Fast evaluation and root finding for polynomials with floating-point coefficients

ISSAC 2023

Rémi Imbach[1] and Guillaume Moroz[1]

[1] Université de Lorraine, CNRS, Inria, LORIA

Light-year (L) in basis 2 (in meters):
100 001 100 111 000 111 101 111 110 111 001 000 100 100 011 011 000 000

Floating point representation: $m \in \mathbb{N}, \log \tau \in \mathbb{N}$

$$\mathbb{R}_{m,\tau}(L) = \underbrace{100\ 001\ 100\ 111\ 000\ 111\ 101\ 111}_{\textbf{mantissa: } m \text{ bits}} \times 10^{\overbrace{00\ 101\ 010}}$$

$\underbrace{\;}_{\textbf{exponent: } \log\tau \text{ bits}}$

Notation: $\log := \log_2$

Light-year (L) in basis 2 (in meters):
100 001 100 111 000 111 101 111 110 111 001 000 100 100 011 011 000 000

Floating point representation: $m \in \mathbb{N}, \log\tau \in \mathbb{N}$

$$\mathbb{R}_{m,\tau}(L) = \underbrace{100\ 001\ 100\ 111\ 000\ 111\ 101\ 111}_{\textbf{mantissa: } m \text{ bits}} \times 10^{\overbrace{00\ 101\ 010}^{\textbf{exponent: } \log\tau \text{ bits}}}$$

Relative error: $$\frac{|L - \mathbb{R}_{m,\tau}(L)|}{|\mathbb{R}_{m,\tau}(L)|} \leq 2^{-m}$$

Size of representation: $O(m + \log\tau)$ Cost of $\times$: $\widetilde{O}(m + \log\tau)$ bit operations

Notation: $\log := \log_2$

Light-year (L) in basis 2 (in meters):
100 001 100 111 000 111 101 111 110 111 001 000 100 100 011 011 000 000

Floating point representation: $m \in \mathbb{N}, \log\tau \in \mathbb{N}$

$$\mathbb{R}_{m,\tau}(L) = \underbrace{100\ 001\ 100\ 111\ 000\ 111\ 101\ 111}_{\textbf{mantissa: } m \text{ bits}} \times 10^{\overbrace{00\ 101\ 010}^{\textbf{exponent: } \log\tau \text{ bits}}}$$

Relative error: $$\frac{|L - \mathbb{R}_{m,\tau}(L)|}{|\mathbb{R}_{m,\tau}(L)|} \leq 2^{-m}$$

Size of representation: $O(m + \log\tau)$ Cost of $\times$: $\widetilde{O}(m + \log\tau)$ bit operations

Polynomials in $\mathbb{R}_{m,\tau}[z]$: $F(z) = f_0 + f_1 z + \ldots + f_d z^d$

Size of (dense) representation: $O(d(m + \log\tau))$

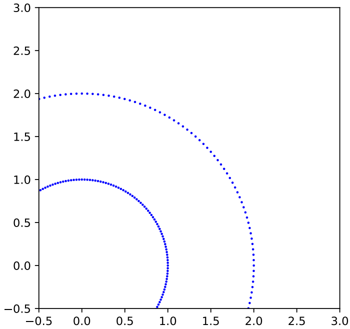
Goal: find relative m-bit approximations of all/some of the roots of $F \in \mathbb{R}_{m,\tau}[z]$ with $\widetilde{O}(d(m + \log\tau))$ bit operations?

Notation: $\log := \log_2$

Relative condition number

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

Relative condition number of a root ζ of F: *"Relatively to $|\zeta|$, measures the displacement of $|\zeta|$ under an relative infinitesimal perturbation applied to F"*

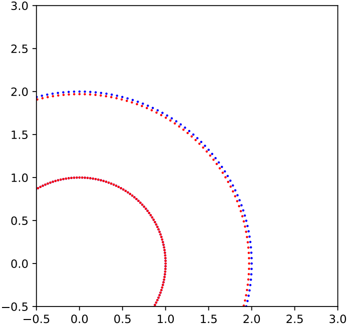


Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

Relative condition number

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$
$$f_0(1+\varepsilon_0) + f_1(1+\varepsilon_1)z + \ldots + f_d(1+\varepsilon_d)z^d$$

Relative condition number of a root ζ of F: *"Relatively to $|\zeta|$, measures the displacement of $|\zeta|$ under an relative infinitesimal perturbation applied to F"*



Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

Relative condition number

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1|z + \ldots + |f_d|z^d$$

Relative condition number of a root ζ of F: *"Relatively to $|\zeta|$, measures the displacement of $|\zeta|$ under an relative infinitesimal perturbation applied to F"*

$$cond(\zeta, F) := \frac{\widetilde{F}(|\zeta|)}{|\zeta||F'(\zeta)|}$$

Relative condition number of F:

$$cond(F) := \max_{\zeta \text{ root of } F} cond(\zeta, F)$$

Relative condition number

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1| z + \ldots + |f_d| z^d$$

Relative condition number of a root ζ of F: *"Relatively to $|\zeta|$, measures the displacement of $|\zeta|$ under an relative infinitesimal perturbation applied to F"*

$$cond(\zeta, F) := \frac{\widetilde{F}(|\zeta|)}{|\zeta||F'(\zeta)|}$$

Relative condition number of F:

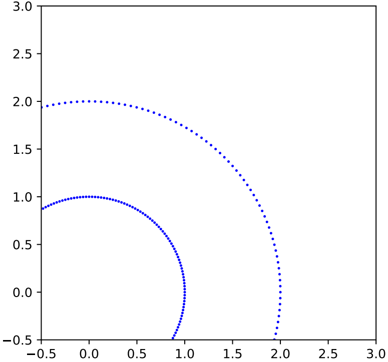
$$cond(F) := \max_{\zeta \text{ root of } F} cond(\zeta, F)$$

Result: find relative m-bit approximations of all the roots of $F \in \mathbb{R}_{m,\tau}[z]$ with $\widetilde{O}(d(m + \log \tau + \log cond(F)))$ bit operations

Piecewise polynomial approximation

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1| z + \ldots + |f_d| z^d$$

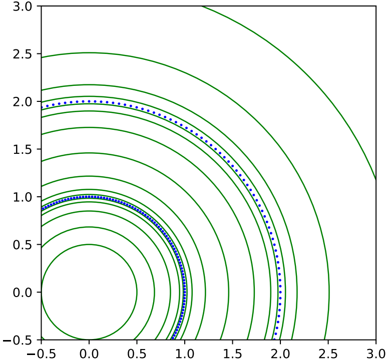


Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

Piecewise polynomial approximation

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1| z + \ldots + |f_d| z^d$$



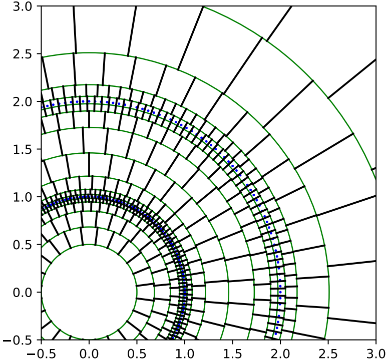
Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

