Raising Small Polynomials with Real Coefficients to Exponential Power

Ivan Adrian Koswara[1], Gleb Pogudin[2],
Svetlana Selivanova[1], Martin Ziegler[1]

[1]KAIST, South Korea

[2]École Polytechnique, France

International Conference on Computability and Complexity in Analysis,
11 September 2020

Talk outline

1. Motivation of problem
2. Problem statement
3. The problem is "stable"
4. Complexity classes in real setting
5. The problem is in $\mathbb{R}\#\mathrm{P}$
6. For certain cases, the problem is in $\mathbb{R}\mathrm{FP}$

Problem overview: polynomial powering

Given:

- polynomial $p \in \mathbb{R}[x]$ (real coefficients)
- natural number E as exponent

Compute coefficients of p^E.

Example: $(\frac{1}{2}x + \frac{1}{2})^E$, then coefficient of x^I is $\binom{E}{I}2^{-E}$.

Motivation: approximating PDEs

Approximating solutions of PDEs

$\Longrightarrow$ Solving a system of linear recurrences

$\Longrightarrow$ Computing matrix power with large matrix and large exponent

Motivation: approximating PDEs

Approximating solutions of PDEs

$\Longrightarrow$ Solving a system of linear recurrences

$\Longrightarrow$ Computing matrix power with large matrix and large exponent

- The matrix is structured and can be turned into a polynomial

$$\begin{pmatrix} a_0 & a_1 & & a_{-1} \\ a_{-1} & a_0 & a_1 & \\ & a_{-1} & a_0 & a_1 \\ a_1 & & a_{-1} & a_0 \end{pmatrix} \text{ into } p(x) = a_{-1}x^{-1} + a_0 + a_1 x$$

- Entries are real numbers

Problem statement

Assumptions:

- fix $p \in \mathbb{R}[x]$ satisfying $\sum |p_i| \leq 1$
- p_i approximated to 2^{-k}
- input is natural numbers E, I, n

Compute $p^E[x^I]$ approximated to error 2^{-n}.

Problem statement

Assumptions:

- fix $p \in \mathbb{R}[x]$ satisfying $\sum |p_i| \leq 1$
- p_i approximated to 2^{-k}
- input is natural numbers E, I, n

Compute $p^E[x^I]$ approximated to error 2^{-n}.

Also,

- complexity is measured in $n + \log E$
- k will depend on n, E

Motivation of problem statement?

Fix $p \in \mathbb{R}[x]$ satisfying $\sum |p_i| \leq 1$, coefficients approximated to error 2^{-k}.
Given naturals E, I, n, compute $p^E[x^I]$ approximated to error 2^{-n}.

Complexity is measured in $n + \log E$. k will depend on n, E.

- n, E are the main parameters
- We want E to be "exponential-size": usually $O(2^n)$
- Naive method works in time poly$(n + E)$

Motivation of problem statement?

Fix $p \in \mathbb{R}[x]$ satisfying $\sum |p_i| \leq 1$, coefficients approximated to error 2^{-k}.
Given naturals E, I, n, compute $p^E[x^I]$ approximated to error 2^{-n}.

Complexity is measured in $n + \log E$. k will depend on n, E.

- We can't have full representation of the numbers, so work with approximations
- Compute output to arbitrary accuracy: output to error 2^{-n}
- Approximating input should suffice: input to error 2^{-k}

Motivation of problem statement?

Fix $p \in \mathbb{R}[x]$ satisfying $\sum |p_i| \leq 1$, coefficients approximated to error 2^{-k}.
Given naturals E, I, n, compute $p^E[x^I]$ approximated to error 2^{-n}.

Complexity is measured in $n + \log E$. k will depend on n, E.

- Input cannot be too long, so we must prove the problem is stable
- Output cannot be too long, so we must add conditions on p

Is the problem polynomially stable?

Fix $p \in \mathbb{R}[x]$ satisfying $\sum |p_i| \leq 1$, coefficients approximated to error 2^{-k}.
Given naturals E, I, n, compute $p^E[x^I]$ approximated to error 2^{-n}.

Complexity is measured in $n + \log E$. k will depend on n, E.

