Polynomial GCDs and Factorization

Tutorial

Jürgen Gerhard

Director of Research
Maplesoft
Waterloo ON, Canada

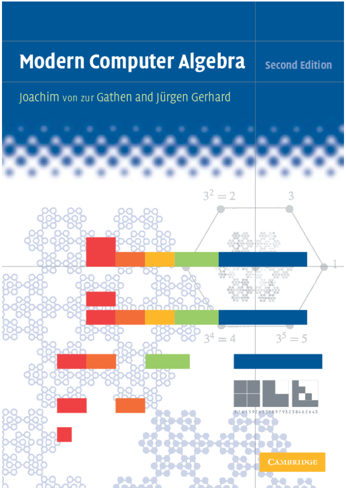
Summation, Integration and Special Functions
in Quantum Field Theory, 2012

Outline

Outline

1. Introduction
   - Unique factorization domains
   - Cost models
2. Univariate GCDs
3. Univariate factorization over finite fields
4. Univariate factorization over the integers
5. Two or more variables

Commercial



Look forward to the 3rd edition!

Examples

$$
\begin{aligned}
x^3 - x &= x \cdot (x^2 - 1) = x \cdot (x+1) \cdot (x-1) \\
x^4 - 1 &= (x^2 + 1) \cdot (x^2 - 1) \\
&= (x^2 + 1) \cdot (x+1) \cdot (x-1) \quad \text{over } \mathbb{Q} \\
&= (x+i) \cdot (x-i) \cdot (x+1) \cdot (x-1) \quad \text{over } \mathbb{C}
\end{aligned}
$$

Common divisors of $x^4 - 1$ and $x^3 - x$:

$$1,\ x+1,\ x-1,\ (x+1)(x-1) = x^2 - 1 = \gcd(x^4 - 1, x^3 - x)$$

Definitions

$(R, +, 0, \cdot, 1)$ commutative ring (often write ab for $a \cdot b = b \cdot a$)

- R *integral domain*: $a \cdot b = a \cdot c \implies a = 0$ or $b = c$ (cancellation law)
- $R^* = \{u \in R : \exists v \in R$ with $u \cdot v = 1\}$ (group of *units*)
  Notation: $v = u^{-1}$
- $a \in R \setminus R^*$ *irreducible*: $a = bc \implies b \in R^*$ or $c \in R^*$
- $a \mid b \; :\Longleftrightarrow \; \exists c$ with $ac = b$
- c *greatest common divisor* (GCD) of a and b:
  $c \mid a$ and $c \mid b$ and $\forall d \; (d \mid a$ and $d \mid b) \implies d \mid c$

Definitions

$(R, +, 0, \cdot, 1)$ commutative ring (often write ab for $a \cdot b = b \cdot a$)

- R *integral domain*: $a \cdot b = a \cdot c \implies a = 0$ or $b = c$ (cancellation law)
- $R^* = \{u \in R : \exists v \in R$ with $u \cdot v = 1\}$ (group of *units*)
  Notation: $v = u^{-1}$
- $a \in R \setminus R^*$ *irreducible*: $a = bc \implies b \in R^*$ or $c \in R^*$
- $a \mid b \ :\Longleftrightarrow \ \exists c$ with $ac = b$
- c *greatest common divisor* (GCD) of a and b:
  $c \mid a$ and $c \mid b$ and $\forall d \ (d \mid a$ and $d \mid b) \implies d \mid c$

Definitions

$(R, +, 0, \cdot, 1)$ commutative ring (often write ab for $a \cdot b = b \cdot a$)

- R *integral domain*: $a \cdot b = a \cdot c \implies a = 0$ or $b = c$ (cancellation law)

- $R^* = \{u \in R : \exists v \in R$ with $u \cdot v = 1\}$ (group of *units*)
  Notation: $v = u^{-1}$

- $a \in R \setminus R^*$ *irreducible*: $a = bc \implies b \in R^*$ or $c \in R^*$

- $a \mid b \;:\Longleftrightarrow\; \exists c$ with $ac = b$

- c *greatest common divisor* (GCD) of a and b:
  $c \mid a$ and $c \mid b$ and $\forall d \;\; (d \mid a$ and $d \mid b) \implies d \mid c$

Definitions

$(R, +, 0, \cdot, 1)$ commutative ring (often write ab for $a \cdot b = b \cdot a$)

- R *integral domain*: $a \cdot b = a \cdot c \implies a = 0$ or $b = c$ (cancellation law)
- $R^* = \{u \in R : \exists v \in R$ with $u \cdot v = 1\}$ (group of *units*)
  Notation: $v = u^{-1}$
- $a \in R \setminus R^*$ *irreducible*: $a = bc \implies b \in R^*$ or $c \in R^*$
- $a \mid b \;:\Longleftrightarrow\; \exists c$ with $ac = b$
- c *greatest common divisor* (GCD) of a and b:
  $c \mid a$ and $c \mid b$ and $\forall d \ \ (d \mid a$ and $d \mid b) \implies d \mid c$

Definitions

$(R, +, 0, \cdot, 1)$ commutative ring (often write ab for $a \cdot b = b \cdot a$)

- R *integral domain*: $a \cdot b = a \cdot c \implies a = 0$ or $b = c$ (cancellation law)
- $R^* = \{u \in R : \exists v \in R$ with $u \cdot v = 1\}$ (group of *units*)
  Notation: $v = u^{-1}$
- $a \in R \setminus R^*$ *irreducible*: $a = bc \implies b \in R^*$ or $c \in R^*$
- $a \mid b \;:\Longleftrightarrow\; \exists c$ with $ac = b$
- c *greatest common divisor* (GCD) of a and b:
  $c \mid a$ and $c \mid b$ and $\forall d \; (d \mid a$ and $d \mid b) \implies d \mid c$

Unique factorization domains

- $a \sim b \;:\Longleftrightarrow\; a \mid b$ and $b \mid a \;\Longleftrightarrow\; \exists u \in R^* \; a = ub$
  (a and b are *associates*)
  Exercise: all units are associates

- R *unique factorization domain* (UFD): R integral domain and $\forall a \in R \setminus \{0\} \; \exists u \in R^*, p_1, \ldots, p_r$ irreducible with $a = up_1 \cdots p_r$

  and if $a = vq_1 \cdots q_s$ with $v \in R^*$ and $q_1, \ldots, q_s$ irreducible, then $r = s$ and $p_1 \sim q_1, \ldots, p_r \sim q_r$ (up to reordering)

- Given the first condition, the second one is equivalent to the existence of a GCD for all $a, b \in R$

Unique factorization domains

- $a \sim b \;:\Longleftrightarrow\; a \mid b$ and $b \mid a \iff \exists u \in R^* \; a = ub$
  (a and b are *associates*)
  Exercise: all units are associates (Proof: $u = ab^{-1}$)

- R *unique factorization domain* (UFD): R integral domain and
  $\forall a \in R \setminus \{0\} \; \exists u \in R^*, p_1, \ldots, p_r$ irreducible with $a = up_1 \cdots p_r$

  and if $a = vq_1 \cdots q_s$ with $v \in R^*$ and $q_1, \ldots, q_s$ irreducible,
  then $r = s$ and $p_1 \sim q_1, \ldots, p_r \sim q_r$ (up to reordering)

- Given the first condition, the second one is equivalent to the existence of a GCD for all $a, b \in R$

Unique factorization domains

- $a \sim b \;:\Longleftrightarrow\; a \mid b$ and $b \mid a \iff \exists u \in R^* \; a = ub$
  (a and b are *associates*)
  Exercise: all units are associates (Proof: $u = ab^{-1}$)

- R *unique factorization domain* (UFD): R integral domain and
  $\forall a \in R \setminus \{0\} \; \exists u \in R^*, \; p_1, \ldots, p_r$ irreducible with $a = up_1 \cdots p_r$

  and if $a = vq_1 \cdots q_s$ with $v \in R^*$ and $q_1, \ldots, q_s$ irreducible,
  then $r = s$ and $p_1 \sim q_1, \ldots, p_r \sim q_r$ (up to reordering)

- Given the first condition, the second one is equivalent to the existence of a GCD for all $a, b \in R$

Unique factorization domains

- $a \sim b \;:\iff\; a \mid b$ and $b \mid a \iff \exists u \in R^* \; a = ub$
  (a and b are *associates*)
  Exercise: all units are associates (Proof: $u = ab^{-1}$)

- R *unique factorization domain* (UFD): R integral domain and
  $\forall a \in R \setminus \{0\} \; \exists u \in R^*$, $p_1, \ldots, p_r$ irreducible with $a = up_1 \cdots p_r$

  and if $a = vq_1 \cdots q_s$ with $v \in R^*$ and $q_1, \ldots, q_s$ irreducible,
  then $r = s$ and $p_1 \sim q_1, \ldots, p_r \sim q_r$ (up to reordering)

- Given the first condition, the second one is equivalent to the existence of a GCD for all $a, b \in R$

UFD examples

- $\mathbb{Z}$ is a UFD with $\mathbb{Z}^* = \{-1, 1\}$; thus $a \sim b \iff a = \pm b$.
  Irreducible elements: prime numbers and their negatives.
  All irreducible factorizations of 6:

  $$6 = 1 \cdot 2 \cdot 3 = 1 \cdot (-2) \cdot (-3) = -1 \cdot 2 \cdot (-3) = -1 \cdot (-2) \cdot 3$$

  -2 and 2 are all the GCDs of 4 and 6.

