# Computing Real Roots of Real Polynomials

Michael Sagraloff (joined work with Kurt Mehlhorn)

max planck institut
informatik

# Isolating Real Roots using the Descartes Method

## Problem

Given a (square-free) polynomial $f \in \mathbb{R}[x]$, compute disjoint intervals $I_1, \ldots, I_m$ (rational endpoints) such that each $I_j$ contains exactly one root and their union covers all real roots.

## The Descartes Method

Recursive interval bisection using Descartes' Rule of Signs to test for roots.

- Easy to understand and to implement
- Performs very well in practice
- Well suited for exact and complete implementation
- It is integrated in many computer algebra systems (e.g., MAPLE, SAGE, CGAL,...).

# The Descartes Method

## Descartes' Rule of Signs for Intervals

For an interval $I = (a, b)$ and $n := \deg f$, let

$$f_I(x) = (x+1)^n \cdot f\left(\frac{ax+b}{x+1}\right) = \sum_{i=0}^{n} c_i x^i$$

and $v := \mathrm{var}(f, I)$ the number of sign variations in $(c_0, \ldots, c_n)$.
Then, for the number $m$ of real roots in $I$, it holds that

- $m \leq v$, and $m \equiv v \bmod 2$.
- In particular, $v \leq 1$ implies $m = v$.

**Example:** $f(x) = x^3 - 2x^2 - x + 1$ and $I = (1/2, 4)$.

Then, $f_I(x) = +(1/8)x^3 - (15/2)x^2 - (43/2)x + 29$, and thus $v = 2$.

$\Rightarrow$ $f$ has 0 or 2 real roots in $I$.

# The Descartes Method

## Descartes' Rule of Signs for Intervals

For an interval $I = (a, b)$ and $n := \deg f$, let

$$f_I(x) = (x+1)^n \cdot f\left(\frac{ax+b}{x+1}\right) = \sum_{i=0}^{n} c_i x^i$$

and $v := \mathrm{var}(f, I)$ the number of sign variations in $(c_0, \dots, c_n)$.
Then, for the number $m$ of real roots in $I$, it holds that

- $m \leq v$, and $m \equiv v \bmod 2$.
- In particular, $v \leq 1$ implies $m = v$.

**Example:** $f(x) = x^3 - 2x^2 - x + 1$ and $I = (1/2, 4)$.

Then, $f_I(x) =$ **+**$(1/8)x^3$**-**$(15/2)x^2$**-**$(43/2)x$**+**$29$, and thus $v = 2$.

$\Rightarrow$ $f$ has 0 or 2 real roots in $I$.

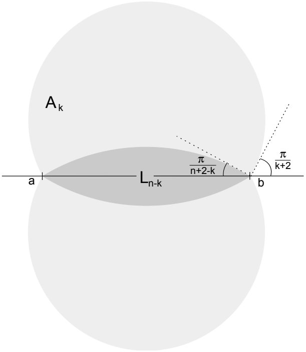
# Some Important Properties

**Sign variation diminishing property:** For any two disjoint intervals $I_1, I_2 \subset I$, we have

$$\mathrm{var}(f, I) \geq \mathrm{var}(f, I_1) + \mathrm{var}(f, I_2)$$

**Generalization of the One- and Two-Circle Theorems:**

[Obreshkoff 1963]




Let $I = (a, b)$ be an interval, then

$\#$ roots in $L_{n-k} \geq k \Rightarrow \mathrm{var}(f, I) \geq k$

$\#$ roots in $A_k \leq k \Rightarrow \mathrm{var}(f, I) \leq k$

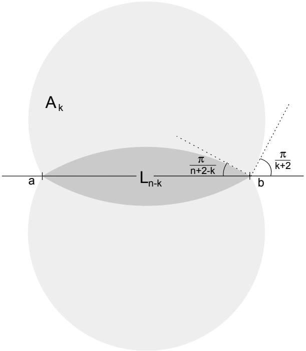
# Some Important Properties

**Sign variation diminishing property:** For any two disjoint intervals $I_1, I_2 \subset I$, we have

$$\mathrm{var}(f, I) \geq \mathrm{var}(f, I_1) + \mathrm{var}(f, I_2)$$

**Generalization of the One- and Two-Circle Theorems:**

[Obreshkoff 1963]