Basic arithmetic operations on numbers of b-bits takes time $\text{poly}(b)$
$\implies$ Manipulating input numbers takes time $\text{poly}(k)$

We want k to be polynomial in $n + \log E$

Will it allow us to compute the output?

Yes, the problem is polynomially stable

If p has degree d and coefficients are approximated to error δ,
p^2 has degree $2d$ and each coefficient can be approximated to error $4d\delta$.

Exponentiation-by-squaring on p ($p \to p^2 \to p^4 \to p^8 \to \ldots \to p^E$)

$\implies$ Output error $2^{-k+\text{poly}(\log E)}$

$\implies$ Must be smaller than 2^{-n} so $k = n + \text{poly}(\log E)$ works

Real complexity classes: $\mathbb{R}$FP

$$\mathrm{FP} \subseteq \mathrm{FPSPACE} \subseteq \mathrm{FEXPTIME}$$

Real complexity classes: $\mathbb{R}$FP

$$\text{FP} \subseteq \text{FPSPACE} \subseteq \text{FEXPTIME}$$

FP

$F : \mathbb{N}^d \to \mathbb{N}$ has algorithm $\mathcal{A}$: on $\vec{x}$, outputs $\mathcal{A}(\vec{x})$ in time $\text{poly}(\ell(\vec{x}))$, where

$$F(\vec{x}) = \mathcal{A}(\vec{x})$$

Real complexity classes: $\mathbb{R}$FP

$$\text{FP} \subseteq \text{FPSPACE} \subseteq \text{FEXPTIME}$$

FP

$F : \mathbb{N}^d \to \mathbb{N}$ has algorithm $\mathcal{A}$: on $\vec{x}$, outputs $\mathcal{A}(\vec{x})$ in time $\text{poly}(\ell(\vec{x}))$, where

$$F(\vec{x}) = \mathcal{A}(\vec{x})$$

$\mathbb{R}$FP

$F : \mathbb{R}^d \to \mathbb{R}$ has polynomials k, m, algorithm $\mathcal{A}$: on $(\vec{a}, n)$ where $\|\vec{x} - \vec{a}/2^{k(n)}\| \leq 2^{-k(n)}$, outputs $\mathcal{A}(\vec{a}, n)$ in time $\text{poly}(n + \ell(\vec{a}))$, where

$$\left| F(\vec{x}) - \frac{\mathcal{A}(\vec{a}, n)}{2^{m(n)}} \right| \leq 2^{-n}$$

Real complexity classes: $\mathbb{R}$FP and $\mathbb{R}\#$P

$$\mathrm{FP} \subseteq \#\mathrm{P} \subseteq \mathrm{FPSPACE} \subseteq \mathrm{FEXPTIME}$$

$\mathbb{R}$FP

$F : \mathbb{R}^d \to \mathbb{R}$ has polynomials k, m, algorithm $\mathcal{A}$: on $(\vec{a}, n)$ where $\|\vec{x} - \vec{a}/2^{k(n)}\| \leq 2^{-k(n)}$, outputs $\mathcal{A}(\vec{a}, n)$ in time $\mathrm{poly}(n + \ell(\vec{a}))$, where

$$\left| F(\vec{x}) - \frac{\mathcal{A}(\vec{a}, n)}{2^{m(n)}} \right| \leq 2^{-n}$$

Real complexity classes: $\mathbb{R}$FP and $\mathbb{R}\#$P

$$\text{FP} \subseteq \#\text{P} \subseteq \text{FPSPACE} \subseteq \text{FEXPTIME}$$

$\mathbb{R}$FP

$F : \mathbb{R}^d \to \mathbb{R}$ has polynomials k, m, algorithm $\mathcal{A}$: on $(\vec{a}, n)$ where $\|\vec{x} - \vec{a}/2^{k(n)}\| \leq 2^{-k(n)}$, outputs $\mathcal{A}(\vec{a}, n)$ in time poly$(n + \ell(\vec{a}))$, where

$$\left| F(\vec{x}) - \frac{\mathcal{A}(\vec{a}, n)}{2^{m(n)}} \right| \leq 2^{-n}$$