- $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, more generally any field F is a UFD with $F^* = F \setminus \{0\}$ and no irreducible elements.

- The univariate polynomail ring $\mathbb{Q}[x]$ is a UFD. More generally, a polynomial ring $R = F[x_1, \ldots, x_n]$ in n variables over a UFD F is a UFD, with $R^* = F^*$. In $\mathbb{C}[x]$, the irreducible elements are exactly the linear polynomials.

UFD examples

- $\mathbb{Z}$ is a UFD with $\mathbb{Z}^* = \{-1, 1\}$; thus $a \sim b \iff a = \pm b$.
  Irreducible elements: prime numbers and their negatives.
  All irreducible factorizations of 6:

  $$6 = 1 \cdot 2 \cdot 3 = 1 \cdot (-2) \cdot (-3) = -1 \cdot 2 \cdot (-3) = -1 \cdot (-2) \cdot 3$$

  -2 and 2 are all the GCDs of 4 and 6.

- $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, more generally any field F is a UFD with $F^* = F \setminus \{0\}$ and no irreducible elements.

- The univariate polynomail ring $\mathbb{Q}[x]$ is a UFD. More generally, a polynomial ring $R = F[x_1, \ldots, x_n]$ in n variables over a UFD F is a UFD, with $R^* = F^*$. In $\mathbb{C}[x]$, the irreducible elements are exactly the linear polynomials.

UFD examples

- $\mathbb{Z}$ is a UFD with $\mathbb{Z}^* = \{-1, 1\}$; thus $a \sim b \iff a = \pm b$.
  Irreducible elements: prime numbers and their negatives.
  All irreducible factorizations of 6:

  $$6 = 1 \cdot 2 \cdot 3 = 1 \cdot (-2) \cdot (-3) = -1 \cdot 2 \cdot (-3) = -1 \cdot (-2) \cdot 3$$

  -2 and 2 are all the GCDs of 4 and 6.

- $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, more generally any field F is a UFD with $F^* = F \setminus \{0\}$ and no irreducible elements.

- The univariate polynomail ring $\mathbb{Q}[x]$ is a UFD. More generally, a polynomial ring $R = F[x_1, \ldots, x_n]$ in n variables over a UFD F is a UFD, with $R^* = F^*$. In $\mathbb{C}[x]$, the irreducible elements are exactly the linear polynomials.

A non-example

$R = \mathbb{Z}[\sqrt{-5}] = \{a + b\sqrt{5}i : a, b \in \mathbb{Z}\}$ is not a UFD.

- A unit $u \in R$ has $\|u\| = a^2 + 5b^2 = 1$; thus $R^* = \{-1, 1\}$.
- $2, 3$ and $1 \pm \sqrt{5}i$ are all irreducible and

$$2 \cdot 3 = 6 = (1 + \sqrt{5}i)(1 - \sqrt{5}i)$$

  are two non-associated factorizations into irreducibles.
- The common factors of $a = 6$ and $b = 2 + 2\sqrt{5}i$ are

$$\{-1,\ 1,\ -2,\ 2,\ 1 + \sqrt{5}i,\ -1 - \sqrt{5}i\},$$

  and hence a and b do not have a GCD.

Units and normalization

It is convenient to have a normalized irreducible factorization and a function $\gcd$ and not have to worry about associates, so we pick a normal form.

- $a \in \mathbb{Z}$ is normalized $\iff a \geq 0$.
  $\gcd(a, b)$ is the unique nonnegative GCD of a and b.
  The normalized irreducible factorization of $a \neq 0$ is
  $a = up_1 \cdots p_r$ such that $u = \pm 1$ and $p_1, \ldots, p_r > 1$ are prime numbers.
- Let F be a field. $a \in F[x] \setminus \{0\}$ is normalized $\iff a$ is monic, i.e., has leading coefficient 1.
  $\gcd(a, b) :=$ unique monic GCD of nonzero polynomials a and b.
  The normalized irreducible factorization of $a \neq 0$ is
  $a = up_1 \cdots p_r$ such that $u \in F \setminus \{0\}$ and $p_1, \ldots, p_r$ are monic (non-constant) irreducible polynomials.

Units and normalization

It is convenient to have a normalized irreducible factorization and a function $\gcd$ and not have to worry about associates, so we pick a normal form.

- $a \in \mathbb{Z}$ is normalized $\iff a \geq 0$.
  $\gcd(a, b)$ is the unique nonnegative GCD of a and b.
  The normalized irreducible factorization of $a \neq 0$ is
  $a = up_1 \cdots p_r$ such that $u = \pm 1$ and $p_1, \ldots, p_r > 1$ are prime numbers.
- Let F be a field. $a \in F[x] \setminus \{0\}$ is normalized $\iff a$ is monic, i.e., has leading coefficient 1.
  $\gcd(a, b) :=$ unique monic GCD of nonzero polynomials a and b.
  The normalized irreducible factorization of $a \neq 0$ is
  $a = up_1 \cdots p_r$ such that $u \in F \setminus \{0\}$ and $p_1, \ldots, p_r$ are monic (non-constant) irreducible polynomials.

Coefficient domains

Remainder of this tutorial: *polynomial* GCDs and factorization. (Integer GCD algorithms are similar; integer factorization is *much* harder.)

- $\mathbb{Q}$
- finite field $\mathbb{F}_p$, where p is a prime number; ``integers modulo p''
- algebraic extensions, e.g., $\mathbb{Q}[i]$ (Gaussian integers)
- transcendental extensions by ``parameters'', e.g., $\mathbb{Q}(t)$ (rational functions in $t$). Expressions containing only parameters are considered ``constants'' (units).

Coefficient domains

Remainder of this tutorial: *polynomial* GCDs and factorization. (Integer GCD algorithms are similar; integer factorization is *much* harder.)

- $\mathbb{Q}$
- finite field $\mathbb{F}_p$, where p is a prime number; ``integers modulo p''
- algebraic extensions, e.g., $\mathbb{Q}[i]$ (Gaussian integers)
- transcendental extensions by ``parameters'', e.g., $\mathbb{Q}(t)$ (rational functions in $t$). Expressions containing only parameters are considered ``constants'' (units).

Coefficient domains

Remainder of this tutorial: *polynomial* GCDs and factorization. (Integer GCD algorithms are similar; integer factorization is *much* harder.)

- $\mathbb{Q}$
- finite field $\mathbb{F}_p$, where p is a prime number; ``integers modulo p''
- algebraic extensions, e.g., $\mathbb{Q}[i]$ (Gaussian integers)
- transcendental extensions by ``parameters'', e.g., $\mathbb{Q}(t)$ (rational functions in $t$). Expressions containing only parameters are considered ``constants'' (units).

Coefficient domains

Remainder of this tutorial: *polynomial* GCDs and factorization. (Integer GCD algorithms are similar; integer factorization is *much* harder.)

- $\mathbb{Q}$
- finite field $\mathbb{F}_p$, where p is a prime number; ``integers modulo p''
- algebraic extensions, e.g., $\mathbb{Q}[i]$ (Gaussian integers)
- transcendental extensions by ``parameters'', e.g., $\mathbb{Q}(t)$ (rational functions in $t$). Expressions containing only parameters are considered ``constants'' (units).

Cost models

F field, $R = F[x_1, \ldots, x_n]$

Model for cost analysis of algorithms in R: *arithmetic RAM*

- Sequential algorithms (parallel algorithms possible by considering length of critical path instead of total cost)
- Unit cost for one arithmetic operation $+, -, \cdot$, or $^{-1}$ in F
- Variables $x_1, \ldots, x_n$ are just ``placeholders'' and multiplication by a product of variables is ``for free''
- If $F = \mathbb{Q}$ or $F = \mathbb{F}_p$, the *word RAM* model also assigns a non-trivial cost to arithmetic operation in F, depending on the size (number of machine words) of the operands in memory
- Cost for zero testing, memory management, loop index arithmetic etc. is considered non-dominant and therefore ignored

Cost models

F field, $R = F[x_1, \ldots, x_n]$
Model for cost analysis of algorithms in R: *arithmetic RAM*

- Sequential algorithms (parallel algorithms possible by considering length of critical path instead of total cost)
- Unit cost for one arithmetic operation $+, -, \cdot$, or $^{-1}$ in F
- Variables $x_1, \ldots, x_n$ are just ``placeholders'' and multiplication by a product of variables is ``for free''
- If $F = \mathbb{Q}$ or $F = \mathbb{F}_p$, the *word RAM* model also assigns a non-trivial cost to arithmetic operation in F, depending on the size (number of machine words) of the operands in memory
- Cost for zero testing, memory management, loop index arithmetic etc. is considered non-dominant and therefore ignored

Cost models

F field, $R = F[x_1, \ldots, x_n]$

Model for cost analysis of algorithms in R: *arithmetic RAM*

- Sequential algorithms (parallel algorithms possible by considering length of critical path instead of total cost)
- Unit cost for one arithmetic operation $+, -, \cdot$, or $^{-1}$ in F
- Variables $x_1, \ldots, x_n$ are just ``placeholders'' and multiplication by a product of variables is ``for free''
- If $F = \mathbb{Q}$ or $F = \mathbb{F}_p$, the *word RAM* model also assigns a non-trivial cost to arithmetic operation in F, depending on the size (number of machine words) of the operands in memory
- Cost for zero testing, memory management, loop index arithmetic etc. is considered non-dominant and therefore ignored

Cost models

F field, $R = F[x_1, \ldots, x_n]$
Model for cost analysis of algorithms in R: *arithmetic RAM*