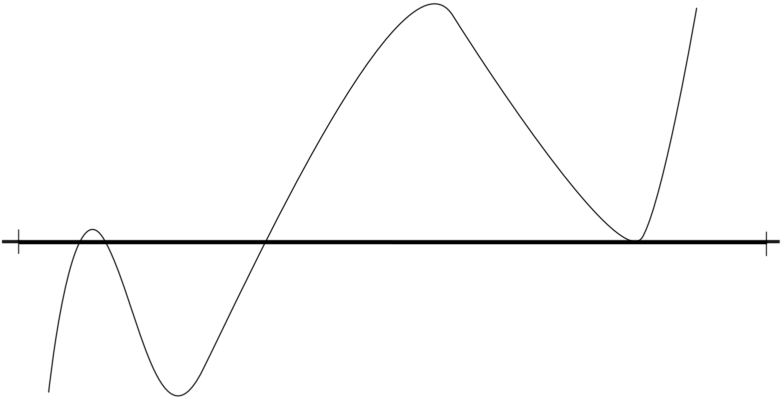

Let $I = (a, b)$ be an interval, then

$$\mathrm{var}(f, I) \geq \#\text{ roots in } L_n$$

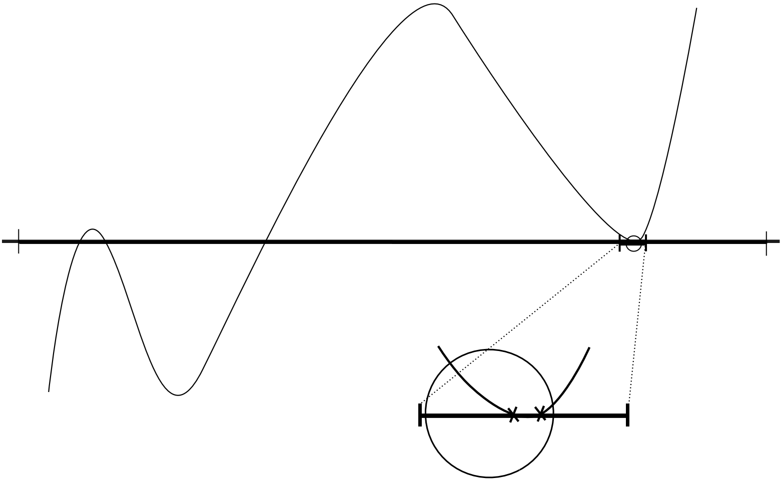
$$\mathrm{var}(f, I) \leq \#\text{ roots in } A_n$$

We denote $L_n$ and $A_n$ the *Obreshkoff Lens* and the *Obreshkoff Area of I*, respectively.

