bisected until either \( \emptyset \not\subset \mathbf{f}(l) \) or \( 0 \not\subset \mathbf{f}'(l) \) holds. We may call these two conditions the exclusion and inclusion predicates.

Eval terminates and solves problem (P2), if we assume the two range functions \( \mathbf{f} \) and \( \mathbf{f}' \) to be convergent on \( I_0 \). It is then clear that each \( l \in Z \) represents a unique root \( \zeta \in \text{Zero}_f(I_0) \), because \( I_0 \) is added to \( Z \) if and only if \( f(a) f(b) \leq 0 \) (line 9), which guarantees the existence of a root by the intermediate value theorem, and if \( f \) is

\(^1\)Note that root multiplicity is not used in the definitions of \( \text{Zero}_f(S) \) and \( \#(S) \). In particular, \( \text{Zero}_f(S) \) is a set, not a multiset.
strictly monotonic on \( I = [a, b] \) (line 8), which assures the uniqueness of that root. Moreover, each \( \zeta \in \text{Zero}_f(I_0) \) is represented by at most two isolators. In case two isolators \( I, J \in Z \) represent \( \zeta \), then \( \zeta \in I \cap J \) is a common endpoint of \( I \) and \( J \). Such duplication is easily detected and removed, or avoided upfront. For example, if \( f \) is a polynomial with rational coefficients and rational arithmetic is used in \( \text{Eval} \), then we can replace the weak inequality in line 9 by the strict inequality \( f(a)f(b) < 0 \) and instead test \( f(m) = 0 \) after line 7, adding the point interval \([m, m]\) to \( Z \) if the test holds.

Despite its simplicity, the subdivision tree of \( \text{Eval} \) is “near-optimal” when \( f \) is an integer polynomial [3, 4, 28] and the box forms \( \mathbf{O} \) and \( \mathbf{O}'' \) are the “maximal” centered Taylor forms \( \mathbf{O}'' \) (see Sec. 2). In other words, it asymptotically matches the tree size of “continuous amortization” [3, 4, 28].

1.3 Some broader literature

Besides the book of Ratschek and Rokne [23] on range functions, we refer to Neumayer [20, Chapter 2.4] and Stahl’s thesis [29] for further investigations of the remainder forms of Cornelius and Lohner [7], which are also referred to as interpolations forms.

To our knowledge, the first version of \( \text{Eval} \) is from Mitchell [18] in the context of ray tracing in computer graphics. Its current formulation as a root isolation algorithm, together with complexity analysis, began with [5]. Yap et al. introduced \( \text{Eval} \) as a 1-dimensional analogue of the 2-dimensional algorithm of Plantinga and Vegter for isotopic approximation of non-singular curves [14, 15, 22]. Besides \( \text{Eval} \), Yap et al. also introduced \( \text{CEval} \) [25] for complex roots, and \( \text{AEval} \) [32] for analytic roots. The complexity analysis of these algorithms can be captured under the elegant framework of “continuous amortization” [3, 4, 28].

Root finding for polynomials is a highly classical problem [16, 17] that has remained active to the present. The modern complexity-theoretic approach to root finding was initiated by Schönhage in 1982 [26]. A basic quest is to construct “near-optimal” algorithms, and in the last decade, significant progress has been made in this direction; see Sagraloff and Mehlhorn [24] (for real roots) and Becker et al. [1, 2] (for complex roots). The new near-optimal algorithms (like \( \text{Eval} \)) are based on the subdivision paradigm; moreover, they were implemented soon after their appearance [12, 13]. In contrast, the original near-optimal algorithm [21] has never been implemented (see [21, p. 703] for some challenges).

1.4 Overview of the paper

In Section 2, we introduce a family of range functions based on Taylor expansions. Technically, these functions are not new, but within the Cornelius–Lohner framework, we highlight their true role as improvements on the remainder parts. In Section 3, we introduce range functions based on recursive Lagrange interpolation. These are new but again, we can view them as improvements of the remainder parts. In Secs. 4 and 5, we evaluate the deployment of eight of these range functions in the \( \text{Eval} \) algorithm; here, the Lagrange form begins to shine because of its “distributed evaluation” scheme (see Sec. 4.1). We conclude in Sec. 6.