- Sequential algorithms (parallel algorithms possible by considering length of critical path instead of total cost)
- Unit cost for one arithmetic operation $+, -, \cdot$, or $^{-1}$ in F
- Variables $x_1, \ldots, x_n$ are just ``placeholders'' and multiplication by a product of variables is ``for free''
- If $F = \mathbb{Q}$ or $F = \mathbb{F}_p$, the *word RAM* model also assigns a non-trivial cost to arithmetic operation in F, depending on the size (number of machine words) of the operands in memory
- Cost for zero testing, memory management, loop index arithmetic etc. is considered non-dominant and therefore ignored

Cost models

F field, $R = F[x_1, \ldots, x_n]$
Model for cost analysis of algorithms in R: *arithmetic RAM*

- Sequential algorithms (parallel algorithms possible by considering length of critical path instead of total cost)
- Unit cost for one arithmetic operation $+, -, \cdot$, or $^{-1}$ in F
- Variables $x_1, \ldots, x_n$ are just ``placeholders'' and multiplication by a product of variables is ``for free''
- If $F = \mathbb{Q}$ or $F = \mathbb{F}_p$, the *word RAM* model also assigns a non-trivial cost to arithmetic operation in F, depending on the size (number of machine words) of the operands in memory
- Cost for zero testing, memory management, loop index arithmetic etc. is considered non-dominant and therefore ignored

Classical vs fast arithmetic

- ``Classical'' algorithms are typically quadratic in the input size. E.g., multiplication of two polynomials of degree $\leq n$ in $F[x]$ takes $(n+1)^2$ multiplications in F and n^2 additions, in total $2n^2 + 2n - 1 \in O(n^2)$ arithmetic operations in F.
- ``Asymptotically fast'' algorithms exist and take only $O(n \log^k n)$ operations for some $k \in \mathbb{N}$.
- Notation: *multiplication time* $\mathsf{M}(n)$ = number of arithmetic operations in F sufficient to multiply two univariate polynomials of degree $\leq n$.
- Classical arithmetic: $\mathsf{M}(n) = 2n^2 + 2n + 1 \in O(n^2)$
- Fast arithmetic: $\mathsf{M}(n) \in O(n \log n \operatorname{loglog} n)$

  (Schönhage & Strassen)

Classical vs fast arithmetic

- ``Classical'' algorithms are typically quadratic in the input size. E.g., multiplication of two polynomials of degree $\leq n$ in $F[x]$ takes $(n+1)^2$ multiplications in F and n^2 additions, in total $2n^2 + 2n - 1 \in O(n^2)$ arithmetic operations in F.
- ``Asymptotically fast'' algorithms exist and take only $O(n \log^k n)$ operations for some $k \in \mathbb{N}$.
- Notation: *multiplication time* $\mathsf{M}(n)$ = number of arithmetic operations in F sufficient to multiply two univariate polynomials of degree $\leq n$.
- Classical arithmetic: $\mathsf{M}(n) = 2n^2 + 2n + 1 \in O(n^2)$
- Fast arithmetic: $\mathsf{M}(n) \in O(n \log n \log\log n)$

(Schönhage & Strassen)

Classical vs fast arithmetic

- ``Classical'' algorithms are typically quadratic in the input size. E.g., multiplication of two polynomials of degree $\leq n$ in $F[x]$ takes $(n+1)^2$ multiplications in F and n^2 additions, in total $2n^2+2n-1 \in O(n^2)$ arithmetic operations in F.
- ``Asymptotically fast'' algorithms exist and take only $O(n \log^k n)$ operations for some $k \in \mathbb{N}$.
- Notation: *multiplication time* $\mathsf{M}(n)$ = number of arithmetic operations in F sufficient to multiply two univariate polynomials of degree $\leq n$.
- Classical arithmetic: $\mathsf{M}(n) = 2n^2+2n+1 \in O(n^2)$
- Fast arithmetic: $\mathsf{M}(n) \in O(n \log n \operatorname{loglog} n)$

  (Schönhage & Strassen)

Classical vs fast arithmetic

- ``Classical'' algorithms are typically quadratic in the input size. E.g., multiplication of two polynomials of degree $\leq n$ in $F[x]$ takes $(n+1)^2$ multiplications in F and n^2 additions, in total $2n^2+2n-1 \in O(n^2)$ arithmetic operations in F.
- ``Asymptotically fast'' algorithms exist and take only $O(n \log^k n)$ operations for some $k \in \mathbb{N}$.
- Notation: *multiplication time* $\mathsf{M}(n)$ = number of arithmetic operations in F sufficient to multiply two univariate polynomials of degree $\leq n$.
- Classical arithmetic: $\mathsf{M}(n) = 2n^2+2n+1 \in O(n^2)$
- Fast arithmetic: $\mathsf{M}(n) \in O(n \log n \operatorname{loglog} n)$

(Schönhage & Strassen)

Classical vs fast arithmetic

- ``Classical'' algorithms are typically quadratic in the input size. E.g., multiplication of two polynomials of degree $\leq n$ in $F[x]$ takes $(n+1)^2$ multiplications in F and n^2 additions, in total $2n^2 + 2n - 1 \in O(n^2)$ arithmetic operations in F.
- ``Asymptotically fast'' algorithms exist and take only $O(n \log^k n)$ operations for some $k \in \mathbb{N}$.
- Notation: *multiplication time* $\mathsf{M}(n) =$ number of arithmetic operations in F sufficient to multiply two univariate polynomials of degree $\leq n$.
- Classical arithmetic: $\mathsf{M}(n) = 2n^2 + 2n + 1 \in O(n^2)$
- Fast arithmetic: $\mathsf{M}(n) \in O(n \log n \operatorname{loglog} n)$

(Schönhage & Strassen)

Basic univariate polynomial arithmetic cost

$f, g \in F[x]$ polynomials, $\deg g = m \leq n = \deg f$, $a \in F$ constant

Operation	Classical	Fast
$f(a)$	$2n-2$	$2n-2$
$f+g$	$m+1$	$m+1$
$f \cdot g$	$2mn + O(n)$	$\mathsf{M}(n)$
$f \operatorname{quo} g$	$O(m(n-m))$	$O(\mathsf{M}(n-m))$
$f \operatorname{rem} g$	$O(m(n-m))$	$O(\mathsf{M}(n))$

Note: $f(a) = f \operatorname{rem} (x-a)$

Outline

Euclidean algorithm I

It is straightforward to compute GCDs from factorizations, but there is a much more efficient and famous algorithm first introduced for integers.

Example: Compute the (monic) $\gcd$ of $x^5 + x^3 + x^2 - 2x$ and $x^4 - x^2 + x$. Iterated division with remainder:

$$\begin{aligned} x^5 + x^3 + x^2 - 2x &= x \cdot (x^4 - x^2 + x) + 2x^3 - 2x, \\ x^4 - x^2 + x &= \frac{1}{2}x \cdot (2x^3 - 2x) + x, \\ 2x^3 - 2x &= (2x^2 - 2) \cdot x + 0, \\ x &= \gcd(x^5 + x^3 + x^2 - 2x, x^4 - x^2 + x) \end{aligned}$$

Euclidean algorithm II

$$\begin{aligned} x^5 + x^3 + x^2 - 2x &= x \cdot (x^4 - x^2 + x) + 2x^3 - 2x, \\ x^4 - x^2 + x &= \frac{1}{2} x \cdot (2x^3 - 2x) + x, \\ 2x^3 - 2x &= (2x^2 - 2) \cdot x + 0, \\ x &= \gcd(x^5 + x^3 + x^2 - 2x, x^4 - x^2 + x) \end{aligned}$$

Observations:

- Even though the input polynomials are monic, the quotients and remainders may not be.
- Even though the input polynomials have integer coefficients, the quotients and remainders may have denominators.
- The degree can decrease by more than 1 in a single step.

Euclidean algorithm II

$$\begin{aligned}
x^5 + x^3 + x^2 - 2x &= x \cdot (x^4 - x^2 + x) + 2x^3 - 2x, \\
x^4 - x^2 + x &= \frac{1}{2} x \cdot (2x^3 - 2x) + x, \\
2x^3 - 2x &= (2x^2 - 2) \cdot x + 0, \\
x &= \gcd(x^5 + x^3 + x^2 - 2x, x^4 - x^2 + x)
\end{aligned}$$

Observations:

- Even though the input polynomials are monic, the quotients and remainders may not be.
- Even though the input polynomials have integer coefficients, the quotients and remainders may have denominators.
- The degree can decrease by more than 1 in a single step.

Euclidean algorithm II

$$\begin{aligned} x^5+x^3+x^2-2x &= x\cdot(x^4-x^2+x)+2x^3-2x, \\ x^4-x^2+x &= \frac{1}{2}x\cdot(2x^3-2x)+x, \\ 2x^3-2x &= (2x^2-2)\cdot x+0, \\ x &= \gcd(x^5+x^3+x^2-2x, x^4-x^2+x) \end{aligned}$$

Observations:

- Even though the input polynomials are monic, the quotients and remainders may not be.
- Even though the input polynomials have integer coefficients, the quotients and remainders may have denominators.
- The degree can decrease by more than 1 in a single step.

Euclidean algorithm III

Input: $f, g \in F[x]$ for a field F
Output: $\gcd(f, g) \in F[x]$

while $g \neq 0$ **do**

$$\begin{pmatrix} f \\ g \end{pmatrix} \leftarrow \begin{pmatrix} g \\ f \operatorname{rem} g \end{pmatrix}$$

return f

Is this correct?