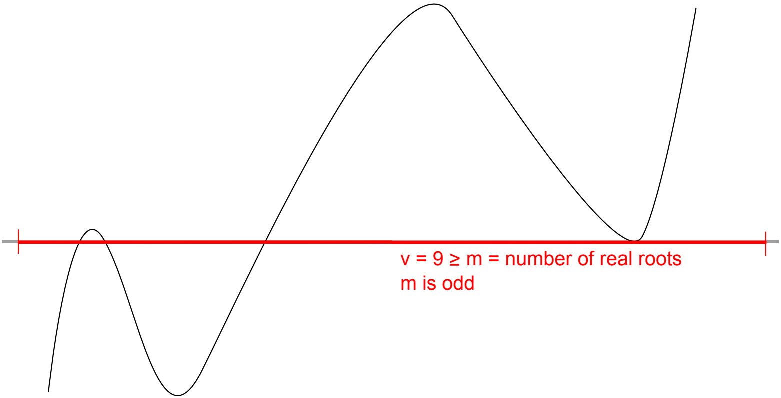
# The Descartes Method



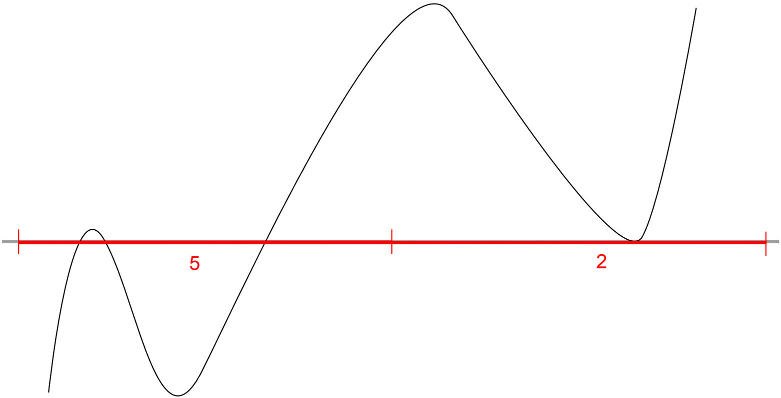
# The Descartes Method



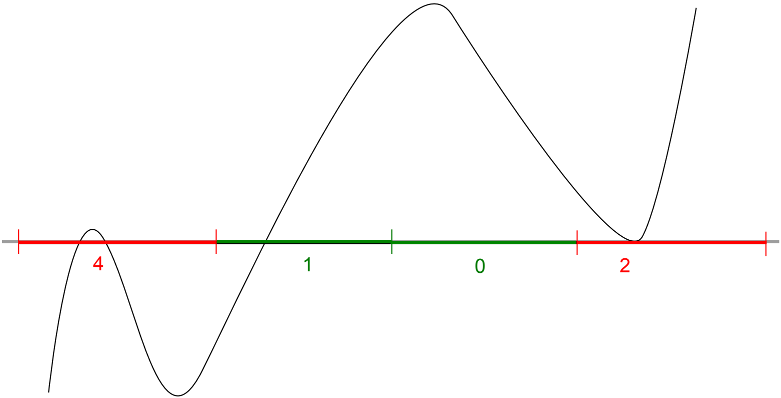
# The Descartes Method



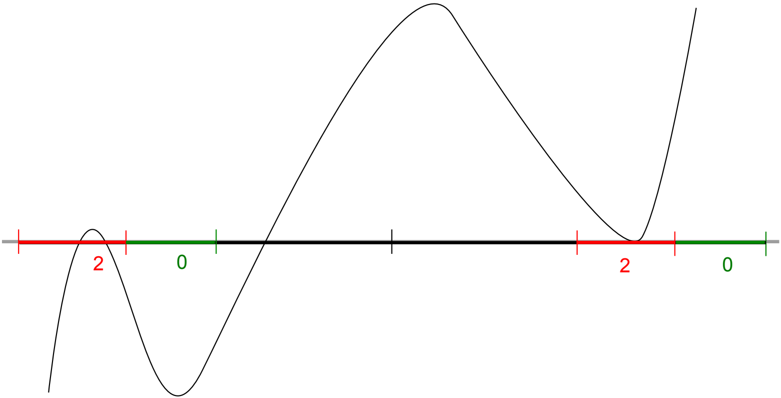
# The Descartes Method



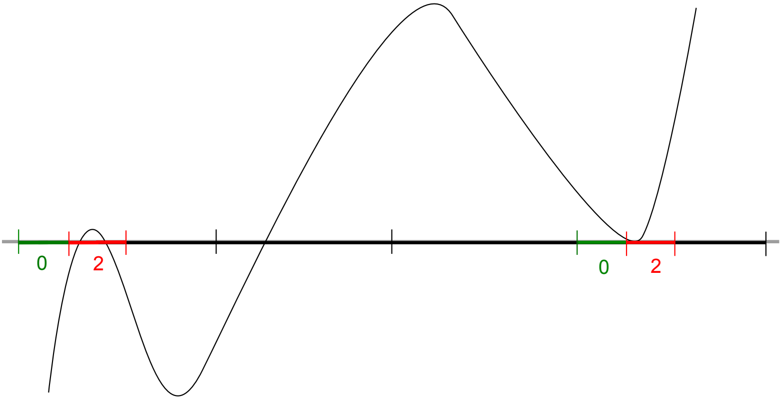
# The Descartes Method



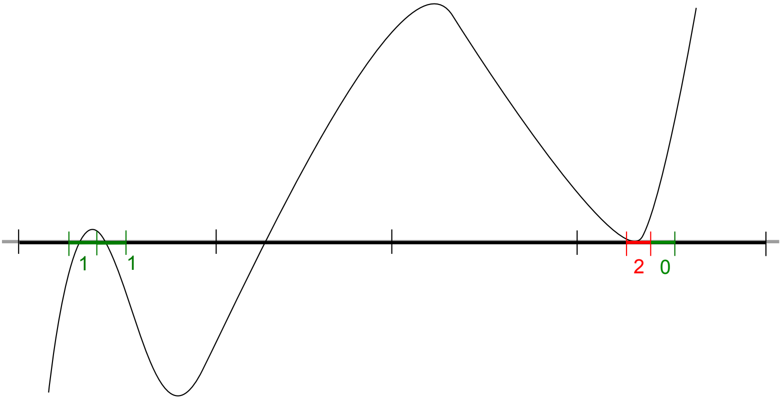
# The Descartes Method



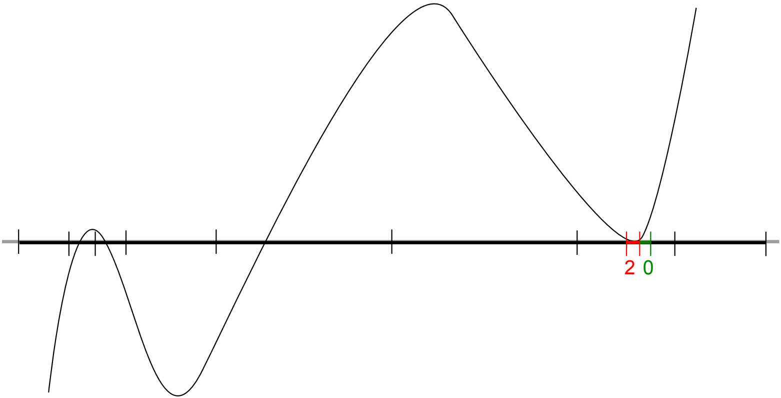
# The Descartes Method



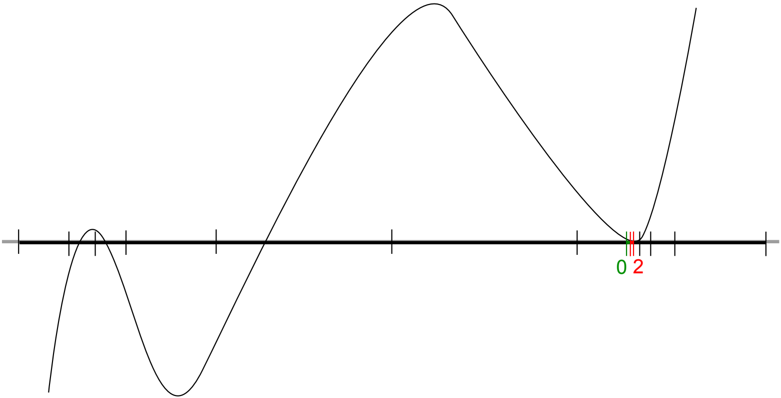
# The Descartes Method



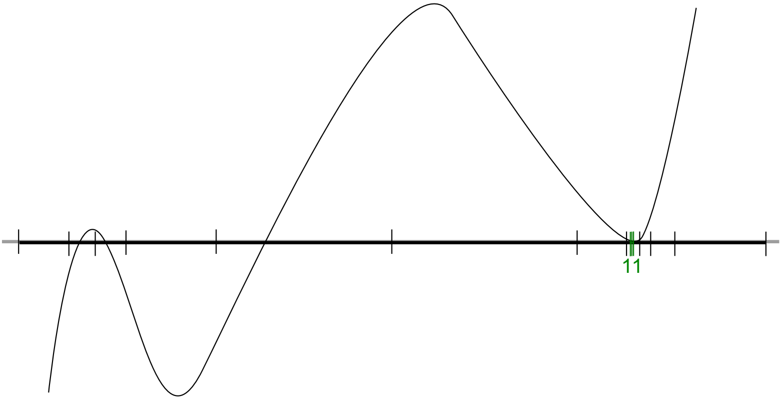
# The Descartes Method



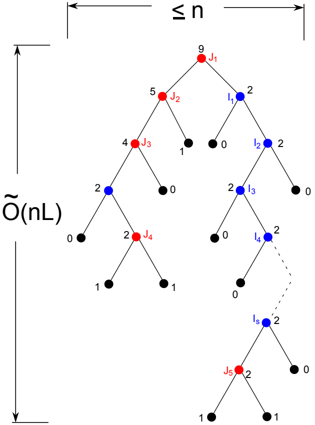
# The Descartes Method



# The Descartes Method



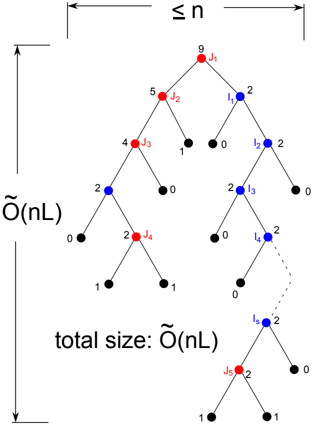