$\mathbb{R}\#$P

$F : \mathbb{R}^d \to \mathbb{R}$ has polynomials k, m, p and algorithm $\mathcal{A}$: on $(\vec{a}, w, n)$ with $\ell(w) = p(n + \ell(\vec{a}))$ and $\|\vec{x} - \vec{a}/2^{k(n)}\| \leq 2^{-k(n)}$, outputs $\mathcal{A}(\vec{a}, w, n)$ in time poly$(n + \ell(\vec{a}))$, where

$$\left| F(\vec{x}) - \sum_{w} \frac{\mathcal{A}(\vec{a}, w, n)}{2^{m(n)}} \right| \leq 2^{-n}$$

Polynomial powering in $\mathbb{R}\#\mathrm{P}$: facts

Cauchy's differentiation formula for polynomials

Given $q \in \mathbb{R}[z]$, we can compute the coefficient of z^j as

$$q[z^j] = \frac{1}{j!}\frac{d^j}{dz^j} q(z)\big|_{z=0} = \frac{1}{2\pi i} \cdot \oint \frac{q(z)}{z^{j+1}}\, dz$$

where each $\oint$ is any closed path with winding number 1 around the origin.

Ker-I Ko, Complexity Theory of Real Functions (1991)

If $f : [0,1] \to \mathbb{R}$ is bounded and poly-time computable, then computing $\int_0^1 f$ is in $\mathbb{R}\#\mathrm{P}$.

Polynomial powering in $\mathbb{R}\#\mathrm{P}$: approach

Given $q \in \mathbb{R}[z]$,

$$q[z^j] = \frac{1}{2\pi i} \cdot \oint \frac{q(z)}{z^{j+1}}\, dz$$

1. Apply Cauchy's differentiation formula on $q := p^E$, $j = I$.
2. Parametrize the integral using polar coordinates.
3. The integrand is poly-time computable and bounded.
4. By Ko, the integral is computable in $\mathbb{R}\#\mathrm{P}$.

Generalization: Same idea works when q is multivariate, with complex matrix coefficients

Restricting the problem

Fix $p(x) = ax + (1 - a)$ where $a \leq 1/2$, coefficients approximated to error 2^{-k}. Fix real c. Given naturals $E, I, n = c \log E$, compute $p^E[x^I]$ approximated to error 2^{-n}.

Complexity is measured in $n + \log E$. k will depend on n, E.

Restricting the problem

Fix $p(x) = ax + (1 - a)$ where $a \leq 1/2$, coefficients approximated to error 2^{-k}. Fix real c. Given naturals $E, I, n = c \log E$, compute $p^E[x^I]$ approximated to error 2^{-n}.

Complexity is measured in $n + \log E$. k will depend on n, E.

Note $E = 2^{n/c}$ so $n + \log E = O(n)$.

The coefficient of z^I in p^E is

$$\binom{E}{I} a^I (1 - a)^{E-I}$$

Powering binomials in $\mathbb{R}$FP: Stirling approximation

Stirling series

For any natural t, there exist fixed rationals $a_1, \ldots, a_t, b$ such that

$$m! = \sqrt{2\pi m}\left(\frac{m}{e}\right)^m \left(1 + \frac{a_1}{m} + \ldots + \frac{a_t}{m^t} + R_t(m)\right)$$

where $|R_t(m)| \leq b/m^{t+1}$.

Powering binomials in $\mathbb{R}$FP: Stirling approximation

Stirling series

For any natural t, there exist fixed rationals $a_1, \ldots, a_t, b$ such that

$$m! = \sqrt{2\pi m}\left(\frac{m}{e}\right)^m \left(1 + \frac{a_1}{m} + \ldots + \frac{a_t}{m^t} + R_t(m)\right)$$

where $|R_t(m)| \leq b/m^{t+1}$.

We can approximate $m!$ with relative error b/m^{t+1}

$\implies$ Approximate $\binom{E}{I} a^I (1-a)^{E-I}$ with relative error constant$/I^{t+1}$

Powering binomials in $\mathbb{R}$FP: Stirling approximation

Stirling series

For any natural t, there exist fixed rationals $a_1, \ldots, a_t, b$ such that

$$m! = \sqrt{2\pi m}\left(\frac{m}{e}\right)^m \left(1 + \frac{a_1}{m} + \ldots + \frac{a_t}{m^t} + R_t(m)\right)$$

where $|R_t(m)| \leq b/m^{t+1}$.