Euclidean algorithm IV

Input: $f, g \in F[x]$ for a field F
Output: $\gcd(f, g) \in F[x]$

1. $\begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \leftarrow \begin{pmatrix} f \\ g \end{pmatrix}$
2. **for** $i \geq 1$ **while** $r_i \neq 0$ **do**
3. $\quad r_{i+1} \leftarrow r_{i-1} \operatorname{rem} r_i$
4. **return** $\dfrac{r_{i-1}}{\operatorname{lc}(r_{i-1})}$

Euclidean algorithm IV

Input: $f, g \in F[x]$ for a field F
Output: $\gcd(f, g) \in F[x]$

1. $\begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \leftarrow \begin{pmatrix} f \\ g \end{pmatrix}$
2. **for** $i \geq 1$ **while** $r_i \neq 0$ **do**
3. $\quad r_{i+1} \leftarrow r_{i-1} \text{ rem } r_i$
4. **return** $\dfrac{r_{i-1}}{\mathrm{lc}(r_{i-1})}$

Euclidean algorithm IV

Input: $f, g \in F[x]$ for a field F
Output: $\gcd(f, g) \in F[x]$

$\begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \leftarrow \begin{pmatrix} f \\ g \end{pmatrix}$

for $i \geq 1$ **while** $r_i \neq 0$ **do**

  $r_{i+1} \leftarrow r_{i-1} \text{ rem } r_i$

return $\dfrac{r_{i-1}}{\text{lc}(r_{i-1})}$

Euclidean algorithm IV

Input: $f, g \in F[x]$ for a field F
Output: $\gcd(f, g) \in F[x]$

1. $\begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \leftarrow \begin{pmatrix} f \\ g \end{pmatrix}$
2. **for** $i \geq 1$ **while** $r_i \neq 0$ **do**
3. $\quad r_{i+1} \leftarrow r_{i-1} \text{ rem } r_i$
4. **return** $\dfrac{r_{i-1}}{\text{lc}(r_{i-1})}$

Remark: $\text{lc}(0) := 1, \deg 0 := -\infty$

Cost

$f, g \in F[x]$, $n = \deg f \geq \deg g = m$, $f \neq 0$
Let $n_i = \deg r_i$ for $1 \leq i \leq \ell$ such that $r_{\ell+1} = 0$.

Cost for division with remainder in step 3: $O(n_i \cdot (n_{i-1} - n_i))$

Cost for normalization in step 4: n_ℓ

Total cost: $n_\ell + \sum_{1 \leq i \leq \ell} O(n_i(n_{i-1} - n_i)) = O(nm)$

Variants

- Monic EA: normalize remainder at every step, not just at the end: still $O(nm)$ but smaller coefficients
- Asymptotically fast EA: $O(\mathsf{M}(n)\log n)$ (divide-and-conquer, Knuth, Schönhage, Moenck, ...)
- Subresultant algorithm: fraction-free (Collins)
- Modular algorithms (Brown, Collins, ...). We'll come back to this later.

Variants

- Monic EA: normalize remainder at every step, not just at the end: still $O(nm)$ but smaller coefficients
- Asymptotically fast EA: $O(\mathsf{M}(n)\log n)$ (divide-and-conquer, Knuth, Schönhage, Moenck, ...)

- Subresultant algorithm: fraction-free (Collins)
- Modular algorithms (Brown, Collins, ...). We'll come back to this later.

Variants

- Monic EA: normalize remainder at every step, not just at the end: still $O(nm)$ but smaller coefficients
- Asymptotically fast EA: $O(\mathsf{M}(n) \log n)$ (divide-and-conquer, Knuth, Schönhage, Moenck, ...)

If $F = \mathbb{Q}$ and $f, g \in \mathbb{Z}[x]$ (improved cost in the word RAM model):

- Subresultant algorithm: fraction-free (Collins)
- Modular algorithms (Brown, Collins, ...). We'll come back to this later.

Variants

- Monic EA: normalize remainder at every step, not just at the end: still $O(nm)$ but smaller coefficients
- Asymptotically fast EA: $O(\mathsf{M}(n)\log n)$ (divide-and-conquer, Knuth, Schönhage, Moenck, ...)

If $F = \mathbb{Q}$ and $f, g \in \mathbb{Z}[x]$ (improved cost in the word RAM model):

- Subresultant algorithm: fraction-free (Collins)
- Modular algorithms (Brown, Collins, ...). We'll come back to this later.

Similar if F is a rational function field and f, g are multivariate polynomials.

Extended Euclidean Algorithm

Input: $f, g \in F[x]$ for a field F
Output: $r \in F[x]$ such that $r = \gcd(f, g)$

$\begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \leftarrow \begin{pmatrix} f \\ g \end{pmatrix}$,

for $i \geq 1$ **while** $r_i \neq 0$ **do**

$\quad q_i \leftarrow r_{i-1} \text{ quo } r_i$

$\quad r_{i+1} \leftarrow r_{i-1} - q_i r_i \; (= r_{i-1} \text{ rem } r_i)$

return $\dfrac{r_{i-1}}{\mathrm{lc}(r_{i-1})}$

Extended Euclidean Algorithm

Input: $f, g \in F[x]$ for a field F

Output: r, $\boldsymbol{s}$, $\boldsymbol{t} \in F[x]$ such that $\boldsymbol{sf} + \boldsymbol{tg} = r = \gcd(f, g)$

$$\begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \leftarrow \begin{pmatrix} f \\ g \end{pmatrix}, \begin{pmatrix} \boldsymbol{s_0} \\ \boldsymbol{s_1} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{t_0} \\ \boldsymbol{t_1} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix}$$

for $i \geq 1$ **while** $r_i \neq 0$ **do**

$\quad q_i \leftarrow r_{i-1} \operatorname{quo} r_i$

$\quad r_{i+1} \leftarrow r_{i-1} - q_i r_i \; (= r_{i-1} \operatorname{rem} r_i)$

$\quad \boldsymbol{s_{i+1} \leftarrow s_{i-1} - q_i s_i}$

$\quad \boldsymbol{t_{i+1} \leftarrow t_{i-1} - q_i t_i}$

return $\dfrac{r_{i-1}}{\operatorname{lc}(r_{i-1})}, \boldsymbol{\dfrac{s_{i-1}}{\operatorname{lc}(r_{i-1})}}, \boldsymbol{\dfrac{t_{i-1}}{\operatorname{lc}(r_{i-1})}}$

Extended Euclidean Algorithm

Input: $f, g \in F[x]$ for a field F

Output: r, $\boldsymbol{s}$, $\boldsymbol{t} \in F[x]$ such that $\boldsymbol{sf} + \boldsymbol{tg} = r = \gcd(f, g)$

1 $\begin{pmatrix} r_0 \\ r_1 \end{pmatrix} \leftarrow \begin{pmatrix} f \\ g \end{pmatrix}$, $\begin{pmatrix} \boldsymbol{s_0} \\ \boldsymbol{s_1} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix}$, $\begin{pmatrix} \boldsymbol{t_0} \\ \boldsymbol{t_1} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix}$

2 **for** $i \geq 1$ **while** $r_i \neq 0$ **do**

3 $\quad q_i \leftarrow r_{i-1} \text{ quo } r_i$

4 $\quad r_{i+1} \leftarrow r_{i-1} - q_i r_i \; (= r_{i-1} \text{ rem } r_i)$

5 $\quad \boldsymbol{s_{i+1} \leftarrow s_{i-1} - q_i s_i}$

6 $\quad \boldsymbol{t_{i+1} \leftarrow t_{i-1} - q_i t_i}$

7 **return** $\dfrac{r_{i-1}}{\mathrm{lc}(r_{i-1})}, \dfrac{\boldsymbol{s_{i-1}}}{\mathbf{lc}(\boldsymbol{r_{i-1}})}, \dfrac{\boldsymbol{t_{i-1}}}{\mathbf{lc}(\boldsymbol{r_{i-1}})}$

Cost: $O(nm)$

Example (cont'd)

i	q_i	r_i	s_i	t_i
0		$x^5 + x^3 + x^2 - 2x$	1	0
1	x	$x^4 - x^2 + x$	0	1
2	$\frac{1}{2}x$	$2x^3 - 2x$	1	$-x$
3	$2x^2 - 2$	x	$-\frac{1}{2}x$	$\frac{1}{2}x^2 + 1$
4		0	$x^3 - x + 1$	$-x^4 - x^2 - x + 2$

- $\ell = \#\text{quotients} = 3$
- $\gcd(f, g) = \frac{r_3}{1} = x = -\frac{1}{2}x \cdot f + \left(\frac{1}{2}x^2 + 1\right) \cdot g = \frac{s_3}{1} f + \frac{t_3}{1} g$
- Invariant I: $r_i = s_i f + t_i g$ for $0 \le i \le \ell + 1$
- Invariant II: $\deg s_i = \deg g - \deg r_{i-1}$ for $1 < i \le \ell + 1$ and $\deg t_i = \deg f - \deg r_{i-1}$ for $1 \le i \le \ell + 1$
- Last row: *cofactors* $u_{\ell+1}, v_{\ell+1}$ such that $f = (-1)^{\ell+1} u_{\ell+1} r_\ell$ and $g = (-1)^\ell v_{\ell+1} r_\ell$