2 NEW RANGE FUNCTIONS BASED ON CENTERED TAYLOR EXPANSIONS

A classic approach for designing a remainder form (1) with quadratic convergence is to choose \( t = 0 \) and \( p_0 = k = 2 \) in (2) and letting \( x_0 = m := m(1) \), so that the exact part is the linear Taylor polynomial of \( f \) about the midpoint of \( I \), that is, \( g_I(x) := f(m) + (x - m) f'(m) \). This gives the centered form \( \Box_g f(I) := g_I(I) + \Box_{R_2}(I) \). One option now is to follow Cornelius and Lohner and express the remainder part as in (4),

\[
\Box R_{g_1}(I) = \frac{1}{2} f''(m)^2, \tag{7}
\]

where \( f''(m) \) is the natural interval extension of \( f''(x) \). We call the resulting version of \( \Box g \) the minimal (centered) Taylor form. This can be improved considerably, if \( f \) is \( n \) times continuously differentiable for \( n > 2 \), by using the \((n - 1)\)-th order Taylor expansion of \( f \) about \( m \) to write the remainder function as

\[
R_g(x) = \sum_{i=2}^{n-1} \frac{f^{(i)}(m)}{i!} (x - m)^i + \frac{f^{(n)}(\xi)}{n!} (x - m)^n, \tag{8}
\]

for some \( \xi \in I \). We now define

\[
c_i := \frac{f^{(i)}(m)}{i!}, \quad i = 0, \ldots, n - 1, \quad c_n := \frac{|f^{(n)}(I)|}{n!}, \tag{9}
\]

where the magnitude of the natural interval extension, \( |f^{(n)}(I)| \), can be replaced by \( f^{(n)}(m) \) in the definition of \( c_n \), if \( f^{(n)} \) is a constant, for example, in the case of \( f \) being a polynomial of degree \( d \leq n \). We then get the following improvement of (7):

\[
\Box R_{g_1}(I) := \sum_{i=2}^{n} c_i (I - m)^i = r^2[-1,1] S_{2,n}, \quad S_{2,n} := \sum_{i=2}^{n} |c_i| r^{i-2}, \tag{10}
\]

where \( r := r(I) \). Computing the \( c_i \)’s takes \( O(n \log n) \) arithmetic steps (or \( O(n^2) \) in simple implementations, as in Sec. 5); for bit-complexity, see [30]. In contrast, the natural interval extension (7) requires \( O(n) \) steps. What do we get in return? Although this does not change the quadratic convergence of the centered form \( \Box_g f(I) \), it may be much better than the remainder part in (7) of Cornelius and Lohner, because successive terms of \( S_{2,n} \) converge with higher and higher order. This is dramatically illustrated below in Tables 2–4 (columns \( E_2^2 \) and \( E_2^3 \)). Recalling that the exact range of \( g_1 \) is
\( g_1(l) = c_0 + r[-1, 1]c_1 \) (see App. A.1), we realize that the resulting centered form
\[
\Omega_{k,m}f(l) := c_0 + r[-1, 1]c_1 + r^2[-1, 1]S_{k,n} \tag{11}
\]
is actually just the classical Taylor form of order \( n \) (or “level \( n \)”) using our terminology below) \cite{23, p. 77}, with the range \( f^{(n)}(l) \) approximated by \(|\Omega f^{(n)}(l)| \cdot [-1, 1] \).

2.1 Taylor forms with order \( k \) convergence
Following Cornelius and Lohner, we can raise the convergence order from quadratic to basically any order \( k > 2 \), simply by replacing \( g_1 \) with the \((k - 1)\)-th order Taylor polynomial of \( f \) about \( m \),
\[
g_{k-1}(x) := \sum_{i=0}^{k-1} \frac{f(i)(m)}{i!}(x-m)^i = \sum_{i=0}^{k-1} c_i(x-m)^i, \tag{12}
\]
But instead of expressing the remainder function \( R_{g_k} = f - g_k \) in terms of the \( k \)-th derivative of \( f \) as (cf. (3))
\[
R_{g_{k-1}}(x) = \frac{1}{k!} f^{(k)}(\xi_x)(x-m)^k, \tag{13}
\]
we continue the Taylor expansion of \( f^{(k)}(x) \) all the way to \( n = 1 \) for some \( n \geq k \) (assuming that the derivatives exist), to obtain (cf. (8))
\[
R_{g_{k-1}}(x) = \sum_{i=k}^{n} \frac{f(i)(m)}{i!}(x-m)^i + \frac{f(n)(\xi_x)}{n!}(x-m)^n, \tag{14}
\]
for some \( \xi_x \in I \). As above (cf. (10) and (11)), we then get the generalized Taylor form of (convergence) order \( k \) and level \( n \):
\[
\Omega_{k,m}^{T} f(l) := g_{k-1}(l) + r^k[-1, 1]S_{k,n}, \quad S_{k,n} := \sum_{i=k}^{n} c_i r^{i-k}, \tag{15}
\]
where the \( c_i \) are defined as in (9). The level \( n \) is minimal if \( n = k \), and maximal if \( n = \infty \). The maximal level is only possible when \( f \) is analytic and \( r \) sufficiently small, so that \( S_{k,\infty} \) is convergent. Clearly, if \( f \) is a polynomial of degree \( d \), then \( S_{k,\infty} \) is a finite sum and convergent for any \( r \). We call the corresponding range functions minimal and maximal Taylor forms of order \( k \), denoted by
\[
\Omega_{k,m}^{T} f(l) \text{ and } \Omega_{k,m} f(l), \text{ respectively. This definition includes the minimal Taylor form based on } g_1 \text{ (cf. (7)) as a special case for } k = n = 2.
\]

