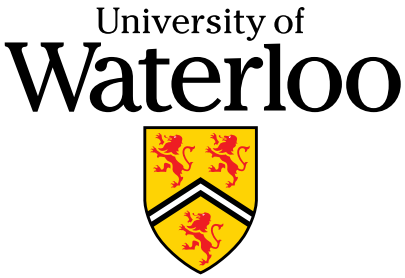
Interpolation of Shifted-Lacunary Polynomials

Mark Giesbrecht Daniel Roche



Symbolic Computation Group
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

MACIS 2007
Paris, France, December 5–7

A Simple, Small Example

Suppose we have a way to evaluate the following unknown polynomial at any chosen point:

$$g(x) = (x-3)^{107} - 485(x-3)^{54}$$

Question

How can we use interpolation to determine $g(x)$?

- Complexity should be proportional to the size of $g(x)$ as written above, in particular log of the degree and number of terms

Dense Interpolation

Definition (Dense Representation)

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,$$

where $n = \deg(f)$ and $a_0, a_1, \ldots, a_n \in \mathsf{R}$

Newton (1711), Waring (1779).

For $g(x) = (x-3)^{107} - 485(x-3)^{54}$, we will have

$$\begin{aligned} g(x) = {} & x^{107} - 321x^{106} + 51039x^{105} - 5359095x^{104} + \cdots \\ & + 4020099274965907985483758515231179267430359014481937 3x \\ & - 1127130637840908780976768693419860197828458989848152 \end{aligned}$$

This is way too big! (twice exponential in the desired size)

Lacunary (i.e. Sparse) Interpolation

Definition (Lacunary Representation)

$$f(x) = b_1 x^{d_1} + b_2 x^{d_2} + \cdots + b_s x^{e_s},$$

where $d_1 < d_2 < \cdots < d_s = n$ and $b_1, \ldots, b_s \in \mathsf{R} \setminus \{0\}$

Baron de Prony (1795), Ben-Or & Tiwari (1988)

- Complexity: Polynomial in s, $\log n$, and $\log \|f\|_\infty$

Example

$$\begin{aligned} g(x) &= (x-3)^{107} - 485(x-3)^{54} \\ g(x+3) &= x^{107} - 485x^{54} \end{aligned}$$

So we can interpolate $g(x+3)$.

Shifted-Lacunary Interpolation

Definition (Shifted-Lacunary Representation)

$$f(x) = c_1(x-\alpha)^{e_1} + c_2(x-\alpha)^{e_2} + \cdots + c_t(x-\alpha)^{e_t},$$

where $e_1 < \cdots < e_t = n$ and t is minimal for any α

- α is called the *sparsest shift* of $f(x)$
- First compute α,
  then interpolate $f(x+\alpha)$.

Question

How to determine the sparsest shifted power basis
of an unknown polynomial (i.e. α)?

Computing the Sparsest Shift

- Borodin & Tiwari (1991)
  Compute sparsest shift from evaluation points (open)
- Grigoriev & Karpinski (1993)
  Compute sparsest shift from a black-box function.
  State need for complexity *not* polynomial in n
- Lakshman & Saunders (1996)
  Compute sparsest shift from dense representation
- Giesbrecht, Kaltofen, Lee (2003)
  Current best results (deterministic & probabilistic)

Uniqueness and Rationality of Sparsest Shift

Theorem (Lakshman & Saunders (1996))

If the degree is at least twice the sparsity,
then the sparsest shift is unique and rational.

Example

$$g(x) = (x-3)^{107} - 485(x-3)^{54}$$

$\Rightarrow$ 3 is the only shift with $\leq$ two terms

Condition not satisfied means polynomial is dense.

Black Box Model

Problem: Arbitrary evauations can be vary large.

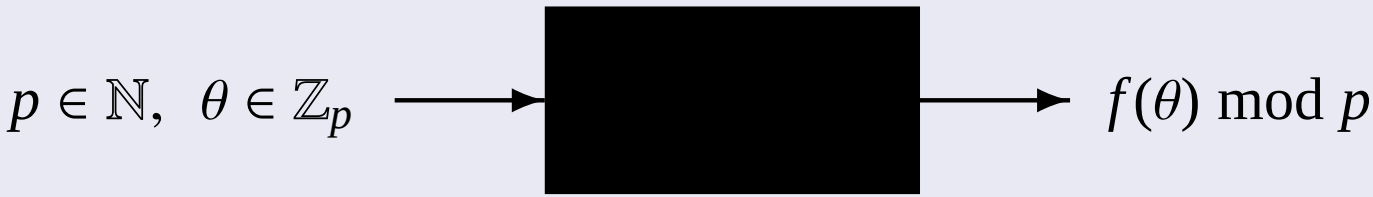
Example

$$g(x) = (x-3)^{107} - 485(x-3)^{54}$$

$$g(1) = -162259276829222100374855109050368$$

To control evaluation size:

The "Modular Black-Box"



Modular Reduction

For $\alpha \equiv \alpha_p \bmod p$ and a prime p:

Definition (Modular-reduced polynomial)

$$f_p(x) = (c_1 \bmod p)(x - \alpha_p)^{e_1 \bmod (p-1)} + \cdots + (c_t \bmod p)(x - \alpha_p)^{e_t \bmod (p-1)}$$

▸ Original definition of f, c_i, e_i, α

- $f(\theta) \bmod p = f_p(\theta \bmod p)$ whenever $\theta \equiv \alpha \bmod p$ (Fermat's Little Theorem)
- α_p is at least a t-sparse shift of $f_p(x)$

Outline of Algorithm

Input: A bound B on the bit length of the lacunary-shifted representation

1. Choose a prime p with $p \in O(B^{O(1)})$
2. Evaluate $f(1), \ldots, f(p-1) \bmod p$ to attempt to interpolate $f_p(x)$
3. Use a dense sparsest shift method to compute α_p
4. Repeat Steps 1–3 enough times to recover α

Example

Unknown Polynomial in $\mathbb{Q}[x]$

$$g(x) = (x-3)^{107} - 485(x-3)^{54}$$

1. Choose a prime p with $p \in O(B^{O(1)})$
2. Evaluate $f(1), \ldots, f(p-1) \bmod p$ to attempt to interpolate $f_p(x)$
3. Use a dense sparsest shift method to compute α_p
4. Repeat Steps 1–3 enough times to recover α

Step 1

$$p = 11$$

Example

Unknown Polynomial in $\mathbb{Q}[x]$

$$g(x) = (x-3)^{107} - 485(x-3)^{54}$$

1. Choose a prime p with $p \in O(B^{O(1)})$
2. Evaluate $f(1), \ldots, f(p-1) \bmod p$ to attempt to interpolate $f_p(x)$
3. Use a dense sparsest shift method to compute α_p
4. Repeat Steps 1–3 enough times to recover α

Step 2

$$g(1), \ldots, g(p-1) \bmod p = 10, 9, 0, 0, 2, 5, 2, 5, 10, 3$$

$$g_p(x) = x^7 + x^6 + 2x^5 + 9x^3 + 2x^2 + 8x + 9$$

Example

Unknown Polynomial in $\mathbb{Q}[x]$

$$g(x) = (x-3)^{107} - 485(x-3)^{54}$$

1. Choose a prime p with $p \in O(B^{O(1)})$
2. Evaluate $f(1), \ldots, f(p-1) \bmod p$ to attempt to interpolate $f_p(x)$
3. Use a dense sparsest shift method to compute α_p
4. Repeat Steps 1–3 enough times to recover α

Step 3

$$g_p(x) \equiv (x-3)^7 + 10(x-3)^4 \mod p$$

$$\alpha_p = 3$$

Example

Unknown Polynomial in $\mathbb{Q}[x]$

$$g(x) = (x-3)^{107} - 485(x-3)^{54}$$

1. Choose a prime p with $p \in O(B^{O(1)})$
2. Evaluate $f(1), \ldots, f(p-1) \bmod p$ to attempt to interpolate $f_p(x)$
3. Use a dense sparsest shift method to compute α_p
4. Repeat Steps 1–3 enough times to recover α

Step 4

$$\alpha_{11} = 3, \quad \alpha_{13} = 3, \quad \alpha_{17} = 3, \quad \ldots$$

$$\alpha = 3$$

Types of Failures

Failure categories:

- $f_p(x)$ is not computed correctly
- The sparsest shift of $f_p(x)$ is not α_p

Second condition is equivalent to $\deg f_p(x) < 2t - 1$

Next we develop sufficient conditions on p to avoid failure.