# Analysis of the Descartes Method




Polynomial $f$ of degree $n$ with integer coefficients of bitsize $\leq L$:

- Distance between roots: $2^{-\tilde{O}(nL)}$

# Analysis of the Descartes Method

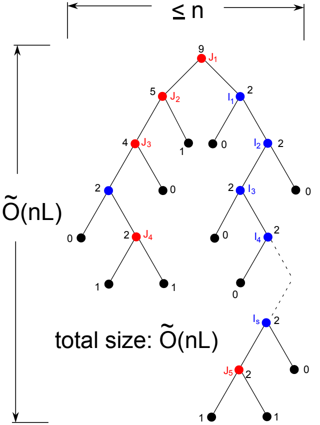



Polynomial $f$ of degree $n$ with integer coefficients of bitsize $\leq L$:

- Distance between roots: $2^{-\tilde{O}(nL)}$
- Only few roots have small distance to each other

[Eigenwillig et al. 2006]

# Analysis of the Descartes Method



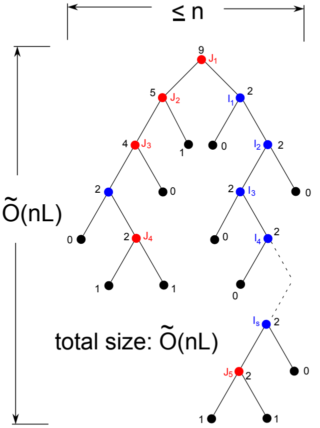

Polynomial $f$ of degree $n$ with integer coefficients of bitsize $\leq L$:

- Distance between roots: $2^{-\tilde{O}(nL)}$
- Only few roots have small distance to each other

[Eigenwillig et al. 2006]