N **Annuli:** $\ldots, A_n, \ldots$

Piecewise polynomial approximation

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1| z + \ldots + |f_d| z^d$$



N **Annuli:** $\ldots, A_n, \ldots$

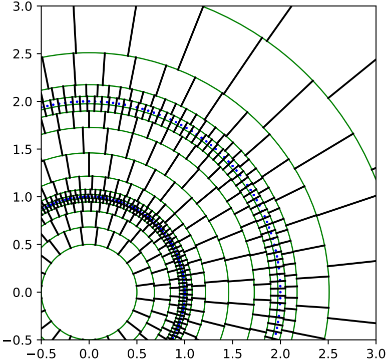
K **Sectors:** $\ldots, S_{n,k}, \ldots$

Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

Piecewise polynomial approximation

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1| z + \ldots + |f_d| z^d$$



Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

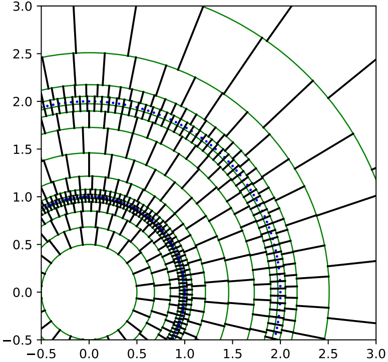
N **Annuli:** $\ldots, A_n, \ldots$
K **Sectors:** $\ldots, S_{n,k}, \ldots$
K **Approximations:**
In sector $S_{n,k}$, we approximate F with:

$$z^{\ell_n} G_{n,k}(z) = z^{\ell_n} \left(g_0 + \ldots + g_{d_n} z^{d_n}\right)$$

Piecewise polynomial approximation

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1| z + \ldots + |f_d| z^d$$



Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

N **Annuli:** $\ldots, A_n, \ldots$
K **Sectors:** $\ldots, S_{n,k}, \ldots$
K **Approximations:**
In sector $S_{n,k}$, we approximate F with:

$$z^{\ell_n} G_{n,k}(z) = z^{\ell_n} \left(g_0 + \ldots + g_{d_n} z^{d_n}\right)$$

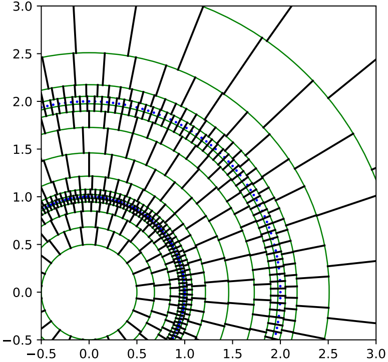
satisfying: $\forall z \in S_{n,k}$,

$$|F(z) - z^{\ell_n} G_{n,k}(z)| \leq 2^{-m} \widetilde{F}(|z|)$$

Piecewise polynomial approximation

$$F(z) = f_0 + f_1 z + \ldots + f_d z^d$$

$$\widetilde{F}(z) = |f_0| + |f_1| z + \ldots + |f_d| z^d$$



Example: $F(z) = (z^{200} - 2^{200})(z^{200} - 1)$

N **Annuli:** $\ldots, A_n, \ldots$
K **Sectors:** $\ldots, S_{n,k}, \ldots$
K **Approximations:**
In sector $S_{n,k}$, we approximate F with:

$$z^{\ell_n} G_{n,k}(z) = z^{\ell_n} \left(g_0 + \ldots + g_{d_n} z^{d_n} \right)$$

satisfying: $\forall z \in S_{n,k}$,

$$|F(z) - z^{\ell_n} G_{n,k}(z)| \leq 2^{-m} \widetilde{F}(|z|)$$

and: $d_n \in O(m + \log d)$

Contributions

Given: $m \in \mathbb{N}$, $\log\tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

We compute:

Contributions

Given: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

We compute:

- a piecewise polynomial approximation satisfying:

$$\forall S_{n,k} \subset \mathbb{C},\ \forall z \in S_{n,k},\ |F(z) - z^{\ell_n} G_{n,k}(z)| \leq 2^{-m} \widetilde{F}(|z|)$$

in $\widetilde{O}(d(m + \log \tau))$ bit operations

Contributions

Given: $m \in \mathbb{N},\ \log \tau \in \mathbb{N},\ F \in \mathbb{R}_{m,\tau}[z]$ of degree d

We compute:

- a piecewise polynomial approximation satisfying:

$$\forall S_{n,k} \subset \mathbb{C},\ \forall z \in S_{n,k},\ |F(z) - z^{\ell_n} G_{n,k}(z)| \leq 2^{-m}\widetilde{F}(|z|)$$

  in $\widetilde{O}(d(m + \log \tau))$ bit operations

- relative m-bits approximations of all the roots ζ of F s.t. $cond(\zeta, F) \leq 2^m$ in $\widetilde{O}(d(m + \log \tau))$ bit operations

Contributions

Given: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

We compute:

- a piecewise polynomial approximation satisfying:

$$\forall S_{n,k} \subset \mathbb{C},\ \forall z \in S_{n,k},\ |F(z) - z^{\ell_n} G_{n,k}(z)| \leq 2^{-m}\widetilde{F}(|z|)$$

  in $\widetilde{O}(d(m + \log \tau))$ bit operations

- relative m-bits approximations of all the roots ζ of F s.t. $cond(\zeta, F) \leq 2^m$ in $\widetilde{O}(d(m + \log \tau))$ bit operations

- if $\kappa = cond(F)$, relative m-bits approximations of all the roots of F in $\widetilde{O}(d(m + \log \tau + \log \kappa))$ bit operations

Root finding: best state-of-the-art approaches

Given: $F \in \mathbb{R}[z]$ of degree d, $b \in \mathbb{N}_{>}$

Root approximation problem:

Compute: b-bit approximations of all the roots ζ of F

Theoretical record: Pan's algorithm (when $b \geq d \log d$):

$\rightarrow \widetilde{O}(d^2 b)$ bit operations for root approximation

Root finding: best state-of-the-art approaches

Given: $F \in \mathbb{R}[z]$ of degree d, $b \in \mathbb{N}_{>}$

Root approximation problem:
Compute: b-bit approximations of all the roots ζ of F