For \( k = 3 \), computing the exact range of the quadratic Taylor polynomial \( g_2 \) is only marginally more costly (see App. A.2) than computing \( g_1(l) \) and the cubic convergence gives a noticeable performance gain when used in \textsc{Eval} (see Sec. 5). But already for \( k = 4 \) the computational overhead of determining the range \( g_3(l) \) exactly (see App. A.3) appears to outweigh the benefit of the better convergence order, at least in the context of \textsc{Eval}, leaving only a slight advantage in terms of running time. Note that there is a similar phenomenon in Newton’s method where quadratic convergence is the sweet spot despite the possibility of achieving cubic (Halley’s method) or higher convergence.

3 NEW RANGE FUNCTIONS BASED ON RECURSIVE INTERPOLATION

Another approach to improving the remainder part is by recursively applying the idea of Cornelius and Lohner. To this end, let \( h_0 \) be the Hermite interpolant of \( f \) for a certain choice of interpolation nodes \( x_i \) and multiplicities \( p_i \) and with degree at most \( k - 1 \). According to (3), the remainder part \( R_{h_0} = f - h_0 \) can be written as
\[
R_{h_0}(x) = \frac{\omega(x)}{k!} r^{(k)}(\xi_x), \quad \omega(x) := \prod_{i=0}^{f} (x-x_i)^{p_i}, \tag{16}
\]
for some \( \xi_x \in I \), and the magnitude of its (exact) range satisfies
\[
| R_{h_0}(I) | \leq \Omega | f^{(k)}(I) |, \quad \Omega := \frac{|\omega(I)|}{k!}, \tag{17}
\]
Here we assume that the range of \( \omega(I) \) can be computed exactly, which is certainly true for small \( k \) (as in Sec. 3.1 below), but it is also possible to replace \( \omega(I) \) with some range estimate \( \Omega \omega(I) \). We now split \( f^{(k)}(I) \) into the Hermite interpolant \( h_1 \) of \( f^{(k)} \) (for the same interpolation nodes and multiplicities) and a remainder part \( R_{h_1} \). Since \( | R_{h_1}(I) | \leq \Omega | f^{(2k)}(I) | \), we obtain
\[
| f^{(k)}(I) | \leq | h_1(I) | \Omega | f^{(k)}(I) | \Omega, \tag{18}
\]
If \( f \) is \( nk \) times continuously differentiable for some \( n \geq 1 \), we may repeat this procedure (always with the same interpolation nodes \( x_i \) and multiplicities \( p_i \)) to obtain Hermite interpolants \( h_j \) of \( f^{(jk)} \) for \( j = 1, \ldots, n \). This gives a recursive remainder bound
\[
| R_{h_k}(I) | \leq \sum_{j=1}^{n} | h_j(I) | | \Omega | f^{(j)}(I) | =: \mathcal{T}_{k,n}. \tag{19}
\]
Since \( \omega(x) \) scales with \( r^k \) as \( l \) varies, we have \( \Omega \in O(r^k) \) and \( \mathcal{T}_{k,n} \in O(r^k) \). It follows that the recursive remainder form of order \( k \) and level \( n \)
\[
\Omega_{k,m}^{R} f(l) := h_0(l) + [-1, 1]\mathcal{T}_{k,n}, \tag{15}
\]
has indeed order \( k \) convergence. The minimal form \( \Omega_{k,m}^{R} f(l) \) for the smallest level \( n = 1 \) is essentially the remainder form of Cornelius and Lohner (cf. (4)), if we replace \( \omega(l) \) in (16) by \( \Omega \omega(l) \). As in Sec. 2, the advantage of higher levels of \( n \) is due to the fact that the terms of \( \mathcal{T}_{k,n} \) converge with successively higher order. Again, the maximal level \( n = \infty \) that induces the maximal recursive remainder form \( \Omega_{k,\infty} \omega(l) \), is only possible if \( \mathcal{T}_{k,n} \) is convergent, which is the case if \( f \) is analytic and \( r \) sufficiently small, or if \( f \) is a polynomial. Note that in the latter case, evaluating this form requires just a finite number of point evaluations of \( f \) and its derivatives, akin to the evaluation of the maximal Taylor forms.