We can approximate $m!$ with relative error b/m^{t+1}
$\implies$ Approximate $\binom{E}{I} a^I (1-a)^{E-I}$ with relative error constant$/I^{t+1}$

If $I/E >$ constant, we can approximate $\binom{E}{I} a^I (1-a)^{E-I}$ with relative error

$$O\left(2^{-n\cdot(c+1)/c}\right)$$

which decays faster than 2^{-n}.

Powering binomials in $\mathbb{R}$FP: interpreting as probability

A coin comes up heads with probability a, approximate the probability it comes out heads I times out of E.

Idea: if I/E is far from a, the probability is very low that 0 is an approximation

Powering binomials in $\mathbb{R}$FP: interpreting as probability

A coin comes up heads with probability a, approximate the probability it comes out heads I times out of E.

Idea: if I/E is far from a, the probability is very low that 0 is an approximation

Chernoff bound

Let $X_1, \ldots, X_n$ are independent random Bernoulli variables, $X = \sum X_i$, and $\mu = \mathbb{E}[X]$. Let $0 \leq \delta \leq 1$. Then

$$\Pr\left(|X - \mu| \geq \delta\mu\right) \leq 2e^{-\delta^2\mu/3}$$

Powering binomials in $\mathbb{R}$FP: interpreting as probability

A coin comes up heads with probability a, approximate the probability it comes out heads I times out of E.

Idea: if I/E is far from a, the probability is very low that 0 is an approximation

Chernoff bound

Let $X_1, \ldots, X_n$ are independent random Bernoulli variables, $X = \sum X_i$, and $\mu = \mathbb{E}[X]$. Let $0 \leq \delta \leq 1$. Then

$$\Pr\left(|X - \mu| \geq \delta\mu\right) \leq 2e^{-\delta^2\mu/3}$$

If I/E is not close to a, we can approximate $\binom{E}{I} a^I (1-a)^{E-I}$ as 0.

Powering binomials in $\mathbb{R}$FP: approach

Fix positive reals a, c with $a \leq 1/2$. Given naturals E, I. Approximate

$$\binom{E}{I} a^I (1-a)^{E-I}$$

to error 2^{-n} where $n = c \log E$.

1. Assume E is large enough.
2. By Chernoff bound, if I/E is not close to a, return 0.
3. Otherwise $I/E >$ fraction of a, which is constant.
4. By Stirling series we can approximate with relative error $< 2^{-n}$.
5. Since the result is < 1, this is absolute error $< 2^{-n}$.

Each step is poly-time $\implies$ The problem is in $\mathbb{R}$FP

Powering binomials in $\mathbb{R}$FP: approach

Fix positive reals a, c with $a \leq 1/2$. Given naturals E, I. Approximate

$$\binom{E}{I} a^I (1-a)^{E-I}$$

to error 2^{-n} where $n = c \log E$.

1. Assume E is large enough.
2. By Chernoff bound, if I/E is not close to a, return 0.
3. Otherwise $I/E >$ fraction of a, which is constant.
4. By Stirling series we can approximate with relative error $< 2^{-n}$.
5. Since the result is < 1, this is absolute error $< 2^{-n}$.

Each step is poly-time $\implies$ The problem is in $\mathbb{R}$FP

Generalizations: p can be multivariate (but still degree 1), coefficients can be reals as long as $\|p\| \leq 1$

Summary

- Polynomial with real coefficients raised to large power
- The problem is stable
- In general the problem is in $\mathbb{R}\#\mathrm{P}$
  1. Use Cauchy's differentiation formula to extract coefficient
  2. Integrand is bounded and poly-time computable, so by Ko, computing integral is in $\mathbb{R}\#\mathrm{P}$
- For degree 1 and dependent accuracy, the problem is in $\mathbb{R}\mathrm{FP}$
  1. If I/E is far from a, by Chernoff bound, approximate by 0
  2. If I/E is close to a, use Stirling series
- Various generalizations
- Future work: Problem $\mathbb{R}\#\mathrm{P}$-complete? Get rid of assumptions?