Example (cont'd)

i	q_i	r_i	s_i	t_i
0		$x^5 + x^3 + x^2 - 2x$	1	0
1	x	$x^4 - x^2 + x$	0	1
2	$\frac{1}{2}x$	$2x^3 - 2x$	1	$-x$
3	$2x^2 - 2$	x	$-\frac{1}{2}x$	$\frac{1}{2}x^2 + 1$
4		0	$x^3 - x + 1$	$-x^4 - x^2 - x + 2$

- $\ell = \#\text{quotients} = 3$
- $\gcd(f, g) = \dfrac{r_3}{1} = x = -\dfrac{1}{2}x \cdot f + \left(\dfrac{1}{2}x^2 + 1\right) \cdot g = \dfrac{s_3}{1}f + \dfrac{t_3}{1}g$
- Invariant I: $r_i = s_i f + t_i g$ for $0 \leq i \leq \ell + 1$
- Invariant II: $\deg s_i = \deg g - \deg r_{i-1}$ for $1 < i \leq \ell + 1$ and $\deg t_i = \deg f - \deg r_{i-1}$ for $1 \leq i \leq \ell + 1$
- Last row: *cofactors* $u_{\ell+1}, v_{\ell+1}$ such that $f = (-1)^{\ell+1} u_{\ell+1} r_\ell$ and $g = (-1)^\ell v_{\ell+1} r_\ell$

Example (cont'd)

i	q_i	r_i	s_i	t_i
0		$x^5 + x^3 + x^2 - 2x$	1	0
1	x	$x^4 - x^2 + x$	0	1
2	$\frac{1}{2}x$	$2x^3 - 2x$	1	$-x$
3	$2x^2 - 2$	x	$-\frac{1}{2}x$	$\frac{1}{2}x^2 + 1$
4		0	$x^3 - x + 1$	$-x^4 - x^2 - x + 2$

- $\ell = \#\text{quotients} = 3$
- $\gcd(f, g) = \frac{r_3}{1} = x = -\frac{1}{2}x \cdot f + \left(\frac{1}{2}x^2 + 1\right) \cdot g = \frac{s_3}{1}f + \frac{t_3}{1}g$
- Invariant I: $r_i = s_i f + t_i g$ for $0 \leq i \leq \ell + 1$
- Invariant II: $\deg s_i = \deg g - \deg r_{i-1}$ for $1 < i \leq \ell + 1$ and $\deg t_i = \deg f - \deg r_{i-1}$ for $1 \leq i \leq \ell + 1$
- Last row: *cofactors* $u_{\ell+1}, v_{\ell+1}$ such that $f = (-1)^{\ell+1} u_{\ell+1} r_\ell$ and $g = (-1)^\ell v_{\ell+1} r_\ell$

Example (cont'd)

i	q_i	r_i	s_i	t_i
0		$x^5 + x^3 + x^2 - 2x$	1	0
1	x	$x^4 - x^2 + x$	0	1
2	$\frac{1}{2}x$	$2x^3 - 2x$	1	$-x$
3	$2x^2 - 2$	x	$-\frac{1}{2}x$	$\frac{1}{2}x^2 + 1$
4		0	$x^3 - x + 1$	$-x^4 - x^2 - x + 2$

- $\ell = \#\text{quotients} = 3$
- $\gcd(f, g) = \dfrac{r_3}{1} = x = -\dfrac{1}{2}x \cdot f + \left(\dfrac{1}{2}x^2 + 1\right) \cdot g = \dfrac{s_3}{1}f + \dfrac{t_3}{1}g$
- Invariant I: $r_i = s_i f + t_i g$ for $0 \le i \le \ell + 1$
- Invariant II: $\deg s_i = \deg g - \deg r_{i-1}$ for $1 < i \le \ell + 1$
  and $\deg t_i = \deg f - \deg r_{i-1}$ for $1 \le i \le \ell + 1$
- Last row: *cofactors* $u_{\ell+1}, v_{\ell+1}$ such that
  $f = (-1)^{\ell+1} u_{\ell+1} r_\ell$ and $g = (-1)^\ell v_{\ell+1} r_\ell$

Example (cont'd)

i	q_i	r_i	s_i	t_i
0		$x^5+x^3+x^2-2x$	1	0
1	x	x^4-x^2+x	0	1
2	$\frac{1}{2}x$	$2x^3-2x$	1	$-x$
3	$2x^2-2$	x	$-\frac{1}{2}x$	$\frac{1}{2}x^2+1$
4		0	x^3-x+1	$-x^4-x^2-x+2$

- $\ell = \#\text{quotients} = 3$
- $\gcd(f,g) = \dfrac{r_3}{1} = x = -\dfrac{1}{2}x \cdot f + \left(\dfrac{1}{2}x^2+1\right) \cdot g = \dfrac{s_3}{1} f + \dfrac{t_3}{1} g$
- Invariant I: $r_i = s_i f + t_i g$ for $0 \le i \le \ell + 1$
- Invariant II: $\deg s_i = \deg g - \deg r_{i-1}$ for $1 < i \le \ell + 1$ and $\deg t_i = \deg f - \deg r_{i-1}$ for $1 \le i \le \ell + 1$
- Last row: *cofactors* $u_{\ell+1}, v_{\ell+1}$ such that $f = (-1)^{\ell+1} u_{\ell+1} r_\ell$ and $g = (-1)^\ell v_{\ell+1} r_\ell$

Application I: modular inverses

Given $f, g \in F[x] \setminus \{0\}$ with f irreducible and $\deg g < \deg f$, compute

$$h = g^{-1} \bmod f,$$

i.e., $h \in F[x]$ with $\deg h < \deg f$ and $f \mid (gh - 1)$.

Solution: Since f is irreducible, $\gcd(f, g) = 1 = sf + tg$, so $h = t$.

This has applications, e.g., in modular arithmetic (later).

Application II: partial fractions

Given $f, g, r \in F[x] \setminus \{0\}$ with $\gcd(f, g) = 1$ and $\deg r < \deg f + \deg g$, find $u, v \in F[x]$ with $\deg u < \deg f$, $\deg v < \deg g$, and

$$\frac{r}{fg} = \frac{u}{f} + \frac{v}{g}.$$

Solution: $\gcd(f, g) = 1 = sf + tg$, so $r = rsf + rtg$. Let $q = rs \operatorname{quo} g$, $v = rs \operatorname{rem} g = rs - qg$ and $u = rt + qf$, then

$$\frac{r}{fg} = \frac{rt}{f} + \frac{rs}{g} = \left(\frac{rt}{f} + \frac{qf}{f}\right) + \left(\frac{rs}{g} - \frac{qg}{g}\right) = \frac{u}{f} + \frac{v}{g}.$$

This has applications, e.g., in symbolic integration (Hermite).

Application III: rational interpolation

Given a collection of n points $(x_j, y_j) \in F^2$, find a rational function $\rho = \frac{u}{v}$, with $u, v \in F[x]$ such that $\deg u \leq k$ and $\deg v < n - k$, that interpolates all the points: $\rho(x_j) = \frac{u(x_j)}{v(x_j)} = y_j$ for $1 \leq j \leq n$.

Solution: $f = (x - x_1) \cdots (x - x_n)$, $g =$ Lagrange interpolation polynomial. In the EEA for f and g, stop at i such that $\deg r_i < k \leq \deg r_{i-1}$. Then

$$r_i(x_j) = s_i(x_j)f(x_j) + t_i(x_j)g(x_j) = t_i(x_j)y_j,$$

so $\rho = u/v = r_i/t_i$ is a solution unless $t_i(x_j) = 0$ for some j (in which case no solution exists).

This has applications, e.g., in bivariate gcd computation (later).

Application IV: Padé approximation

Given a sufficiently smooth function $c : F \to F$, find a rational function $\rho = \frac{u}{v}$, with $u, v \in F[x]$ such that $\deg u \leq k$ and $\deg v < n - k$, such that the Taylor expansions of c and ρ at $x = 0$ agree for the first n terms: $\rho^{(j)}(0) = c^{(j)}(0)$ for $0 \leq j < n$

Solution: $f = x^n$, $g = n$th Taylor polynomial of c. In the EEA for f and g, stop at i such that $\deg r_i < k \leq \deg r_{i-1}$. Then $\rho = u/v = r_i/t_i$ is a solution since

$$\rho^{(j)}(0) = \left(\frac{r_i}{t_i}\right)^{(j)}(0) = \left(\frac{s_i}{t_i}x^n\right)^{(j)}(0) + g^{(j)}(0) = c^{(j)}(0),$$

unless $t_i(0) = 0$ (in which case no solution exists).

This has applications, e.g., in coding theory (Berlekamp-Massey algorithm) and bivariate factorization (later).

Asymptotically fast EEA

It is not possible to compute *all* r_i, s_i, t_i for $1 \le i \le \ell$ in time $O(\mathsf{M}(n) \log n)$, but all the previous applications require is r_i, s_i, t_i for one specific value of i (e.g., $i = \ell$), and that can be computed in time $O(\mathsf{M}(n) \log n)$.

Outline

Modular arithmetic

Let $p > 1$ be a prime number.