3.1 Recursive Lagrange form with cubic convergence

One particular instance of the recursive remainder form (19) that will prove beneficial for \textsc{Eval} is based on the endpoints and the midpoint of \( I = [a, b] \) as simple interpolation nodes, that is, to use \( f = 2, x_0 = a, x_1 = m, x_2 = b \) in (2) and \( p_0 = p_1 = p_2 = 1 \), so that \( k = 3 \). In this setting, \( h_j \) is the Lagrange interpolant of \( f^{(3j)} \) at \( a, m, \text{ and } b \), which can be expressed in centered form as
\[
h_j^T(x) := d_{j,0} + d_{j,1}(x-m) + d_{j,2}(x-m)^2 \tag{20}
\]
with coefficients
\[
d_{j,0} := f^{(3j)}(m), \quad d_{j,1} := \frac{f^{(3j)}(b) - f^{(3j)}(a)}{2r}, \tag{21}
\]
\[
d_{j,2} = \frac{f^{(3j)}(b) - 2f^{(3j)}(m) + f^{(3j)}(a)}{2r^2}, \tag{22}
\]
Another variant of the recursive Lagrange form can be obtained
3.2 Recursive Lagrange form with quartic convergence
Another variant of the recursive Lagrange form can be obtained by applying Cornelius and Lohner’s general trick to get one extra order of convergence. To this end (cf. (20)), let
\[\hat{h}^0_t(x) := h^0_t(x) + \frac{f'''(m)}{6} \cdot \omega_3(x)\]
where \(\hat{d}_{0,1} = d_{0,1} - r \cdot \frac{f'''(m)}{6}\), and \(\hat{d}_{0,3} = \frac{f'''(m)}{6}\), be the (unique) cubic polynomial that interpolates \(f\) at \(a, m, b, \) and \(b\), like \(h^0_t\), and also matches the third derivative of \(f\) at \(m\), in the sense that \((\hat{h}^0_t)'''(m) = f'''(m)\). Similarly as above, we then have
\[|R_{h^{1,t}}(I)| \leq \Omega_3|f'''(I) - f'''(m)| = \Omega_3|\hat{f}_3(I)|,\]
where \(\hat{f}_3(x) := f'''(x) - f'''(m)\). We now split \(\hat{f}_3\) into the Lagrange interpolant
\[h^{1,t}_3(x) := h^{1,t}_3(x) - f'''(m) = (d_{1,1} + d_{1,2}(x-m))(x-m)\]
of \(\hat{f}_3\) at \(a, m, b\) and the remainder \(R_{h^{1,t}}\), which satisfies
\[|R_{h^{1,t}}(I)| \leq \Omega_3|\hat{f}_3'''(I)| = \Omega_3|f'''(I)|,\]
hence \(|\hat{f}_3(I)| \leq |\hat{h}^1_t(I)| + \Omega_3|f'''(I)|\). From here on, we repeat the splitting procedure as in the construction of \(\hat{h}^1_t\) and finally arrive at the recursive Lagrange form of level \(n\)
\[\hat{h}^n_t(I) := \hat{h}^n_t(I) + [-1, 1]T_{3,n},\]
where (cf. (22))
\[T_{3,n} := \sum_{j=1}^{n-1} |h_j^0(I)|\Omega_3 + \Omega_3^2|\Omega_3^3(I)| \in O(r^3),\]
If \(f\) is a polynomial of degree \(d\), then the maximal recursive Lagrange form
\[\hat{h}^n_t(I) := \hat{h}^n_t(I) + [-1, 1]T_{3,n}\]
deeply depends on the \(3([d/3]+1)\) values
\(f^{(3j)}(a), f^{(3j)}(m), f^{(3j)}(b), j = 0, \ldots, [d/3]\), which is comparable to the \(d+1\) values needed for the maximal Taylor forms \(\hat{h}^n_t(I)\).
As the cubic convergence of \(\hat{h}^n_t(I)\) is independent of how the
range of \(h^0_t\) is estimated for \(j \geq 1\) in (22), we can replace the exact
evaluation of \(h^0_t(I)\) by the cheaper centered form evaluation
\[\hat{h}^n_t(I) = d_{0,0} + r[-1,1]d_{0,1} + r^2[-1,1]d_{0,2}\]
This yields a less tight range function (cf. (21))
\[\hat{h}^n_t(I) + [-1, 1]T_{3,n} ,\]
where
\[T_{3,n} := \sum_{j=1}^{n-1} |d_{0,0}| + |r|d_{0,1} + r^2|d_{0,2}|\Omega_3 + \Omega_3^2|\Omega_3^3(I)| \in O(r^3),\]
which depends on the same data values as \(\hat{h}^n_t(I)\). In the context of EVAL, this increases the size of the subdivision tree slightly, but seems to be more efficient in terms of running time (see Sec. 5).