Approximate factorization problem:
Compute: d linear factors H_i s.t. $\|F - \prod_i H_i\|_1 \leq 2^{-b}\|F\|_1$

Theoretical record: Pan's algorithm (when $b \geq d \log d$):
$\rightarrow \widetilde{O}(db)$ bit operations for approximate fact.[Pan2002]
$\rightarrow \widetilde{O}(d^2 b)$ bit operations for root approximation

[Pan2002] Victor Y. Pan. *Univariate Polynomials: Nearly Optimal Algorithms for Numerical Factorization and Root-finding*
Journal of Symbolic Computation, 2002

Root finding: best state-of-the-art approaches

Given: $F \in \mathbb{R}[z]$ of degree d, $b \in \mathbb{N}_>$

Root approximation problem:
Compute: b-bit approximations of all the roots ζ of F

Theoretical record: Pan's algorithm (when $b \geq d \log d$):

$\rightarrow \widetilde{O}(d^2 b)$ bit operations for root approximation

User's choice: MPSolve [BR2014] (simultaneous Newton-like iterations)
$\rightarrow$ each iteration in $O(d^2)$ arithmetic operations
$\rightarrow$ no known bound on the number of iterations

[BR2014] Dario A. Bini and Leonardo Robol. *Solving secular and polynomial equations: A multiprecision algorithm*
Journal of Computational and Applied Mathematics, 2014

Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

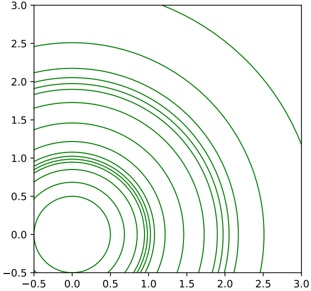
Output:

Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

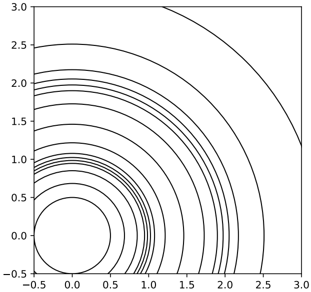
F $\xrightarrow{\textbf{Step 1}}$



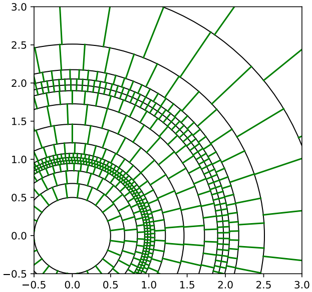
Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$
K polynomials $\ldots, G_{n,k}, \ldots$



Step 2 $\longrightarrow$



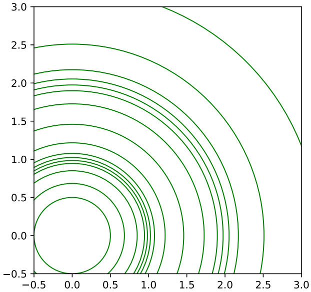
Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

Step 1:

$F \xrightarrow{\textbf{Step 1}}$

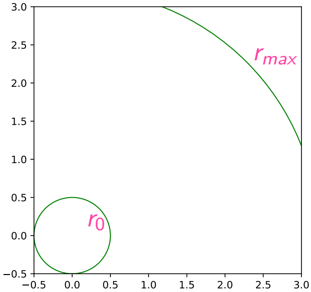


Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

Step 1: choose reals $r_0 < r_{max}$

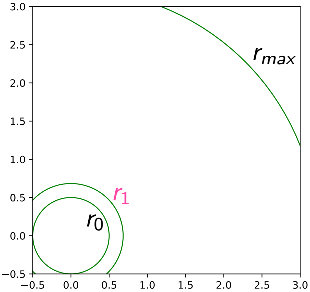


Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$

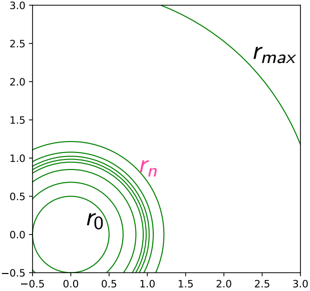


Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$

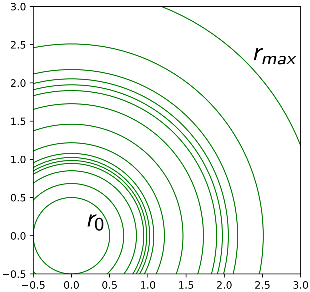


Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$
together with, for each n:
$\ell_n \in \mathbb{N}_>$, $F_n \in \mathbb{R}_{m,\tau}[z]$ of degree δ_n



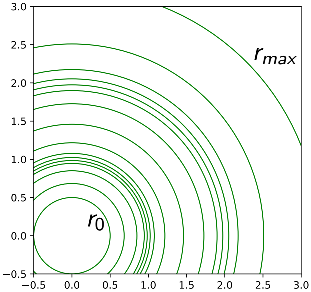
Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$
together with, for each n:
$\ell_n \in \mathbb{N}_>$, $F_n \in \mathbb{R}_{m,\tau}[z]$ of degree δ_n
satisfying:

$$r_n \leq |z| \leq r_{n+1} \Rightarrow |F(z) - z^{\ell_n} F_n(z)| \leq d2^{-m} \widetilde{F}(|z|)$$



Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

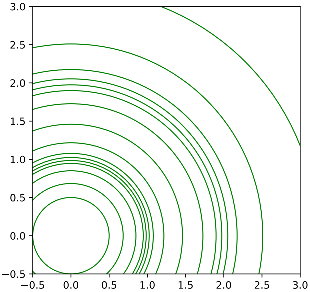
Output: N annulii $\ldots, A_n, \ldots$
N polynomials $\ldots, F_n, \ldots$

Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$
together with, for each n:
$\ell_n \in \mathbb{N}_>$, $F_n \in \mathbb{R}_{m,\tau}[z]$ of degree δ_n
satisfying:

$$r_n \leq |z| \leq r_{n+1} \Rightarrow |F(z) - z^{\ell_n} F_n(z)| \leq d2^{-m}\widetilde{F}(|z|)$$

and

$$r_{n+1} = 2^{m/\delta_n} r_n$$



Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$

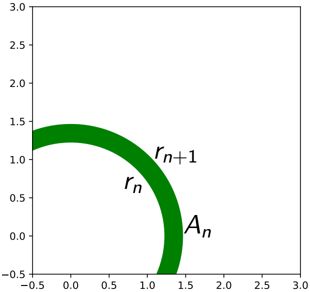
Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$
together with, for each n:
$\ell_n \in \mathbb{N}_>$, $F_n \in \mathbb{R}_{m,\tau}[z]$ of degree δ_n
satisfying:

$$r_n \leq |z| \leq r_{n+1} \Rightarrow |F(z) - z^{\ell_n} F_n(z)| \leq d2^{-m}\widetilde{F}(|z|)$$