Exponents Vanish

Example ($f_p(x)$ computed incorrectly)

$$f(x) = 10(x-1)^{12} + 8(x-1)^3$$

$$p = 7$$

$$f_7(x) = 3(x-1)^0 + (x-1)^3$$

$$f(1) \bmod 7 = 0 \not\equiv 3 = f_7(1)$$

Condition: $(p-1) \nmid \max\{1, e_1\} \cdot e_2 \cdot e_3 \cdots e_t$

Test: Constant coeff. of computed $f_p(x)$ equals constant coeff. of $f(x)$ modulo p

Exponents Too Small

Example (Sparsest shift of $f_p(x)$ is not α_p)

$$f(x) = -4(x-2)^{145} + 14(x-2)^{26} + 3$$

$$p = 13$$

$$\begin{aligned} f_{13}(x) &= 9(x-2)^1 + (x-2)^2 + 3 \\ &\equiv (x-4)^2 + 12 \end{aligned}$$

Condition: $(p-1) \nmid e_t(e_t-1)(e_t-2)\cdots(e_t-(2t-2))$

Test: $\deg f_p(x) \geq 2B - 1 \geq 2t - 1$

Exponents Collide

Example (Sparsest shift of $f_p(x)$ is not α_p)

$$f(x) = 4(x-1)^{59} + 2(x-1)^{21} + 7(x-1)^{19} + 20$$

$$p = 11$$

$$\begin{aligned} f_{11}(x) &= 4(x-1)^9 + 2(x-1)^1 + 7(x-1)^9 + 9 \\ &= 2(x-1) + 9 \\ &\equiv 2(x-2) \end{aligned}$$

Condition: $(p-1) \nmid (e_t - e_1)(e_t - e_2)\cdots(e_t - e_{t-1})$

Test: $\deg f_p(x) \geq 2B - 1 \geq 2t - 1$

Coefficients Vanish

Example (Sparsest shift of $f_p(x)$ is not α_p)

$$f(x) = 69(x-5)^{42} - 12(x-5)^{23} + 4$$

$$p = 23$$

$$\begin{aligned} f_{23}(x) &= 0(x-5)^{20} + 11(x-5)^{1} + 4 \\ &= 11(x-5) + 4 \\ &\equiv 11(x-13) \end{aligned}$$

Condition: $p \nmid c_t$

Test: $\deg f_p(x) \geq 2B - 1 \geq 2t - 1$

Choosing Good Primes

Primes satisfying these conditions will be “good”:

- $p \nmid c_t$
- $(p-1) \nmid C$,

where $c_t < 2^B$ and $C < 2^{4B^2}$

Approach: Choose $p = qk + 1$ for unique primes q

Choosing Primes Deterministically

1. Let $q_1, \ldots, q_k$ be the first k primes, for $k \in O(B^2 \log B)$.
2. Let p_i be the smallest prime s.t. $p_i \equiv 1 \bmod q_i$.
3. Set $\mathcal{P} = \{p_1, p_2, \ldots, p_k\}$.

Example (Choosing a p_i)

$$q_7 = 17$$

$$p_7 = 103 \equiv 1 \bmod 17$$

- We know $p_i \in O(q_i^{5.5})$
  (Linnik's Theorem, Heath-Brown '92)
- Probably $p_i < q_i^2$

Complexity of Deterministic Algorithm

Step 3 (computing sparsest shift of $f_p(x)$) dominates.

Deterministic complexity: $O(B^{78}(\log B)^{24} \log\log B)$

Factor	Source
5.5	Bounds on Linnik's constant
2	Need $(p-1) \nmid C$ and $\log C \in O(B^2)$
7	Deterministic sparsest shift algorithm

Improving from the Deterministic Complexity

Finding primes p probabilistically is much easier:

1. Choose a random prime $q \in O(B^2 \log B)$.
2. Choose random k's less than q until a prime $p = qk + 1$ is found.

Using randomization and Rousselet (1988),
exponent in complexity is reduced from 78 to 17.

Interpolation

Once α is known, we can construct a modular black box for evaluating $f(x + \alpha)$.

Then use lacunary interpolation along the lines of Kaltofen, Lakshman, & Wiley (1990) and Avendaño, Krick, & Pacetti (2006).

Conclusions

- Shifted-lacunary interpolation can be performed in polynomial time, for rational polynomials given by a modular black box.
- How to apply these techniques to other problems on lacunary polynomials?
- What about domains other than $\mathbb{Q}[x]$?
- What about multivariate rational polynomials?

Shifted-Lacunary Representation

Definition (Shifted-Lacunary Representation)

$$f(x) = c_1(x-\alpha)^{e_1} + c_2(x-\alpha)^{e_2} + \cdots + c_t(x-\alpha)^{e_t},$$

where $e_1 < \cdots < e_t = n$ and t is minimal for any α

Back to Modular Reduction