- $f_I(x)$ **has bitsize** $\tilde{O}(n^2L)$, computational cost at each node: $\tilde{O}(n^3L)$

# Analysis of the Descartes Method



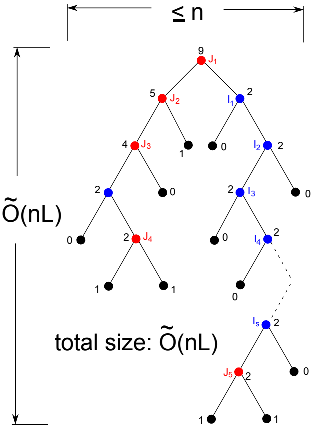

Polynomial $f$ of degree $n$ with integer coefficients of bitsize $\leq L$:

- Distance between roots: $2^{-\tilde{O}(nL)}$
- Only few roots have small distance to each other

[Eigenwillig et al. 2006]

- $f_I(x)$ **has bitsize** $\tilde{O}(n^2L)$, computational cost at each node: $\tilde{O}(n^3L)$
- Total cost: $\tilde{O}(n^4L^2)$

# Analysis of the Descartes Method




Polynomial $f$ of degree $n$ with integer coefficients of bitsize $\leq L$:

- Distance between roots: $2^{-\tilde{O}(nL)}$
- Only few roots have small distance to each other

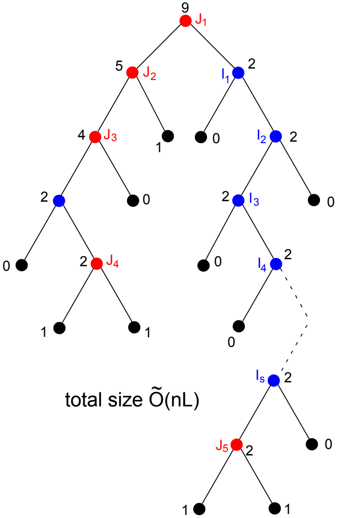
[Eigenwillig et al. 2006]

- $f_I(x)$ **has bitsize** $\tilde{O}(n^2 L)$, computational cost at each node: $\tilde{O}(n^3 L)$
- Total cost: $\tilde{O}(n^4 L^2)$

**Precision $\mathbf{n^2L}$ is needless!** *Approximate but certified computation* with precision **nL** suffices. $\Rightarrow$ total cost $\tilde{O}(n^3 L^2)$ (one of the reasons why MAPLE's "solve" is so fast!)

[Rouillier, Zimmermann 2004], [S. 2010]

# Can we improve upon bisection?



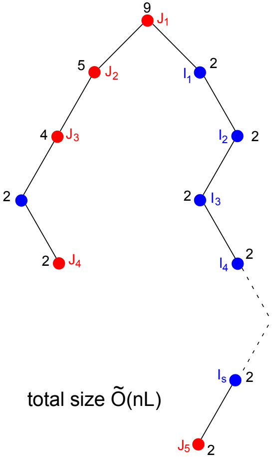

We denote a node $I$ in the subdivision tree $\mathcal{T}$ (starting internal $I_0$)

- a **milestone** if $I = I_0$, or each child of $I$ counts less sign variations than $I$,
- **terminal** if $\text{var}(f, I) \leq 1$, and
- **ordinary**, otherwise.

$n' := \#$ of milestones $\leq \text{var}(f, I_0) \leq n$,

($\sum_I \text{var}(f, I) - \#\{I : \text{var}(f, I) > 0\}$ is non-negative and decreases by at least one at each milestone.)

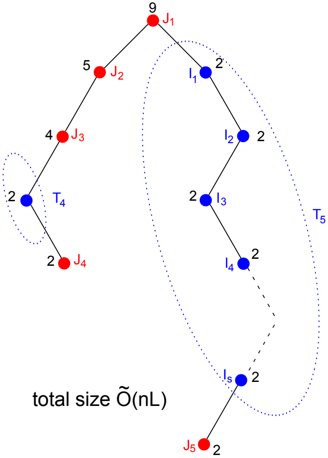
# Can we improve upon bisection?




Consider the subtree $\mathcal{T}'$ of $\mathcal{T}$ obtained from removing the terminal nodes of $\mathcal{T}$. $\mathcal{T}'$ partitions into