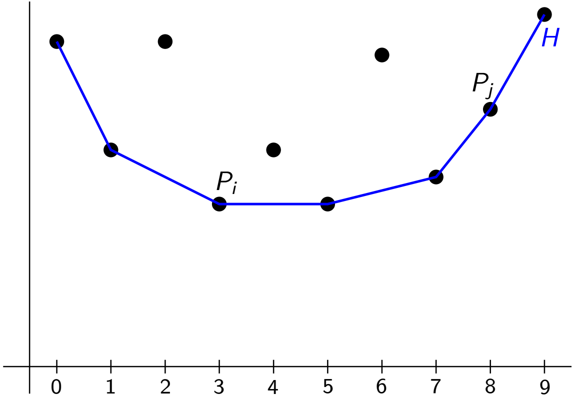
and

$$r_{n+1} = 2^{m/\delta_n} r_n$$

let $A_n = D(0, r_{n+1}) \setminus D(0, r_n)$



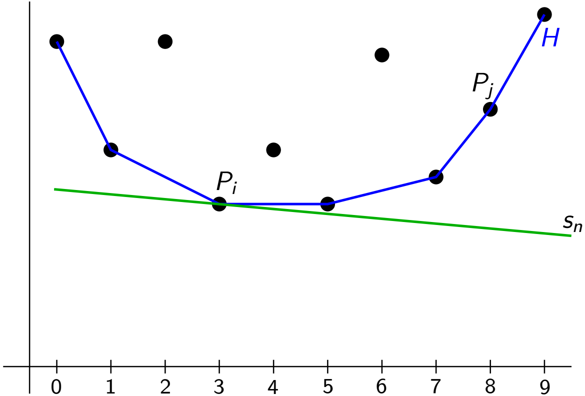
Newton Polygon



Notations:

$P_i := (i, -\log|f_i|)$

Newton Polygon



Notations:

$P_i := (i, -\log|f_i|)$

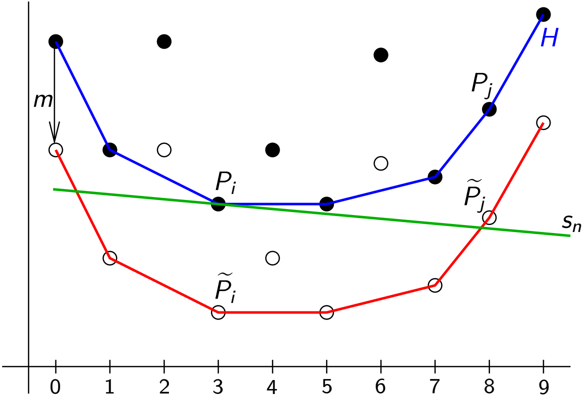
s_n: line of slope r_n tangent to H

P_j above $s_n \Leftrightarrow |f_j||r_n|^j < |f_i||r_n|^i$

thus $|f_i||r_n|^i = \max_k |f_k||r_n|^k$

$\leq \widetilde{F}(|r_n|)$

Newton Polygon



Notations:

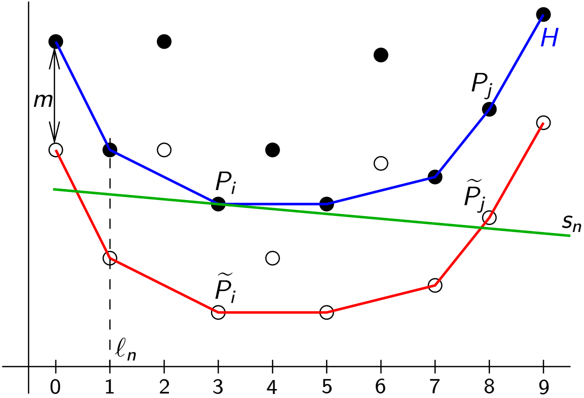
$P_i := (i, -\log|f_i|)$
$\widetilde{P}_i := (i, -\log|f_i| - m)$

s_n: line of slope r_n tangent to H

P_j above $s_n \Leftrightarrow |f_j||r_n|^j < |f_i||r_n|^i$
thus $|f_i||r_n|^i = \max_k |f_k||r_n|^k$
$\leq \widetilde{F}(|r_n|)$

$\widetilde{P}_j$ above $s_n \Leftrightarrow |f_j||r_n|^j < 2^{-m}\widetilde{F}(|r_n|)$

Newton Polygon



Notations:

$P_i := (i, -\log|f_i|)$

$\widetilde{P}_i := (i, -\log|f_i| - m)$

s_n: line of slope r_n tangent to H

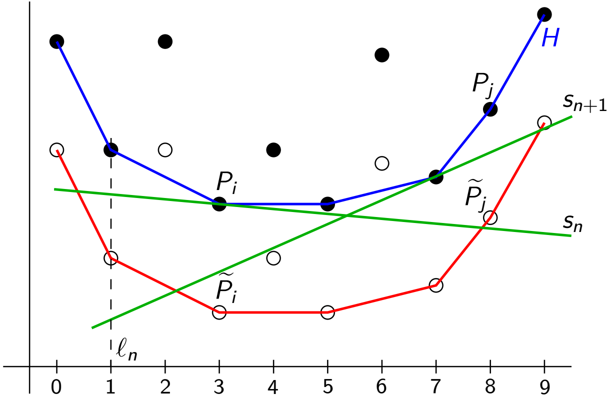
P_j above $s_n \Leftrightarrow |f_j||r_n|^j < |f_i||r_n|^i$
thus $|f_i||r_n|^i = \max_k |f_k||r_n|^k$

$\widetilde{P}_j$ above $s_n \Leftrightarrow |f_j||r_n|^j < 2^{-m}\widetilde{F}(|r_n|)$

let ℓ_n be the leftmost index s.t. $\widetilde{P_{\ell_n}}$ is below s_n:

$$\forall r_n < r,\ \forall j < \ell_n,\ \ |f_j||r|^j < 2^{-m}\widetilde{F}(|r|)$$

Approximation on an annulus $A_n = D(0, r_{n+1}) \setminus D(0, r_n)$



Notations:

$P_i := (i, -\log|f_i|)$
$\widetilde{P}_i := (i, -\log|f_i| - m)$

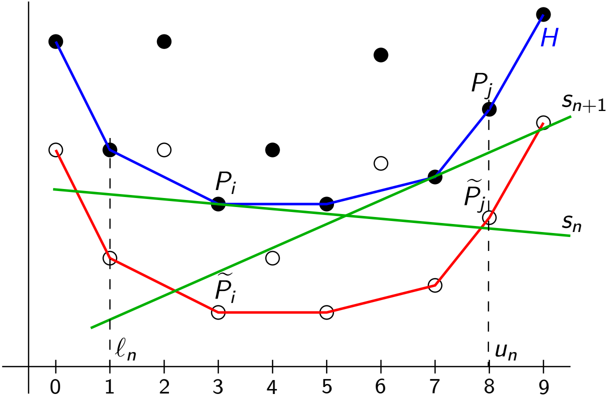
s_n: line of slope r_n tangent to H

s_{n+1}: line of slope r_{n+1} tangent to H

let ℓ_n be the leftmost index s.t. $\widetilde{P_{\ell_n}}$ is below s_n:

$$\forall r_n < r,\ \forall j < \ell_n,\ \ |f_j||r|^j < 2^{-m}\widetilde{F}(|r|)$$

Approximation on an annulus $A_n = D(0, r_{n+1}) \setminus D(0, r_n)$



Notations:

$P_i := (i, -\log|f_i|)$

$\widetilde{P}_i := (i, -\log|f_i| - m)$

s_n: line of slope r_n tangent to H

s_{n+1}: line of slope r_{n+1} tangent to H

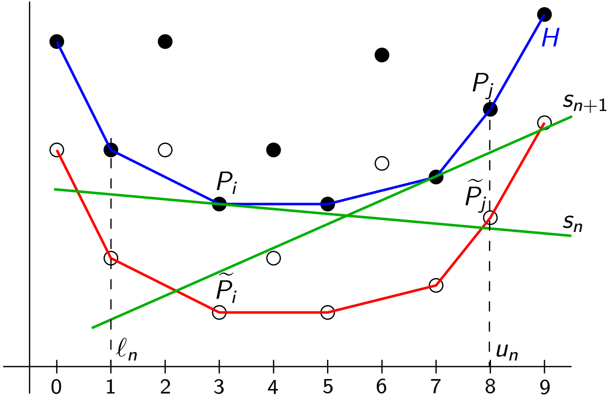
let u_n be the rightmost index s.t. $\widetilde{P_{u_n}}$ is below s_{n+1}:

$$\forall r \leq r_{n+1}, \forall j > u_n, \quad |f_j||r|^j < 2^{-m}\widetilde{F}(|r|)$$

let ℓ_n be the leftmost index s.t. $\widetilde{P_{\ell_n}}$ is below s_n:

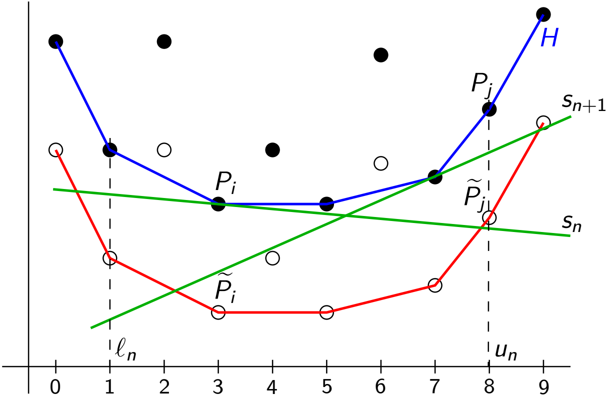
$$\forall r_n < r, \ \forall j < \ell_n, \quad |f_j||r|^j < 2^{-m}\widetilde{F}(|r|)$$

Approximation on an annulus $A_n = D(0, r_{n+1}) \setminus D(0, r_n)$



let $F_n(z) := f_{\ell_n} + \ldots + f_{u_n} z^{u_n - \ell_n}$ and $\delta_n := u_n - \ell_n$,

Approximation on an annulus $A_n = D(0, r_{n+1}) \setminus D(0, r_n)$



let $F_n(z) := f_{\ell_n} + \ldots + f_{u_n} z^{u_n - \ell_n}$ and $\delta_n := u_n - \ell_n$, then

$$\forall r_n \leq |z| \leq r_{n+1}, \quad |F(z) - z^{\ell_n} F_n(z)| \leq d 2^{-m} \widetilde{F}(|z|)$$

Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

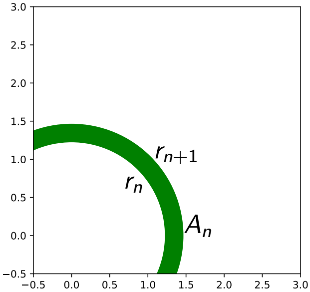
Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$

Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$
together with, for each n:
$\ell_n \in \mathbb{N}_>$, $F_n \in \mathbb{R}_{m,\tau}[z]$ of degree δ_n
satisfying:

$$r_n \leq |z| \leq r_{n+1} \Rightarrow |F(z) - z^{\ell_n} F_n(z)| \leq d 2^{-m} \widetilde{F}(|z|)$$

and

$$r_{n+1} = 2^{m/\delta_n} r_n$$



Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log\tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$

Lemma 1: Bounding the number of terms

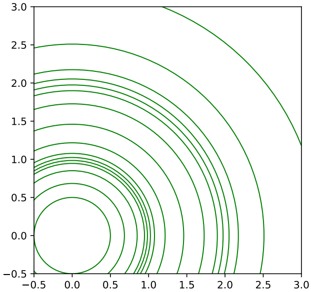
If, for all n, $r_{n+1} = 2^{m/\delta_n} r_n$ then $\sum \delta_n \in O(d)$

Step 1: compute $r_0 < r_1 < \ldots < r_n < \ldots < r_N = r_{max}$
together with, for each n:
$\ell_n \in \mathbb{N}_>$, $F_n \in \mathbb{R}_{m,\tau}[z]$ of degree δ_n
satisfying:

$$r_n \leq |z| \leq r_{n+1} \Rightarrow |F(z) - z^{\ell_n} F_n(z)| \leq d2^{-m}\widetilde{F}(|z|)$$

and

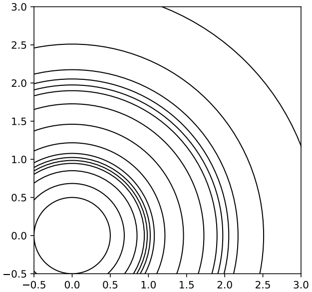
$$r_{n+1} = 2^{m/\delta_n} r_n$$



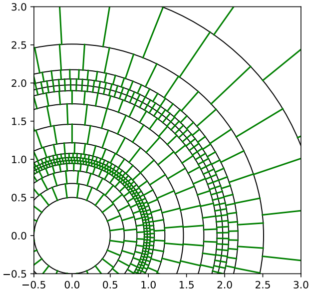
Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\dots, A_n, \dots$
N polynomials $\dots, F_n, \dots$
K sectors $\dots, S_{n,0}, \dots, S_{n,k}, \dots, S_{n,K_n}, \dots$
K polynomials $\dots, G_{n,k}, \dots$