4 REAL ROOT ISOLATION WITH EVAL AND THE NEW RANGE FUNCTIONS
4.1 Advantage of the Lagrange form in EVAL
What is to recommend the generalized Taylor form or the recursive Lagrange form? We give the intuition for the advantages of the Lagrange form in the context of root isolation with EVAL for polynomials. Recall that computing the maximal Taylor form \(\hat{h}^n_t(I)\) for a polynomial of degree \(d\) requires us to evaluate \(f^{(i)}\) at \(m = m(I)\) for \(i = 0, \ldots, d\). To compute the maximal recursive Lagrange form \(\hat{h}^n_t(I)\), we must evaluate \(f^{(3j)}\) at \(a, m, b,\) where \(i = [a, b]\) for \(j = 0, \ldots, [d/3]\). Considered in isolation, the two forms are comparable in computational complexity, since they each need about \(d\) function or derivative evaluations. But in the context of the EVAL algorithm, the Lagrange form begins to shine: after estimating the range of \(f\) over \([a, b]\), we would typically need to further estimate the ranges over \([a, m]\) and \([m, b]\). For the Lagrange form, estimating the range over \([a, m]\) needs only \([d/3] + 1\) additional evaluations of \(f^{(3j)}\) at \((a + m)/2\), since we already computed \(f^{(3j)}(a)\) and \(f^{(3j)}(b)\). In contrast, the Taylor form must still make \(d+1\) evaluations of \(f\) and its derivatives at \((a + m)/2\). A similar remark holds for \([m, b]\). Therefore, we may expect a roughly 3-fold speed up of EVAL when using the Lagrange instead of the Taylor form, although
we should keep in mind that the performance is also influenced by other factors. For example, the tightness of the two forms is not identical and the Lagrange form requires a more elaborate memory management so that some of the data needed for processing \( [a, m] \) and \([m, b] \) can be inherited from the data computed for \( [a, b] \).

4.2 Range functions for derivatives

Before presenting the results of our numerical experiments, there is one more issue that needs to be dealt with: EVAL not only needs to estimate the range of \( f \) over \( I \), but also the range of \( f' \).

For the generalized Taylor form, a simple calculation shows that the generalized Taylor form (of level \( n-1 \) applied to \( f' \)) is

\[
\mathcal{D}_{k,n-1} f' (l) = g'_k (l) + r^k [-1, 1] S'_{k,n-1}, \quad S'_{k,n-1} = \sum_{i=k+1}^{n} \left|i \right| c_i r^{i-k-1},
\]

where \( g_k \) is the \( k \)-th order Taylor polynomial of \( f \) about \( m \), that is, \( g_k (x) = \sum_{i=1}^{k} c_i (x - m)^{i-1} \), and the \( c_i \) are defined as in (9).

Therefore, \( \mathcal{D}_{k,n} f (l) \) and \( \mathcal{D}_{k,n-1} f' (l) \) both have order \( k \) convergence and depend on the same data.

For the Lagrange form, it is more complicated, since \( \mathcal{D}_{k,n} f' (l) \) depends on the evaluation of \( f^{(3j+1)} \) at \( a, m \) and \( b \) and would thus double the computational cost. To re-use the data needed for computing \( \mathcal{D}_{k,n} f (l) \), we recall a result by Shadrin [27], which asserts that the error between the \( k \)-th derivative of \( f \) and the \( k \)-th derivative of the Lagrange polynomial \( h (x) \) that interpolates \( f \) at the \( \ell + 1 \) nodes \( x_0, \ldots, x_\ell \) in \( I \) satisfies

\[
\left| f^{(k)} (x) - h^{(k)} (x) \right| \leq |\omega^{(k)} (I)| \frac{|f^{(\ell+1)} (I)|}{\ell!}, \quad x \in I,
\]

for \( k = 0, \ldots, \ell \) and \( \omega (x) = \prod_{i=0}^{\ell} (x - x_i) \). In the context of \( \mathcal{D}_{k,n} \), this implies

\[
\left| f^{(k)} (x) - (h'_k) (x) \right| \leq |\omega'_k (I)| \frac{|f''' (I)|}{6}, \quad x \in I.
\]

Since \( |\omega'_k (I)| = r^k [1-2] \) and \( \Omega_3 |f''' (I)| \leq T_{3,n} \), we conclude that \( f' (l) \) can be estimated by the recursive Lagrange forms

\[
\mathcal{D}_{k,n} f' (l) := (h'_k) (l) + \frac{3 \sqrt{3}}{r} [-1, 1] T_{3,n},
\]

and

\[
\mathcal{D}_{k,n} f' (l) := (h'_k) (l) + \frac{3 \sqrt{3}}{r} [-1, 1] T_{3,n},
\]

which have only quadratic convergence, but depend on the same data as \( \mathcal{D}_{k,n} f (l) \) and \( \mathcal{D}_{k,n} f' (l) \). Note that we cannot derive a similar range function for \( f' \) with cubic convergence from \( \mathcal{D}_{k,n} \), because \( h' \) is not a Lagrange interpolant and Shadrin’s result does not apply.

Table 1: Combinations of range functions for \( f \) and \( f' \) used by EVAL in our experiments.

