Efficient Polynomial Factoring Modulo p^4

Nitin Saxena (*CSE@IIT Kanpur, India*)

with Ashish Dwivedi & Rajat Mittal

2019, ISSAC, Beihang University, China

Contents

- The problem
- Importance
- Prior work
- Our work
- Proof ideas
- Conclusion

The problem

- Input: Integral polynomial $f(x)$ and prime-power p^k (in bits).
- Output: Nontrivial factor $g(x)$ of $f(x)$ mod p^k (*if one exists*).
- We want a *practical* algorithm.
  - With time-complexity poly(deg(f), k.log p) .
- Brute-force: Search for all $g(x)$.
  - Takes time >> $p^{k.\deg(f)}$.
- OPEN: Can something better be done?
- *Obstacle:* Number of factors $g(x)$ could be really **huge**!
  - *Loss of unique* factorization when $k>1$.

Mod p^2

Eg 1. $\mathbf{x^2 + px}$

Eg 2. $\mathbf{x^2 + p}$

Contents

- The problem
- Importance
- Prior work
- Our work
- Proof ideas
- Conclusion

Importance

- Factoring is a fundamental problem in computation.
  - Special case of *root finding* is equally important.
- *Factoring mod p* : Is the most important case.
  - Case of rationals, number fields, finite fields rely on it.
- *Factoring mod p^k* : Is the natural next case to tackle.
  - Case of p-adic fields,
  - Galois rings, formal power series rely on it.
- *Factoring mod n* : Strongly related to the above & integer factoring!

Contents

- The problem
- Importance
- Prior work
- Our work
- Proof ideas
- Conclusion

Prior work

- For large k, (von zur Gathen, Hartlieb '96; Cheng,Labahn '01) gave a fast algorithm.
  - $k >$ valuation of discriminant of f.
- Related case is that of p-adic factoring. It was solved by (Chistov '87; Cantor,Gordon '00).
- Small k case is *notorious*. Only $k=2$ solved by (Sălăgean '05).
  - $k=3$ studied by (Sircana '17), but algorithm question left *open*.
- Hard to connect factors with "roots" mod p^k, for *small* $k>1$.
- Foundational case $k=1$, has celebrated algorithms via roots; eg. (Berlekamp '67; Cantor,Zassenhaus '81) .

Prior work

- On the other hand, root finding has *practical* solutions known.
  - (Berthomieu,Lecerf,Quintin '13) could find, and count, roots mod p^k.
  - (Cheng,Gao,Rojas,Wan '18) count roots in deterministic $\mathrm{poly}(2^k, \ldots)$ time.
  - (Dwivedi,Mittal,S. '19) count roots in deterministic poly-time.

  - This allows computing Igusa's local zeta function of univariate polynomials.

  - other applications in p-adic computation, coding theory, etc.

Contents

- The problem
- Importance
- Prior work
- Our work
- Proof ideas
- Conclusion

Our work

- We factor $f(x)$ mod p^4 in randomized poly(deg(f), log p) time.
  - Or, output that $f(x)$ mod p^4 is irreducible.
- Such methods were unknown before.
- Rough idea:

  We connect factors of $f(X)$ mod p^4 to "roots" in the ring $\mathbb{Z}[x]/\langle p^4, x^l \rangle$.

Contents

- The problem
- Importance
- Prior work
- Our work
- Proof ideas
- Conclusion

Proof idea-- factoring to *root*-finding

- Hensel lifting reduces $f(x)$ to $\varphi(x)^e$ mod p^4,
  - where φ is an *irreducible* mod p.
- So, find factors $h(x) =: \varphi^a - py$, for $a \leq e/2$.
  - y is the *unknown*.
- Inspires the *cofactor* calculation :
  - $g(x) := f / (\varphi^a - py) = (f/\varphi^a).(1 - py/\varphi^a)^{-1}$
  - $= [f/\varphi^{4a}].[\varphi^{3a} + (py).\varphi^{2a} + (py)^2.\varphi^a + (py)^3] \bmod p^4$
  - $=: [E(y) / \varphi^{4a}] \bmod p^4$.
- $\Rightarrow$ Need *roots* of $E(y)$ in the ring $\mathbb{Z}[x]/\langle p^4, \varphi^{4a}\rangle$.

Proof idea-- root-finding mod *principal* ideal

- $E(y) := f.[\varphi^{3a} + (py).\varphi^{2a} + (py)^2.\varphi^{a} + (py)^3]$ over $\mathbb{Z}[x]/\langle p^4, \varphi^{4a}\rangle$.

- <u>Idea</u>: Work in characteristic p.
  - Write $y =: y_0 + py_1 + p^2y_2 + p^3y_3$.
  - y_i's in $F_p[x]/\langle\varphi^{4a}\rangle$.
  - y_2 , y_3 play *no role* mod $\langle p^4, \varphi^{4a}\rangle$.

- Also, $E(y) \in \langle p^2, \varphi^{4a}\rangle$.

- First, solve $E(y_0 + py_1) / p^2 \in \langle p, \varphi^{4a}\rangle$.

- Next, solve $E(y_0 + py_1) / p^3 \in \langle p, \varphi^{4a}\rangle$.

Proof idea-- finding those roots

- $E(y_0 + py_1) / p^2 \bmod \langle p, \varphi^{4a} \rangle$ is free of y_1 .
  - Solve y_0 using (Berthomieu,Lecerf,Quintin '13).
- $E' := E(y_0 + py_1) / p^3 \bmod \langle p, \varphi^{4a} \rangle$ is linear in y_1 .
  - Next, solve y_1 modifying (Berthomieu,Lecerf,Quintin '13).
- In the details, we use the fact that:
  - coefficient of y_1 in E' is linear in y_0 .

Contents

- The problem
- Importance
- Prior work
- Our work
- Proof ideas
- Conclusion

At the end …

- Mod p^3 & Mod p^4, we give the first randomized poly-time algorithms for factoring.
- Open : For higher k, randomized subexp-time algorithm ?
- The new methods hold promise ...

Thank you!