Step 2 $\longrightarrow$

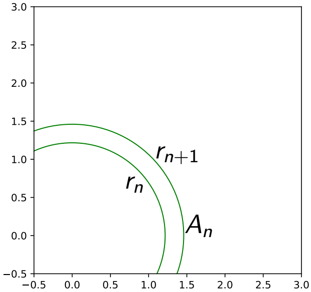


Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$

Step 2: for $n = 0, \ldots, N-1$:

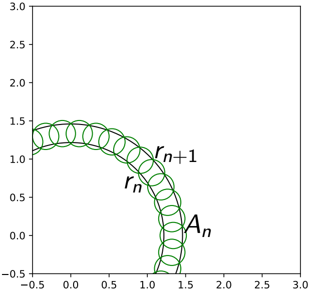


Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$

Step 2: for $n = 0, \ldots, N-1$:
2.1 cover A_n with K_n disks $D(\gamma_{n,k}, \rho_n)$



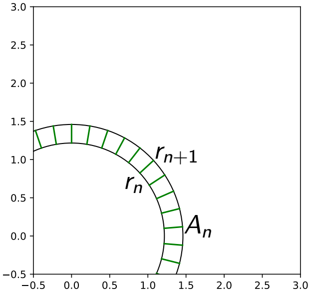
Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log\tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$

Step 2: for $n = 0, \ldots, N-1$:
2.1 cover A_n with K_n disks $D(\gamma_{n,k}, \rho_n)$
$\longrightarrow K_n$ sectors $S_{n,0}, \ldots, S_{n,K_n}$

Lemma 2: If $r_{n+1} = 2^{m/\delta_n} r_n$ then $K_n \in O(\delta_n/m)$



Piecewise polynomial approximation algorithm: overview

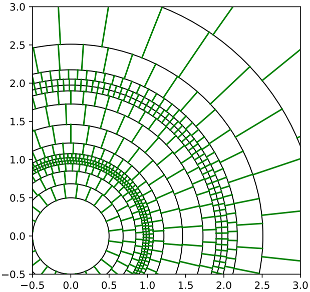
Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$

Lemma 1: Bounding the number of terms

If, for all n, $r_{n+1} = 2^{m/\delta_n} r_n$ then $\sum \delta_n \in O(d)$

Step 2: for $n = 0, \ldots, N-1$:
2.1 cover A_n with K_n disks $D(\gamma_{n,k}, \rho_n)$
$\longrightarrow K_n$ sectors $S_{n,0}, \ldots, S_{n,K_n}$



Lemma 2: If $r_{n+1} = 2^{m/\delta_n} r_n$ then $K_n \in O(\delta_n/m)$

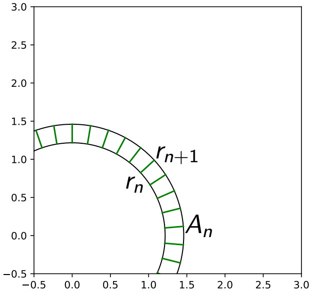
Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$
K polynomials $\ldots, G_{n,k}, \ldots$ of degree $\in O(m)$

Step 2: for $n = 0, \ldots, N-1$:
2.1 cover A_n with K_n disks $D(\gamma_{n,k}, \rho_n)$

2.2 compute the first $4m$ coeffs of
$F_n(\gamma_{n,k} + \rho_n z)$, for $k = 0, \ldots, K_n - 1$



Piecewise polynomial approximation algorithm: overview

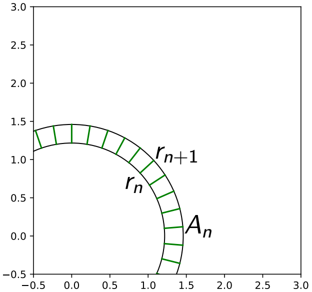
Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$
K polynomials $\ldots, G_{n,k}, \ldots$ of degree $\in O(m)$ s.t.

$$z \in S_{n,k} \Rightarrow |F(z) - z^{\ell_n} G_{n,k}(z)| \leq d2^{-m}\widetilde{F}(|z|)$$

Step 2: for $n = 0, \ldots, N-1$:
2.1 cover A_n with K_n disks $D(\gamma_{n,k}, \rho_n)$

2.2 compute the first $4m$ coeffs of
$F_n(\gamma_{n,k} + \rho_n z)$, for $k = 0, \ldots, K_n - 1$
$\longrightarrow$ K_n polynomials $G_{n,k}$



Piecewise polynomial approximation algorithm: overview

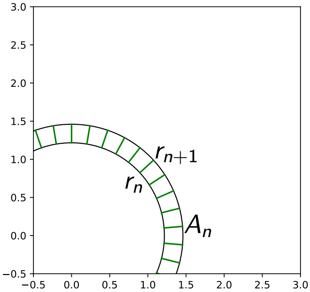
Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\dots, A_n, \dots$ and N integers $\dots, \ell_n, \dots$
N polynomials $\dots, F_n, \dots$ of degrees $\dots, \delta_n, \dots$
K sectors $\dots, S_{n,0}, \dots, S_{n,k}, \dots, S_{n,K_n}, \dots$ with $K \in O(d/m)$
K polynomials $\dots, G_{n,k}, \dots$ of degree $\in O(m)$ s.t.

$$z \in S_{n,k} \Rightarrow |F(z) - z^{\ell_n} G_{n,k}(z)| \leq d2^{-m}\widetilde{F}(|z|)$$

Step 2: for $n = 0, \dots, N-1$:
2.1 cover A_n with K_n disks $D(\gamma_{n,k}, \rho_n)$

2.2 compute the first $4m$ coeffs of
$F_n(\gamma_{n,k} + \rho_n z)$, for $k = 0, \dots, K_n - 1$



Lemma 3: using $4m$ FFTs of length K_n, step **2.2** is addressed with $\widetilde{O}(\delta_n(m + \log\tau))$ bit ops.