<table>
<thead>
<tr>
<th>( f )</th>
<th>( E_T )</th>
<th>( E_T' )</th>
<th>( E_T'' )</th>
<th>( E_T''' )</th>
<th>( E_T'''' )</th>
<th>( E_T''''' )</th>
<th>( E_T'''''' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>range of ( f )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
</tr>
<tr>
<td>range of ( f' )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
<td>( \mathcal{D}_{k,n} )</td>
</tr>
</tbody>
</table>

5 NUMERICAL EXPERIMENTS

We implemented a general version of the EVAL procedure (see Algo. 1) in C++ and derived from it eight versions (see Table 1) that differ by the concrete range functions used for estimating the ranges of \( f \) and \( f' \) in lines 5 and 6. The first version \( E_T \) estimates both ranges with the minimal Taylor form (cf. (7) in Sec. 2). The next three versions \( E_T' \) for \( k = 2, 3, 4 \) employ the order-\( k \) convergent Taylor form for both ranges (see Sec. 2). The remaining four versions use recursive Lagrange forms with cubic or quartic convergence (see Secs. 3.1 and 3.2) to estimate the range of \( f \) and the recursive Lagrange form with quadratic convergence (see Sec. 4.2) for \( f' \). Note that the version \( E_T'' \) represents the state-of-the-art of EVAL [5] and serves as the “baseline” for performance. Except for \( E_T \), all these Taylor and Lagrange forms are the maximal versions.

The input data for our experiments come from four representative families of integer polynomials: dense with all roots real (Chebyshev, \( T_n \) and Hermite, \( H_n \), dense with only 2 real roots in \( I \) (Mignotte cluster, \( M_{2k+1} = x^{2k+1} - 2(4x^2 - 1)^k \), from [12]) and sparse without real roots (\( S_n (x) = 1 + x + \sum_{i=0}^{n} \frac{x^i}{i!} \)). Note that these polynomials do not have multiple roots, a prerequisite for EVAL’s halting. Our implementation, including these data and experiments, may be downloaded from the Core Library webpage [6, 33].

We summarize the results of our experiments in three tables, with columns grouped by convergence order. Table 2 reports the size of the EVAL subdivision tree for the various polynomials. It is a good measure of the tightness of the various range functions, since the size of the recursion tree is inversely proportional to the tightness of the range functions used. In each row, we underscore the smallest tree size, which is always achieved by \( E_T \). In general, we observe that the tree size decreases as the convergence order of the range functions increases and that the “cheaper” variants of the recursive Lagrange forms lead to (slightly) larger subdivision trees. The difference between the tree sizes for the Taylor and Lagrange
Table 3: Average running time of the Eval algorithm with 1024-bit floating point arithmetic in seconds.