- **milestones** $J_1, \ldots, J_{n'}$, and

# Can we improve upon bisection?



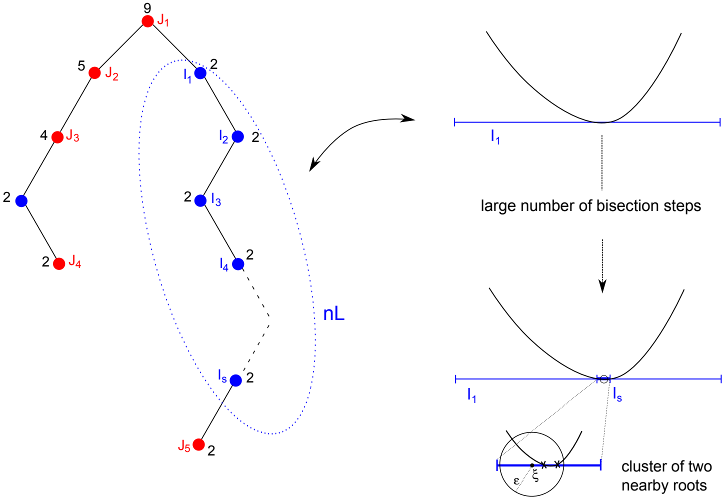

Consider the subtree $\mathcal{T}'$ of $\mathcal{T}$ obtained from removing the terminal nodes of $\mathcal{T}$. $\mathcal{T}'$ partitions into

- **milestones** $J_1, \ldots, J_{n'}$, and
- **chains** $T_i$ of ordinary nodes connecting the milestone $J_i$ with a unique $J_k \supset J_i$

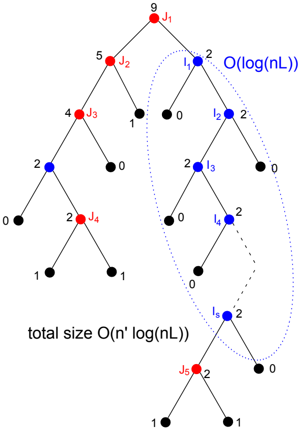
$$|\mathcal{T}| = O(|\mathcal{T}'|) = O(n') + O(\sum_i |T_i|)$$

For the bisection strategy, some of the chains $T_i$ may have length $nL$ (e.g., Mignotte polynomial).

# Can we improve upon bisection?



# Idea: Combine Descartes and Newton iteration




- Newton iteration for multiple roots (cluster of $k$ roots behaves similarly as a $k$-fold root)
- Bisection only if Newton "fails"
- Similar subdivision strategy as in Abbott's QIR method to further refine isolating intervals.

[Abbott 2006],[Kerber and S. 2011]

- Quadratic convergence except for $O(\log(nL))$ many in each chain
- **Tree size reduces by factor L**
- treesize is only logarithmic for sparse polynomials!

# Newton Iteration

Let $\xi$ be a $k$-fold root of $f$.

- If $x_0$ is sufficiently close to $\xi$ (compared to the remaining roots of $f$), then the sequence

$$x_i := x_{i-1} - k \cdot \frac{f(x_{i-1})}{f'(x_{i-1})}$$

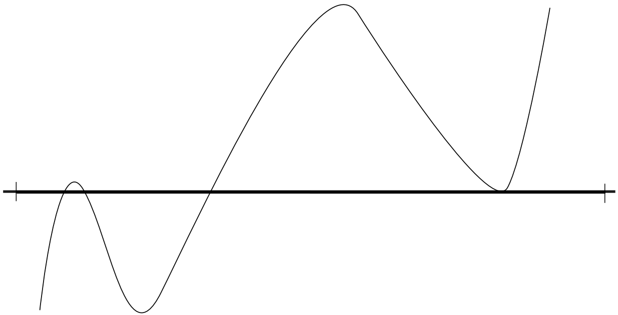
  converges quadratically to $\xi$.
- Applies also to a cluster $\mathcal{C}$ of $k$ nearby roots at $\xi$
- Cluster must be well separated from the remaining roots
- $x_i$ must be separated from the cluster

**Algorithmic Problem**

How can we test in our subdivision algorithm whether such a situation is given?

# Algorithm NEWDSC: A Trial and Error Approach

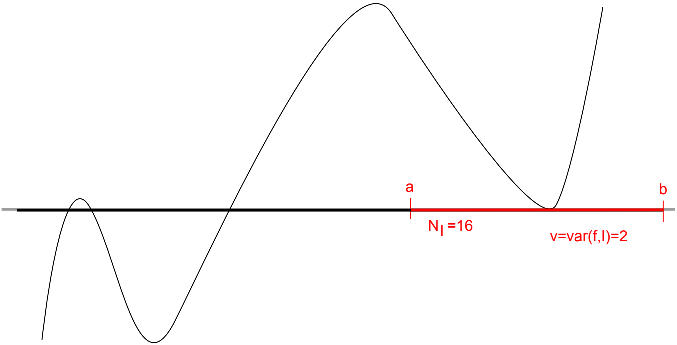
For a given $f = \sum_{i=0}^{n} a_i x^i \in \mathbb{Z}[x]$, $|a_i| < 2^L$, $I_0 := (-2^{L+1}, 2^{L+1})$ contains all real roots of $f$. Let $N_{I_0} := 4$, $\mathcal{A} := \{(I_0, N_{I_0})\}$, $\mathcal{O} := \emptyset$.