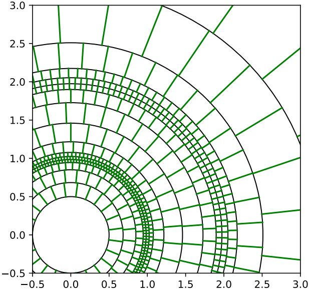
Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$
K polynomials $\ldots, G_{n,k}, \ldots$ of degree $\in O(m)$ s.t.

$$z \in S_{n,k} \Rightarrow |F(z) - z^{\ell_n} G_{n,k}(z)| \leq d2^{-m}\widetilde{F}(|z|)$$

Cost: $\widetilde{O}(d(m + \log \tau))$ bit operations



Lemma 3: using $4m$ FFTs of length K_n, step **2.2** is addressed with $\widetilde{O}(\delta_n(m + \log \tau))$ bit ops.

Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m)$
K polynomials $\ldots, G_{n,k}, \ldots$ of degree $\in O(m)$ s.t.

$$z \in S_{n,k} \Rightarrow |F(z) - z^{\ell_n} G_{n,k}(z)| \leq d 2^{-m} \widetilde{F}(|z|)$$

Cost: $\widetilde{O}(d(m + \log \tau))$ bit operations

Piecewise polynomial approximation algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

$\leftarrow$ do it for $m' \simeq m + \log d$

Output: N annulii $\ldots, A_n, \ldots$ and N integers $\ldots, \ell_n, \ldots$
N polynomials $\ldots, F_n, \ldots$ of degrees $\ldots, \delta_n, \ldots$
K sectors $\ldots, S_{n,0}, \ldots, S_{n,k}, \ldots, S_{n,K_n}, \ldots$ with $K \in O(d/m')$
K polynomials $\ldots, G_{n,k}, \ldots$ of degree $\in O(m')$ s.t.

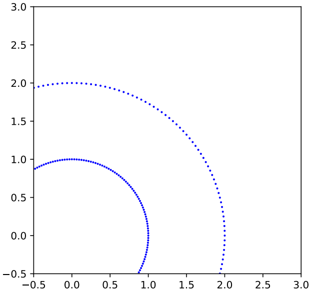
$$z \in S_{n,k} \Rightarrow |F(z) - z^{\ell_n} G_{n,k}(z)| \leq 2^{-m} \widetilde{F}(|z|)$$

Cost: $\widetilde{O}(d(m + \log \tau))$ bit operations

Root-finding algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

$$F \longrightarrow$$



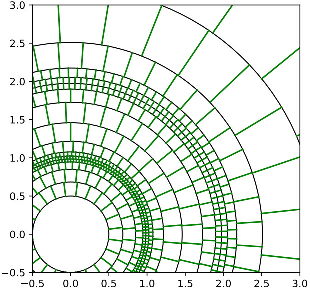
Root-finding algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Step 0: take $m' \in O(m + \log d)$

Step 1: compute a P.P.A F_{pw} s.t. $|F(z) - F_{pw}(z)| \leq 2^{-m}\widetilde{F}(|z|)$ with:
K sectors $\ldots, S_{n,k}, \ldots$ with $K \in O(d/m')$
K polynomials $\ldots, G_{n,k}, \ldots$ with degree in $O(m')$

$$F \longrightarrow$$



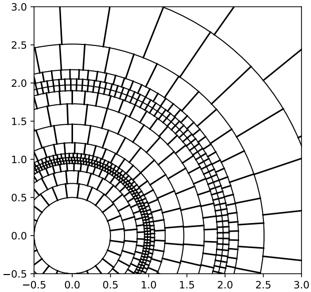
Root-finding algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

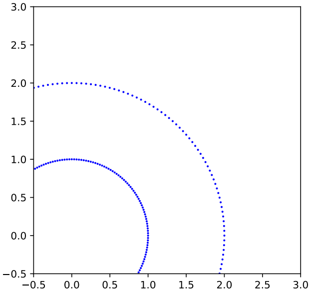
Step 0: take $m' \in O(m + \log d)$

Step 1: compute a P.P.A F_{pw} s.t. $|F(z) - F_{pw}(z)| \leq 2^{-m}\widetilde{F}(|z|)$ with:
K sectors $\ldots, S_{n,k}, \ldots$ with $K \in O(d/m')$
K polynomials $\ldots, G_{n,k}, \ldots$ with degree in $O(m')$

Step 2: find relative m-bit approximations of roots of F



$\longrightarrow$



Root finding: best state-of-the-art approaches

Given: $F \in \mathbb{R}[z]$ of degree d, $b \in \mathbb{N}_{>}$

Root approximation problem:
Compute: b-bit approximations of all the roots ζ of F

Approximate factorization problem:
Compute: d linear factors H_i s.t. $\|F - \prod_i H_i\|_1 \leq 2^{-b}\|F\|_1$

Theoretical record: Pan's algorithm (when $b \geq d \log d$):
$\rightarrow \widetilde{O}(db)$ bit operations for approximate fact.[Pan2002]

User's choice: MPSolve [BR2014] (simultaneous Newton-like iterations)

Root-finding algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Step 0: take $m' \in O(m + \log d)$

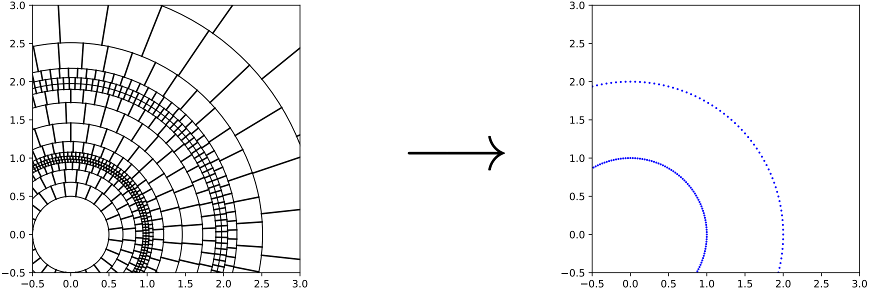
Step 1: compute a P.P.A F_{pw} s.t. $|F(z) - F_{pw}(z)| \leq 2^{-m}\widetilde{F}(|z|)$ with:

K sectors $\ldots, S_{n,k}, \ldots$ with $K \in O(d/m')$

K polynomials $\ldots, G_{n,k}, \ldots$ with degree in $O(m')$

Step 2: find relative m-bit approximations of roots of F

$\leftarrow$ can be done in $\widetilde{O}(d(m + \log \tau))$ bit operations



Root-finding algorithm: overview

Input: $m \in \mathbb{N}$, $\log \tau \in \mathbb{N}$, $F \in \mathbb{R}_{m,\tau}[z]$ of degree d

Step 0: take $m' \in O(m + \log d)$

Step 1: compute a P.P.A F_{pw} s.t. $|F(z) - F_{pw}(z)| \leq 2^{-m}\widetilde{F}(|z|)$ with:
K sectors $\ldots, S_{n,k}, \ldots$ with $K \in O(d/m')$
K polynomials $\ldots, G_{n,k}, \ldots$ with degree in $O(m')$