<table>
<thead>
<tr>
<th>$f$</th>
<th>$l_0$</th>
<th>$T_2$</th>
<th>$E_2^T$</th>
<th>$T_3$</th>
<th>$E_3^T$</th>
<th>$T_4$</th>
<th>$E_4^T$</th>
<th>$T_4'$</th>
<th>$E_4'^T$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{20}$</td>
<td>0.2005</td>
<td>0.02917</td>
<td>0.01966</td>
<td>0.02115</td>
<td>0.01656</td>
<td>0.02004</td>
<td>0.02255</td>
<td>0.01758</td>
<td>1.76</td>
<td></td>
</tr>
<tr>
<td>$T_{30}$</td>
<td>123.1</td>
<td>0.1928</td>
<td>0.1305</td>
<td>0.1083</td>
<td>0.0837</td>
<td>0.1320</td>
<td>0.1127</td>
<td>0.0868</td>
<td>2.30</td>
<td></td>
</tr>
<tr>
<td>$T_{40}$</td>
<td>1.520</td>
<td>1.026</td>
<td>0.659</td>
<td>0.534</td>
<td>0.964</td>
<td>0.643</td>
<td>0.519</td>
<td>2.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{60}$</td>
<td>13.28</td>
<td>8.86</td>
<td>4.74</td>
<td>3.95</td>
<td>8.27</td>
<td>4.65</td>
<td>3.88</td>
<td>3.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{80}$</td>
<td>159.8</td>
<td>104.8</td>
<td>52.4</td>
<td>45.7</td>
<td>94.9</td>
<td>50.7</td>
<td>44.1</td>
<td>3.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H_{20}$</td>
<td>0.1024</td>
<td>0.02337</td>
<td>0.01716</td>
<td>0.01779</td>
<td>0.01378</td>
<td>0.01639</td>
<td>0.01968</td>
<td>0.01521</td>
<td>1.70</td>
<td></td>
</tr>
<tr>
<td>$H_{30}$</td>
<td>10.37</td>
<td>0.1364</td>
<td>0.1010</td>
<td>0.0871</td>
<td>0.0660</td>
<td>0.1018</td>
<td>0.0897</td>
<td>0.0683</td>
<td>2.07</td>
<td></td>
</tr>
<tr>
<td>$H_{40}$</td>
<td>0.7977</td>
<td>0.725</td>
<td>0.484</td>
<td>0.379</td>
<td>0.632</td>
<td>0.494</td>
<td>0.389</td>
<td>2.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H_{60}$</td>
<td>6.80</td>
<td>5.44</td>
<td>3.02</td>
<td>2.37</td>
<td>5.19</td>
<td>3.06</td>
<td>2.39</td>
<td>2.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H_{80}$</td>
<td>71.7</td>
<td>61.8</td>
<td>31.9</td>
<td>25.9</td>
<td>47.6</td>
<td>31.9</td>
<td>25.1</td>
<td>2.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_{12}$</td>
<td>0.9342</td>
<td>0.01787</td>
<td>0.009825</td>
<td>0.01176</td>
<td>0.008525</td>
<td>0.009681</td>
<td>0.01172</td>
<td>0.009060</td>
<td>2.10</td>
<td></td>
</tr>
<tr>
<td>$M_{14}$</td>
<td>0.1047</td>
<td>0.05708</td>
<td>0.05195</td>
<td>0.03939</td>
<td>0.05636</td>
<td>0.05217</td>
<td>0.04041</td>
<td>2.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_{16}$</td>
<td>0.7824</td>
<td>0.4081</td>
<td>0.3086</td>
<td>0.2459</td>
<td>0.4023</td>
<td>0.3012</td>
<td>0.2349</td>
<td>3.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_{18}$</td>
<td>6.937</td>
<td>3.707</td>
<td>2.258</td>
<td>1.887</td>
<td>3.630</td>
<td>2.184</td>
<td>1.786</td>
<td>3.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_{21}$</td>
<td>85.82</td>
<td>43.78</td>
<td>25.58</td>
<td>21.94</td>
<td>42.03</td>
<td>24.49</td>
<td>20.65</td>
<td>3.91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{100}$</td>
<td>1.039</td>
<td>1.180</td>
<td>0.615</td>
<td>0.509</td>
<td>0.404</td>
<td>0.596</td>
<td>0.500</td>
<td>0.393</td>
<td>2.92</td>
<td></td>
</tr>
<tr>
<td>$S_{200}$</td>
<td>8.019</td>
<td>11.17</td>
<td>5.70</td>
<td>3.87</td>
<td>3.24</td>
<td>5.52</td>
<td>3.72</td>
<td>3.09</td>
<td>3.45</td>
<td></td>
</tr>
<tr>
<td>$S_{400}$</td>
<td>103.4</td>
<td>154.0</td>
<td>76.2</td>
<td>45.8</td>
<td>41.1</td>
<td>73.8</td>
<td>43.6</td>
<td>39.7</td>
<td>3.75</td>
<td></td>
</tr>
<tr>
<td>$S_{800}$</td>
<td>1556.2</td>
<td>2322</td>
<td>1160</td>
<td>636</td>
<td>589</td>
<td>1123</td>
<td>569</td>
<td>561</td>
<td>3.94</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Average running time of the Eval algorithm with multi-precision rational arithmetic in seconds.

<table>
<thead>
<tr>
<th>$f$</th>
<th>$l_0$</th>
<th>$E_2^T$</th>
<th>$E_3^T$</th>
<th>$E_4^T$</th>
<th>$E_4'^T$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{20}$</td>
<td>1039</td>
<td>1180</td>
<td>615</td>
<td>509</td>
<td>404</td>
<td>2.92</td>
</tr>
<tr>
<td>$T_{30}$</td>
<td>8019</td>
<td>1117</td>
<td>570</td>
<td>387</td>
<td>324</td>
<td>3.45</td>
</tr>
<tr>
<td>$T_{40}$</td>
<td>103.4</td>
<td>154.0</td>
<td>76.2</td>
<td>45.8</td>
<td>41.1</td>
<td>3.75</td>
</tr>
<tr>
<td>$T_{80}$</td>
<td>1556.2</td>
<td>2322</td>
<td>1160</td>
<td>636</td>
<td>589</td>
<td>3.94</td>
</tr>
</tbody>
</table>

versions of Eval with the same convergence order is mainly due to the inferior Recursive Lagrange form with only quadratic convergence that is used for $f'$. In fact, if we use $\mathbb{Q}^2$ instead of $\mathbb{Q}$ for the range of $f'$ in $\mathbb{L}$, then the tree sizes are almost identical to those of $\mathbb{L}$, and likewise for $\mathbb{L}$ versus $\mathbb{L}$. However, the price of larger subdivision trees seems to be well compensated for when it comes to the actual performance of the different Eval variants.