- $\mathbb{F}_p := \{0, \dots, p-1\}$ with addition $(a+b)$ rem p, negation $-a = p - a$ (for $a \neq 0$), and multiplication $(a \cdot b)$ rem p.
  Examples: $3 + 5 = 1$, $-2 = 5$, and $3 \cdot 5 = 1$ in $\mathbb{F}_7$.
- Every nonzero element $a \in \mathbb{F}_p$ has a multiplicative inverse: EEA in $\mathbb{Z}$ computes $1 = sp + ta$, so ta rem $p = (sp + ta)$ rem $p = 1$.
  Thus: $\mathbb{F}_p$ is a field.
- Example: $\gcd(7, 3) = 1 = 1 \cdot 7 - 2 \cdot 3$, so $3^{-1} = -2 = 5$ in $\mathbb{F}_7$.
- Cost for one arithmetic operation in $\mathbb{F}_p$ in word RAM model:

	classical	asymptotically fast
$+/-$	$O(\log p)$	$O(\log p)$
$\cdot$	$O(\log^2 p)$	$O(\mathsf{M}(\log p))$
$^{-1}$	$O(\log^2 p)$	$O(\mathsf{M}(\log p) \operatorname{loglog} p)$

Modular arithmetic

Let $p > 1$ be a prime number.

- $\mathbb{F}_p := \{0, \ldots, p-1\}$ with addition $(a+b) \text{ rem } p$, negation $-a = p - a$ (for $a \neq 0$), and multiplication $(a \cdot b) \text{ rem } p$.
  Examples: $3 + 5 = 1$, $-2 = 5$, and $3 \cdot 5 = 1$ in $\mathbb{F}_7$.
- Every nonzero element $a \in \mathbb{F}_p$ has a multiplicative inverse: EEA in $\mathbb{Z}$ computes $1 = sp + ta$, so $ta \text{ rem } p = (sp + ta) \text{ rem } p = 1$.
  Thus: $\mathbb{F}_p$ is a field.
- Example: $\gcd(7, 3) = 1 = 1 \cdot 7 - 2 \cdot 3$, so $3^{-1} = -2 = 5$ in $\mathbb{F}_7$.
- Cost for one arithmetic operation in $\mathbb{F}_p$ in word RAM model:

	classical	asymptotically fast
$+/-$	$O(\log p)$	$O(\log p)$
$\cdot$	$O(\log^2 p)$	$O(\mathsf{M}(\log p))$
$^{-1}$	$O(\log^2 p)$	$O(\mathsf{M}(\log p) \operatorname{loglog} p)$

Modular arithmetic

Let $p > 1$ be a prime number.

- $\mathbb{F}_p := \{0, \ldots, p-1\}$ with addition $(a+b)$ rem p, negation $-a = p - a$ (for $a \neq 0$), and multiplication $(a \cdot b)$ rem p. Examples: $3 + 5 = 1$, $-2 = 5$, and $3 \cdot 5 = 1$ in $\mathbb{F}_7$.
- Every nonzero element $a \in \mathbb{F}_p$ has a multiplicative inverse: EEA in $\mathbb{Z}$ computes $1 = sp + ta$, so $ta \text{ rem } p = (sp + ta) \text{ rem } p = 1$. Thus: $\mathbb{F}_p$ is a field.
- Example: $\gcd(7, 3) = 1 = 1 \cdot 7 - 2 \cdot 3$, so $3^{-1} = -2 = 5$ in $\mathbb{F}_7$.
- Cost for one arithmetic operation in $\mathbb{F}_p$ in word RAM model:

	classical	asymptotically fast
$+/-$	$O(\log p)$	$O(\log p)$
$\cdot$	$O(\log^2 p)$	$O(\mathsf{M}(\log p))$
$^{-1}$	$O(\log^2 p)$	$O(\mathsf{M}(\log p) \operatorname{loglog} p)$

Modular arithmetic

Let $p > 1$ be a prime number.

- $\mathbb{F}_p := \{0, \ldots, p-1\}$ with addition $(a + b) \operatorname{rem} p$, negation $-a = p - a$ (for $a \neq 0$), and multiplication $(a \cdot b) \operatorname{rem} p$.
  Examples: $3 + 5 = 1$, $-2 = 5$, and $3 \cdot 5 = 1$ in $\mathbb{F}_7$.
- Every nonzero element $a \in \mathbb{F}_p$ has a multiplicative inverse: EEA in $\mathbb{Z}$ computes $1 = sp + ta$, so $ta \operatorname{rem} p = (sp + ta) \operatorname{rem} p = 1$.
  Thus: $\mathbb{F}_p$ is a field.
- Example: $\gcd(7, 3) = 1 = 1 \cdot 7 - 2 \cdot 3$, so $3^{-1} = -2 = 5$ in $\mathbb{F}_7$.
- Cost for one arithmetic operation in $\mathbb{F}_p$ in word RAM model:

	classical	asymptotically fast
$+/-$	$O(\log p)$	$O(\log p)$
$\cdot$	$O(\log^2 p)$	$O(\mathsf{M}(\log p))$
-1	$O(\log^2 p)$	$O(\mathsf{M}(\log p) \operatorname{loglog} p)$

Fermat's Little Theorem

$$a^p = a \text{ for all } a \in \mathbb{F}_p$$

Proof: Induction on a and the fact that $\binom{p}{j}$ is divisible by p for $0 < j < p$.

Note: The polynomial $x^p - x \in \mathbb{F}_p[x]$ is not the zero polynomial but vanishes at all points $a \in \mathbb{F}_p$.

Exercise: Devise a method to compute inverses in $\mathbb{F}_p$ using FLT instead EEA.

Fermat's Little Theorem

$$a^p = a \text{ for all } a \in \mathbb{F}_p$$

Proof: Induction on a and the fact that $\binom{p}{j}$ is divisible by p for $0 < j < p$.

Note: The polynomial $x^p - x \in \mathbb{F}_p[x]$ is not the zero polynomial but vanishes at all points $a \in \mathbb{F}_p$.

Exercise: Devise a method to compute inverses in $\mathbb{F}_p$ using FLT instead EEA.
Answer: if $a = 0$ then $a^{p-1} = 1$, so $a^{-1} = a^{p-2}$.

Univariate factorization over finite fields

$f \in \mathbb{F}_p[x]$ monic, $\deg f = n > 1$

- *Squarefree factorization*: $f = f_1^1 \cdots f_n^n$ such that f_i monic *squarefree* (i.e., $g^2 \nmid f_i$ for all nonconstant polynomials $g \in \mathbb{F}_p[x]$) and $\gcd(f_i, f_j) = 1$ for $i \neq j$.
- *Distinct-degree factorization*: g monic squarefree, $g = g_1 \cdots g_n$ such $h \mid g_i \implies \deg h = i$ for all nonconstant polynomials $h \in \mathbb{F}_p[x]$ (*equal-degree polynomial*).
- *Equal-degree factorization*: h monic squarefree equal-degree polynomial of degree $n = ki$, compute the monic irreducible factors $h_1, \ldots, h_k$ of degree i such that $h = h_1 \cdots h_k$.

Univariate factorization over finite fields

$f \in \mathbb{F}_p[x]$ monic, $\deg f = n > 1$

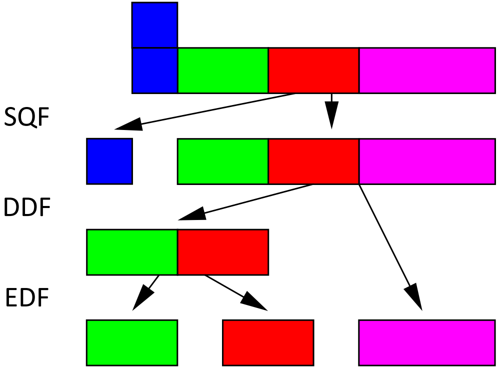
- *Squarefree factorization*: $f = f_1^1 \cdots f_n^n$ such that f_i monic *squarefree* (i.e., $g^2 \nmid f_i$ for all nonconstant polynomials $g \in \mathbb{F}_p[x]$) and $\gcd(f_i, f_j) = 1$ for $i \neq j$.
- *Distinct-degree factorization*: g monic squarefree, $g = g_1 \cdots g_n$ such $h \mid g_i \implies \deg h = i$ for all nonconstant polynomials $h \in \mathbb{F}_p[x]$ (*equal-degree polynomial*).
- *Equal-degree factorization*: h monic squarefree equal-degree polynomial of degree $n = ki$, compute the monic irreducible factors $h_1, \ldots, h_k$ of degree i such that $h = h_1 \cdots h_k$.

Univariate factorization over finite fields

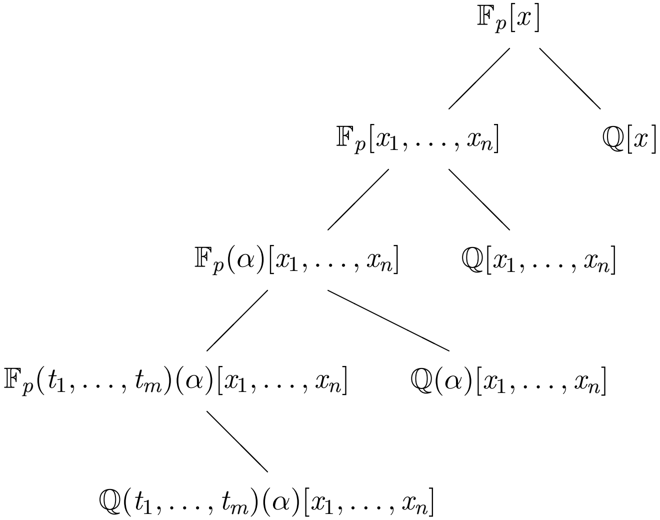
$f \in \mathbb{F}_p[x]$ monic, $\deg f = n > 1$

- *Squarefree factorization*: $f = f_1^1 \cdots f_n^n$ such that f_i monic *squarefree* (i.e., $g^2 \nmid f_i$ for all nonconstant polynomials $g \in \mathbb{F}_p[x]$) and $\gcd(f_i, f_j) = 1$ for $i \neq j$.
- *Distinct-degree factorization*: g monic squarefree, $g = g_1 \cdots g_n$ such $h \mid g_i \implies \deg h = i$ for all nonconstant polynomials $h \in \mathbb{F}_p[x]$ (*equal-degree polynomial*).
- *Equal-degree factorization*: h monic squarefree equal-degree polynomial of degree $n = ki$, compute the monic irreducible factors $h_1, \ldots, h_k$ of degree i such that $h = h_1 \cdots h_k$.