# Algorithm NewDsc: A Trial and Error Approach

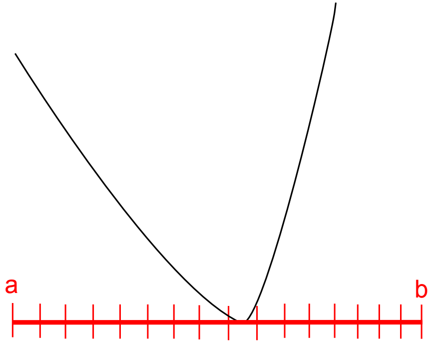
In each iteration, pick some $(I, N_I) \in \mathcal{A}$ (and remove it from $\mathcal{A}$)

- If $v := \text{var}(f, I) = 0$, do nothing. If $v = 1$, add $I$ to $\mathcal{O}$. If $v > 1$:
- Determine a $k^* \in \{1, \ldots, n\}$ such that if there exists a cluster of $k$ roots, then $k^* = k$: Use the fact that, in the latter case, $t - k \cdot \frac{f(t)}{f'(t)} \approx t' - k \cdot \frac{f(t')}{f'(t')}$ for most pairs of points $t, t' \in I$.



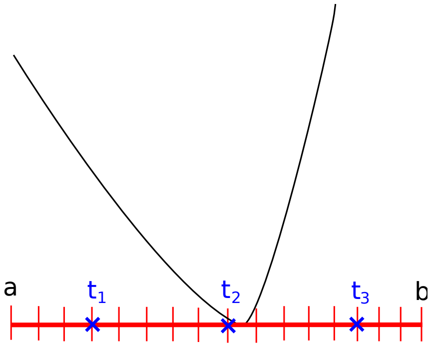
# Algorithm NEWDSC: A Trial and Error Approach

(Conceptually) subdivide $I$ into $N_I$ equally sized subintervals
$I' = (a + \ell \cdot \frac{w(I)}{N_I}, a + (\ell + 1) \cdot \frac{w(I)}{N_I})$



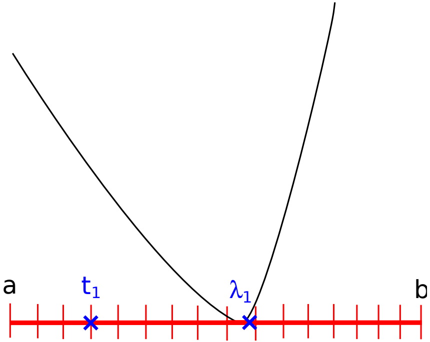
# Algorithm NEWDSC: A Trial and Error Approach

- Consider well distributed sample points $t_1, t_2, t_3 \in I$



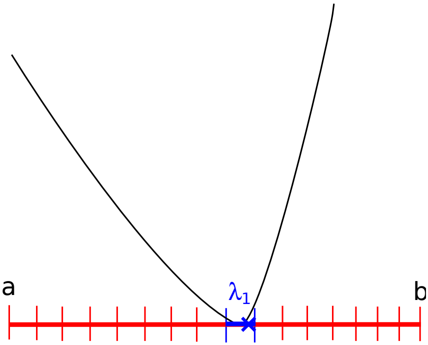
# Algorithm NewDsc: A Trial and Error Approach

- Consider well distributed sample points $t_1, t_2, t_3 \in I$
- Compute $\lambda_i := t_i - k^* \cdot \frac{f(t_i)}{f'(t_i)}$



# Algorithm NEWDSC: A Trial and Error Approach

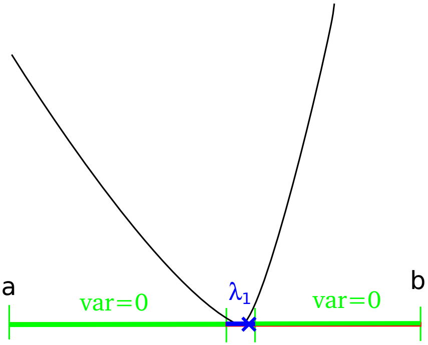
- Consider well distributed sample points $t_1, t_2, t_3 \in I$
- Compute $\lambda_i := t_i - k^* \cdot \frac{f(t_i)}{f'(t_i)}$
- Determine corresponding subinterval $I_i' = (a_i', b_i')$ (if existent) that contains $\lambda_i$



# Algorithm NEWDSC: A Trial and Error Approach

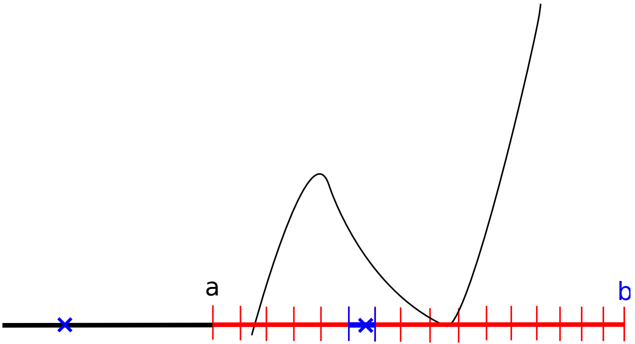
- Let $v_{i,\ell} := \text{var}(f,(a,a_i'))$ and $v_{i,r} := \text{var}(f,(b_i',b))$.
- If there exists an $i$ with $v_{i,\ell} = v_{i,r} = 0$, add $(I_i', N_{I_i'}) := (I_i', N_I^2)$ to $\mathcal{A}$