Step 2: find relative m-bit approximations of roots of F
$\leftarrow$ can be done in $\widetilde{O}(d(m + \log \tau))$ bit operations

Theorem:

For each root ζ of F with $cond(\zeta, F) \leq 2^m$, our algorithm output a disc $D(\dot{\zeta}, r_{\dot{\zeta}})$ s.t.:

- $r_{\dot{\zeta}} \leq 2^{-m}|\dot{\zeta}|$
- the unique root of F in $D(\dot{\zeta}, r_{\dot{\zeta}})$ is ζ

Root finding: best state-of-the-art approaches

Given: $F \in \mathbb{R}[z]$ of degree d, $b \in \mathbb{N}_{>}$

Root approximation problem:
Compute: b-bit approximations of all the roots ζ of F

Approximate factorization problem:
Compute: d linear factors H_i s.t. $\|F - \prod_i H_i\|_1 \leq 2^{-b}\|F\|_1$

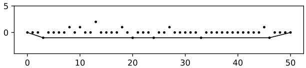
Theoretical record: Pan's algorithm (when $b \geq d \log d$):
$\rightarrow \widetilde{O}(db)$ bit operations for approximate fact.[Pan2002]
$\leftarrow$ **for complexity result**

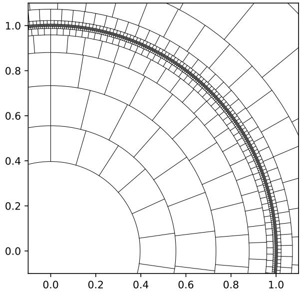
User's choice: MPSolve [BR2014] (simultaneous Newton-like iterations)
$\leftarrow$ **for prototypal implementation**

Three families of well-conditioned random polynomials

c_j are i.i.d random Gaussian variables with mean 0

hyperbolic: $\sum c_j z^j$



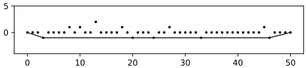


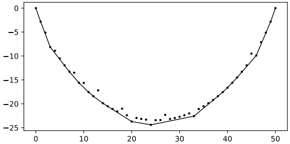
Three families of well-conditioned random polynomials

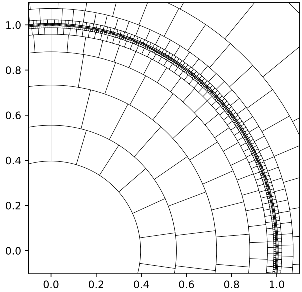
c_j are i.i.d random Gaussian variables with mean 0

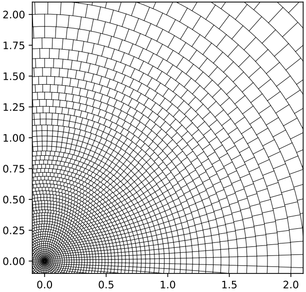
hyperbolic: $\sum c_j z^j$

elliptic: $\sum c_j \sqrt{\binom{d}{j}} z^j$





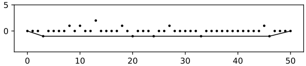


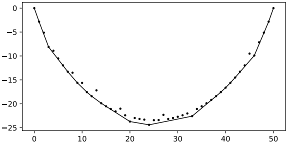


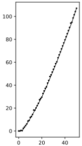
Three families of well-conditioned random polynomials

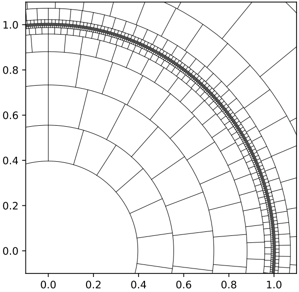
c_j are i.i.d random Gaussian variables with mean 0

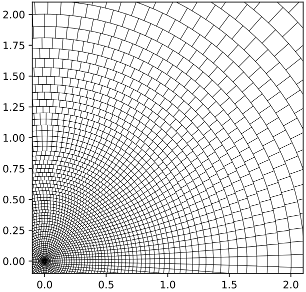
hyperbolic: $\sum c_j z^j$ **elliptic:** $\sum c_j \sqrt{\binom{d}{j}} z^j$ **flat:** $\sum c_j \sqrt{\frac{1}{j!}} z^j$

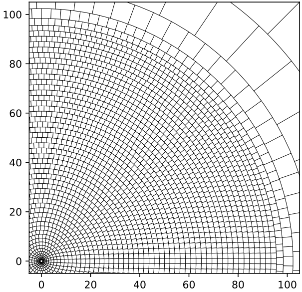




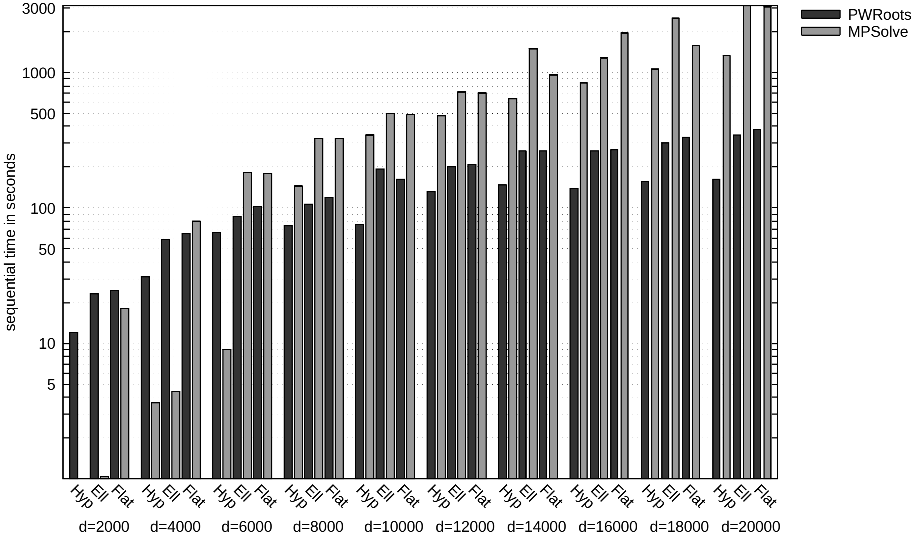








Benchmarks: root-finding with input $m = 30$



Conclusion

- multipoint evaluation for the same bit complexity
- new data structure generalizing floating point representation to functions
- first release of `PWRoots` soon available[1]:

  `C` library based on `Arb` with `Python` interface

Main perspective:

- systems of two bivariate polynomials

Thank you!

[1] `https://gitlab.inria.fr/gamble/pwpoly`