Meta-algorithm



Factorization in various domains



Squarefree factorization

F field (finite or not), $f \in F[x]$ monic with $\deg f = n > 1$ and squarefree decomposition $f = f_1^1 \cdots f_n^n$. (Also assume $n > p$ if $\mathbb{F}_p \subseteq F$.) Then

$$f' = \frac{\partial f}{\partial x} = f_1^0 \cdots f_n^{n-1} \cdot \underbrace{({f_1}' f_2 \cdots f_n + \ldots + n f_1 \cdots f_{n-1} {f_n}')}_{g}$$

The assumptions about the squarefree decomposition imply that $\gcd(f_i, g) = 1$ for all i, and therefore

$$\gcd(f, f') = f_1^0 \cdots f_n^{n-1}.$$

Let $h = f_1 \cdots f_n$, the *squarefree part* of f. Then

$$h' = {f_1}' f_2 \cdots f_n + \ldots + f_1 \cdots f_{n-1} {f_n}'$$

and

$$f_i = \gcd(h, g - ih')$$

Example

Let $f = x^4 + x^3 = x^3(x+1)$. Expect to find $f_1 = x + 1$, $f_3 = x$, and $f_2 = f_4 = 1$.

- $f' = 4x^3 + 3x^2$
- $\gcd(f, f') = x^2$
- $g = f' / \gcd(f, f') = 4x + 3$
- $h = f / \gcd(f, f') = x^2 + x$
- $h' = 2x + 1$
- $\gcd(h, g - h') = \gcd(x^2 + x, 2x + 2) = x + 1 = f_1$
- $\gcd(h, g - 2h') = \gcd(x^2 + x, 1) = 1 = f_2$
- $\gcd(h, g - 3h') = \gcd(x^2 + x, -2x) = x = f_3$
- $\gcd(h, g - 4h') = \gcd(x^2 + x, -4x - 1) = 1 = f_4$

Yun's algorithm

Input: $f \in F[x] \setminus \{0\}$ monic, $\deg f = n$
Output: Monic squarefree decomposition $f = f_1^1 \cdots f_n^n$

1. $g_0 \leftarrow \dfrac{f'}{\gcd(f, f')}, \quad h \leftarrow \dfrac{f}{\gcd(f, f')}$
2. **for** $i = 1, \ldots, n$ **do**
3. $\quad g_i \leftarrow g_{i-1} - h'$
4. $\quad f_i \leftarrow \gcd(h, g_i)$

6. **return** $f_1, f_2, \ldots, f_n$

Yun's algorithm

Input: $f \in F[x] \setminus \{0\}$ monic, $\deg f = n$
Output: Monic squarefree decomposition $f = f_1^1 \cdots f_n^n$

1. $g_0 \leftarrow \dfrac{f'}{\gcd(f, f')}, \quad h \leftarrow \dfrac{f}{\gcd(f, f')}$
2. **for** $i = 1, \ldots, n$ **while** $h \neq 1$ **do**
3. $\quad g_i \leftarrow g_{i-1} - h'$
4. $\quad f_i \leftarrow \gcd(h, g_i)$
5. $\quad h \leftarrow \dfrac{h}{f_i}, \quad g_i \leftarrow \dfrac{g_i}{f_i}$
6. **return** $f_1, f_2, \ldots, 1, 1$

Cost dominated by step 1: $O(n^2)$ classical / $O(\mathsf{M}(n) \log n)$ fast

Distinct-degree factorization

Fermat's Little Theorem: $x^p - x = \prod_{a \in \mathbb{F}_p} (x - a) = \prod_{\substack{w \text{ monic irreducible} \\ \deg w = 1}} w$

Generalization (Gauß): For $i \in \mathbb{N}$, $x^{p^i} - x = \prod_{\substack{w \text{ monic irreducible} \\ (\deg w) | i}} w$

Algorithm: Given monic squarefree $g \in \mathbb{F}_p[x]$,
for $i = 1, 2, \ldots$ compute $\gcd(x^{p^i} - x, g)$ and remove it from g

Gauß' DDF algorithm

Input: $g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$
Output: Monic distinct-degree decomposition $g = g_1 \cdots g_n$

1. $a_0 \leftarrow x$
2. **for** $i \geq 1$ **while** $\deg g \geq 2i$ **do**
3. $\quad a_i \leftarrow a_{i-1}^p \text{ rem } g \quad (= x^{p^i} \text{ rem } g)$
4. $\quad g_i \leftarrow \gcd(g, a_i - x)$
5. $\quad g \leftarrow \frac{g}{g_i}, \quad a_i \leftarrow a_i \text{ rem } g$
6. **if** $g = 1$
   then return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots\ldots\ldots\ldots, 1$
   else return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots, 1, g, 1, \ldots, 1$

Gauß' DDF algorithm

Input: $g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$
Output: Monic distinct-degree decomposition $g = g_1 \cdots g_n$

```
a_0 ← x
for i ≥ 1 while deg g ≥ 2i do
    a_i ← a_{i-1}^p rem g    (= x^{p^i} rem g)
    g_i ← gcd(g, a_i − x)
    g ← g / g_i,    a_i ← a_i rem g
if g = 1
then return g_1, g_2, ..., g_{i-1}, 1, ............., 1
else return g_1, g_2, ..., g_{i-1}, 1, ..., 1, g, 1, ..., 1
```

Gauß' DDF algorithm

Input: $g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$
Output: Monic distinct-degree decomposition $g = g_1 \cdots g_n$

1. $a_0 \leftarrow x$
2. **for** $i \geq 1$ **while** $\deg g \geq 2i$ **do**
3. $\quad a_i \leftarrow a_{i-1}^p \operatorname{rem} g \quad (= x^{p^i} \operatorname{rem} g)$
4. $\quad g_i \leftarrow \gcd(g, a_i - x)$
5. $\quad g \leftarrow \frac{g}{g_i}, \quad a_i \leftarrow a_i \operatorname{rem} g$
6. **if** $g = 1$
   then return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots\ldots\ldots\ldots, 1$
   else return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots, 1, g, 1, \ldots, 1$

Gauß' DDF algorithm

Input: $g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$
Output: Monic distinct-degree decomposition $g = g_1 \cdots g_n$

1. $a_0 \leftarrow x$
2. **for** $i \geq 1$ **while** $\deg g \geq 2i$ **do**
3. $\quad a_i \leftarrow a_{i-1}^p \text{ rem } g \quad (= x^{p^i} \text{ rem } g)$
4. $\quad g_i \leftarrow \gcd(g, a_i - x)$
5. $\quad g \leftarrow \dfrac{g}{g_i}, \quad a_i \leftarrow a_i \text{ rem } g$
6. **if** $g = 1$
   then return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots\ldots\ldots\ldots, 1$
   else return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots, 1, g, 1, \ldots, 1$

Gauß' DDF algorithm

Input: $g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$
Output: Monic distinct-degree decomposition $g = g_1 \cdots g_n$

1. $a_0 \leftarrow x$
2. **for** $i \geq 1$ **while** $\deg g \geq 2i$ **do**
3. $\quad a_i \leftarrow a_{i-1}^p \operatorname{rem} g \quad (= x^{p^i} \operatorname{rem} g)$
4. $\quad g_i \leftarrow \gcd(g, a_i - x)$
5. $\quad g \leftarrow \dfrac{g}{g_i}, \quad a_i \leftarrow a_i \operatorname{rem} g$
6. **if** $g = 1$
   then return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots\ldots\ldots\ldots, 1$
   else return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots, 1, g, 1, \ldots, 1$

Gauß' DDF algorithm

Input: $g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$
Output: Monic distinct-degree decomposition $g = g_1 \cdots g_n$

```
a_0 ← x
for i ≥ 1 while deg g ≥ 2i do
    a_i ← a_{i-1}^p rem g    (= x^{p^i} rem g)
    g_i ← gcd(g, a_i − x)
    g ← g / g_i,    a_i ← a_i rem g
if g = 1
then return g_1, g_2, ..., g_{i-1}, 1, ............., 1
else return g_1, g_2, ..., g_{i-1}, 1, ..., 1, g, 1, ..., 1
```

Gauß' DDF algorithm

Input: $g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$
Output: Monic distinct-degree decomposition $g = g_1 \cdots g_n$

1. $a_0 \leftarrow x$
2. **for** $i \geq 1$ **while** $\deg g \geq 2i$ **do**
3. $\quad a_i \leftarrow a_{i-1}^p \text{ rem } g \quad (= x^{p^i} \text{ rem } g)$
4. $\quad g_i \leftarrow \gcd(g, a_i - x)$
5. $\quad g \leftarrow \dfrac{g}{g_i}, \quad a_i \leftarrow a_i \text{ rem } g$
6. **if** $g = 1$
   then return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots\ldots\ldots\ldots, 1$
   else return $g_1, g_2, \ldots, g_{i-1}, 1, \ldots, 1, g, 1, \ldots, 1$

Step 2 loop invariant:

$$w \in \mathbb{F}_p[x] \text{ and } \deg w \geq 1 \text{ and } w \mid g \implies \deg w \geq i$$

Example

Let $g = x^6 + x^3 - x^2 - x = x(x+1)(x-1)(x^3+x+1) \in \mathbb{F}_7[x]$.
We expect to find $g_1 = x^3 - x$, $g_3 = x^3 + x + 1$, and
$g_2 = g_4 = g_5 = g_6 = 1$.