**(success case)**



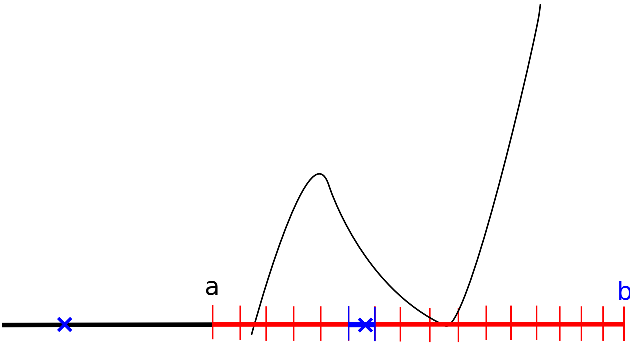
# Algorithm NEWDSC: A Trial and Error Approach

Otherwise,. . .



# Algorithm NEWDSC: A Trial and Error Approach

Otherwise, we fall back to bisection, that is, we add $((a, \text{mid}(I)), \max(4, \sqrt{N_I}))$ and $((\text{mid}(I), b), \max(4, \sqrt{N_I}))$ to $\mathcal{A}$ **(failure case).**



# Exact vs. Approximate Computation

Above description of the algorithm assumes exact arithmetic:

- applies only to rational input polynomials
- bit complexity of $\tilde{O}(n^3 L)$; amortized cost per node is $\tilde{O}(n^2 L)$

  [S. 2012]

- extension to polynomials with arbitrary real coefficients that can only be approximated
- precision demand?

**Solution:**

- computation of $v := \text{var}(f, I)$ for polynomials with approximate coefficients
- For the special cases $v = 0$ and $v = 1$, the precision demand $\rho$ is related to the absolute values of $f$ at the end points of $I$:

$$\rho = O(n + \log \|f\|_\infty + n \log \max(|a|, |b|) + \log \max(|f(a)|^{-1}, |f(b)|^{-1}))$$

# Exact vs. Approximate Computation

- comparable bound for the Newton step; precision related to the values $|f(t_i)|$
- **Idea:** Choose subdivision points, where $|f|$ becomes large; instead of $t_i$, consider approximations $\tilde{t}_i$, where $|f|$ becomes large
- Main Tool: Approximate (Multipoint) Evaluation
- Cost for processing an interval $I$ at a node can be mapped to an arbitrary root $z_i$ contained in the one-circle region of $I$:

$$\tilde{O}(n(n + \log \|f\|_\infty + n \log |z_i| + \log |f'(z_i)^{-1}|))$$

- each root is considered only a logarithmic number of times

# Results

**Main Result:** Let $f(x) = a_n x^n + \ldots + a_1 x^1 + a_0 \in \mathbb{R}[x]$ be a real, square-free polynomial of degree $n$ with $1/4 \leq a_n \leq 1$. We can determines isolating intervals for all real roots of $f$ of size less than $2^{-\kappa}$ with a number of bit operations bounded by

$$\tilde{O}(n(n^2 + n \log \mathrm{Mea}(f) + \log |\mathrm{Disc}(f)^{-1}|) + n\kappa).$$

The coefficients of $f$ must be approximated with absolute error

$$\tilde{O}(n + \log \|f\|_\infty + \max_i (n \log |z_i| + \log |f'(z_i)^{-1}|) + \kappa),$$

where $z_1$ to $z_n$ are the roots of $f$,
$\mathrm{Mea}(f) := |a_n| \cdot \prod_{i=1}^{n} \max(1, |z_i|)$ denotes the *Mahler Measure* of $f$, $\mathrm{Disc}(f)$ is the *discriminant* of $f$, and $f'$ is the derivative of $f$.

[S. and Mehlhorn 2013]

# Results

- For polynomials with integer coefficients, the bound writes as $\tilde{O}(n^3 + n^2 L + n\kappa)$
- matches complexity of the best known method due to Pan [Pan 2002]
- much simpler and more practical
- can be used to compute the real roots in a given interval only; no need to compute all complex roots
- Improvement of the bounds for isolating the roots of polynomials with algebraic coefficients

# Outlook

- Efficient implementation based on the current version of RS (together with F. Rouillier)
- Optimality of the bound?

# Outlook

- Efficient implementation based on the current version of RS (together with F. Rouillier)
- Optimality of the bound?

Thank you very much for your attention!