Our experimental platform is a Windows 10 laptop with 1.8 GHz Intel Core i7-8565U processor and 16 GB RAM. The average running times (over 1600 runs for $T_n$, $H_n$, $M_{12k+1}$, and 4000 runs for $S_n$) of our eight versions of Eval on our list of 19 polynomials are obtained by using two kinds of computer arithmetic: 1024-bit floating point arithmetic (Table 3) and multi-precision rational arithmetic (Table 4). No times (and tree sizes in Table 2) are reported, if an Eval version did not terminate within 1 hour. Both arithmetic variants come from the multiple-precision arithmetic library GMP [10]. For rational arithmetic, we replaced the constant $\sqrt{3}$ in the definitions of $\Omega_3$ and $\Omega_4$ with the slightly larger rational number 17320508075688773/10^{16}, so that the validity of the bounds is not altered. Moreover, we temporarily switch to 1024-bit floating point arithmetic for computing square roots. The latter is unavoidable when computing the exact ranges of cubic polynomials (see Sec. A.3) and thus needed by the range functions with quartic convergence.
We draw several conclusions from the tables: 1) The EVAL version $E^2_T$ based on minimal forms may be utterly non-competitive with the maximal form $E^3_T$ (the former timed out after 1 hour for degrees $n > 40$ for the first 3 sets of polynomials). We expect the same conclusion for other minimal forms. 2) The EVAL versions based on recursive Lagrange forms outperform the ones based on Taylor forms with the same convergence order, despite the larger subdivision trees. We attribute this to the fewer numbers (about one-third) of derivative values that are computed. 3) It does not pay to use range functions with quartic convergence order, because the overhead of computing Taylor expansion and Lagrange interpolation. We see that this re-

$\text{REFERENCES}$


A EXACT RANGES FOR LOW DEGREE POLYNOMIALS

Let \( I = [a, b], r = (b - a)/2, m = (a + b)/2, \) and
\[
A := \min \{g(a), g(b)\}, \quad B := \max \{g(a), g(b)\}. \tag{30}
\]

A.1 Linear polynomials

The exact range \( g(I) = [A, B] \) of the linear polynomial \( g_1(x) = c_0 + c_1(x - m) \) is given by the assignment in (30), because \( g \) is monotonic. The range can also be expressed as
\[
g(I) = c_0 + r[-1, 1]c_1. \tag{31}
\]

A.2 Quadratic polynomials

To determine the exact range \( g(I) = [A, B] \) of the quadratic polynomial \( g(x) = c_0 + c_1(x - m) + c_2(x - m)^2 \) we first observe that the extremum of \( g \) occurs at
\[
x^* = m - \frac{c_1}{2c_2},
\]
which is inside \( I \), if and only if \( |c_1| < 2|c_2| \). If \( x^* \notin I \), then \( g \) is monotonic on \( I \) and the assignment in (30) gives the correct range. Otherwise, we check the sign of \( c_2 \) to see if \( g \) has a minimum \( c_2 > 0 \) or a maximum \( c_2 < 0 \) at \( x^* \) and accordingly replace \( A \) or \( B \) with
\[
g(x^*) = c_0 - \frac{c_1^2}{4c_2}.
\]

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

A.3 Cubic polynomials

To find the exact range \( g(I) = [A, B] \) of the cubic polynomial \( g(x) = c_0 + c_1(x - m) + c_2(x - m)^2 + c_3(x - m)^3 \), we assume that \( c_3 \neq 0 \). If \( c_3 = 0 \), then \( g(x) \) is a polynomial of degree at most two and its range can be found with the method in Sec. A.2.

We now analyze the stationary points of \( g \) by considering the quadratic equation
\[
g'(x) = c_1 + 2c_2(x - m) + 3c_3(x - m)^2 = 0
\]
and in particular its discriminant
\[
\Delta = c_2^2 - 3c_1c_3.
\]

If \( \Delta < 0 \), then \( g \) does not have any local extrema, and if \( \Delta = 0 \), then \( g \) has a stationary point of inflection. In both cases, \( g \) is monotonic and its range is given by the assignment in (30).

If \( \Delta > 0 \), then \( g \) has a local minimum at \( x^- \) and a local maximum at \( x^+ \), where
\[
x^\pm = m - \frac{c_2 \mp \sqrt{\Delta}}{3c_3} \tag{31}
\]

To determine the range of \( g \), we need to know whether \( x^- \) and \( x^+ \) are inside or outside \( I \) and must distinguish four cases:

1. If \( x^-, x^+ \in (a, b) \), then \( g(I) = [\min\{A, g(x^-)\}, \max\{B, g(x^+)\}] \).
2. If \( x^- \notin (a, b) \) and \( x^+ \notin (a, b) \), then \( g(I) = [g(x^-), B] \).
3. If \( x^+ \notin (a, b) \) and \( x^- \notin (a, b) \), then \( g(I) = [A, g(x^+)] \).
4. If \( x^-, x^+ \notin (a, b) \), then \( g \) is monotonic over \( I \) and \( g(I) = [A, B] \).