1 $a_0 \leftarrow x$

2 $i = 1$ and $\deg g = 6 \geq 2 \cdot 1$

3 $a_1 \leftarrow x^7 \text{ rem } x^6 + x^3 - x^2 - x = -x^4 + x^3 + x^2$

4 $g_1 \leftarrow \gcd(a_1 - x, x^6 + x^3 - x^2 = x) = x^3 - x$

5 $g \leftarrow \dfrac{x^6 + x^3 - x^2 - x}{x^3 - x} = x^3 + x + 1$,
$a_1 \leftarrow a_1 \text{ rem } x^3 + x + 1 = 2x^2 - 1$

2 $i = 2$ and $\deg g = 3 < 2 \cdot 2$

6 **return** $x^3 - x, 1, x^3 + x + 1, 1, 1, 1$

Best done using MAPLE or other CAS

Cost analysis

$g \in \mathbb{F}_p[x] \setminus \{0\}$ monic squarefree, $\deg g = n$

2 at most $\frac{n}{2}$ iterations of:

3 $a_i \leftarrow a_{i-1}^p \text{ rem } g$ using *square-and-multiply*:
$O(\mathsf{M}(n) \log p)$ / $O(n^2 \log p)$ classical

4 $g_i \leftarrow \gcd(a_i - x, g)$: $O(\mathsf{M}(n) \log n)$ / $O(n^2)$ classical

5 $g \leftarrow \dfrac{g}{g_i}$ and $a_i \leftarrow a_i \text{ rem } g$: $O(\mathsf{M}(n))$ / $O(n^2)$ classical

Total cost: $O(n\,\mathsf{M}(n) \log(np))$ / $O(n^3 \log p)$ classical

Worst case: input is irreducible

Modular root finding I

$p > 2$ odd prime,
$h = (x - a_1) \cdots (x - a_n) \in \mathbb{F}_p[x]$ with $n > 1$ and $a_i \neq a_j$ for $i \neq j$
Goal: find $a_1, \ldots, a_n$

Fermat's Little Theorem: For $c \in \mathbb{F}_p$,

$$0 = c^p - c = c(c^{\frac{p-1}{2}} - 1)(c^{\frac{p-1}{2}} + 1)$$

So either $c = 0$ or $c^{\frac{p-1}{2}} = 1$ or $c^{\frac{p-1}{2}} = -1$, with probabilities $\frac{1}{p}$ or $\frac{1}{2}(1 - \frac{1}{p})$, respectively.

Modular root finding II

$p > 2$ odd prime,
$h = (x - a_1) \cdots (x - a_n) \in \mathbb{F}_p[x]$ with $n > 1$ and $a_i \neq a_j$ for $i \neq j$

Choose $b \in \mathbb{F}_p[x]$ with $\deg b < n$ uniformly at random. By the uniqueness of Lagrange interpolation, $b(a_i)$ is a uniformly random element of $\mathbb{F}_p$ and independent of $b(a_j)$ for $i \neq j$. Thus $b(a_i)^{\frac{p-1}{2}} = 1$ with probability $\frac{1}{2}(1 - \frac{1}{p})$, and the probability that $b(a_i)^{\frac{p-1}{2}} = b(a_j)^{\frac{p-1}{2}}$ for all i, j is

$$\left(\frac{1}{p}\right)^n + 2\left(\frac{1}{2}\left(1 - \frac{1}{p}\right)\right)^n < 2^{1-n}\frac{1}{p} + 2^{1-n}\left(1 - \frac{n}{p}\right) < 2^{1-n} \leq \frac{1}{2}.$$

Modular root finding III

$p > 2$ odd prime, $b \in \mathbb{F}_p[x]$,
$h = (x - a_1) \cdots (x - a_n) \in \mathbb{F}_p[x]$ with $n > 1$ and $a_i \neq a_j$ for $i \neq j$

$$\begin{aligned}
\forall i \;\; b(a_i) = 0 \quad &\iff \quad b \operatorname{rem} h = 0 \\
&\iff \quad \gcd(h, b) = h, \\
\forall i \;\; b(a_i) \neq 0 \quad &\iff \quad \gcd(h, b) = 1, \\
\forall i \;\; b(a_i)^{\frac{p-1}{2}} = 1 \quad &\iff \quad (b^{\frac{p-1}{2}} - 1) \operatorname{rem} h = 0 \\
&\iff \quad \gcd(h, b^{\frac{p-1}{2}} - 1) = b^{\frac{p-1}{2}} - 1, \\
\forall i \;\; b(a_i)^{\frac{p-1}{2}} \neq 1 \quad &\iff \quad \gcd(h, b^{\frac{p-1}{2}} - 1) = 1.
\end{aligned}$$

Algorithm: Choose b with $\deg b < n$ at random and compute $\gcd(h, b)$ and $\gcd(h, b^{\frac{p-1}{2}} - 1)$. This will split h with probability $> \frac{1}{2}$. Recurse.

Examples

Let $h = x^3 - x \in \mathbb{F}_7[x]$.

- If $b \in \mathbb{F}_p$, then $\gcd(h, b) \in \{1, h\}$ and $\gcd(h, b^{\frac{p-1}{2}} - 1) \in \{1, h\}$.
- If $b \in \{x, x+1, x-1\}$, then $\gcd(h, b) = b$ splits h.
- If $b = x + 2$, then $\gcd(h, b) = 1$, $b^{\frac{p-1}{2}} = b^3 = x^3 - x^2 - 2x + 1$ and $\gcd(h, b^3 - 1) = x^2 + x$ splits h.
- If $b = x^2 + 1$, then $\gcd(h, b) = 1$, $b^{\frac{p-1}{2}} = b^3 = x^6 + 3x^4 + 3x^2 + 1$ and $\gcd(h, b^3 - 1) = h$.
- If $b = -x^2 - 1$, then $\gcd(h, b) = 1$, $b^{\frac{p-1}{2}} = b^3 = -x^6 - 3x^4 - 3x^2 - 1$ and $\gcd(h, b^3 - 1) = 1$.

Cantor-Zassenhaus algorithm

Input: $h \in \mathbb{F}_p[x]$ monic squarefree with all irreducible factors of degree 1, where $p > 2$ and $1 \leq n = \deg h$
Output: The monic irreducible factors $h_1, \ldots, h_{n/i}$ of h

1. **if** $n = 1$ **then return** h
2. Choose $b \in \mathbb{F}_p[x]$ with $0 < \deg b < n$ uniformly at random
3. $u \leftarrow \gcd(h, b)$
4. **if** $u \neq 1$ **then recurse** on both u and on $\frac{h}{u}$ and **return** the combined results
5. $v \leftarrow b^{\frac{p-1}{2}} \operatorname{rem} h$, $\quad v \leftarrow \gcd(h, v)$
6. **if** $v \in \{1, h\}$ **then** go back to step 2 and repeat
7. **recurse** on both v and on $\frac{h}{v}$ and **return** the combined results

Cantor-Zassenhaus algorithm

Root finding algorithm generalizes to equal-degree factorization

Input: $h \in \mathbb{F}_p[x]$ monic squarefree with all irreducible factors of degree $\boldsymbol{i}$, where $p > 2$ and $1 \le \boldsymbol{i} \mid \boldsymbol{n} = \deg h$
Output: The monic irreducible factors $h_1, \ldots, h_{n/i}$ of h

1. **if** $\boldsymbol{n} = \boldsymbol{i}$ **then return** h
2. Choose $b \in \mathbb{F}_p[x]$ with $0 < \deg b < n$ uniformly at random
3. $u \leftarrow \gcd(h, b)$
4. **if** $u \ne 1$ **then recurse** on both u and on $\frac{h}{u}$ and **return** the combined results
5. $v \leftarrow b^{\frac{\boldsymbol{p^i - 1}}{\boldsymbol{2}}} \text{ rem } h, \quad v \leftarrow \gcd(h, v)$
6. **if** $v \in \{1, h\}$ **then** go back to step 2 and repeat
7. **recurse** on both v and on $\frac{h}{v}$ and **return** the combined results

Cost analysis

$p > 2$ prime, $h \in \mathbb{F}_p[x]$ monic squarefree with all irreducible factors of degree i, where $1 \leq i \mid n = \deg h$
(similar algorithm for $p = 2$ exists)

3 $\gcd(h, b)$: $O(\mathsf{M}(n) \log n)$ / $O(n^2)$ classical
5 $v \leftarrow b^{\frac{p^i-1}{2}} \operatorname{rem} h$ via square-and-multiply:
$O(i\,\mathsf{M}(n) \log p)$ / $O(in^2 \log p)$ classical
$\gcd(h, v)$: $O(\mathsf{M}(n) \log n)$ / $O(n^2)$ classical
6 Expected number of iterations: ≤ 2
7 Expected recursion depth: $O(\log \frac{n}{i})$

Expected total cost: $O(i\,\mathsf{M}(n) \log(np))$ / $O(in^2 \log(np))$ classical

Worst case: $i = \frac{n}{2}$

Probabilistic vs deterministic EDF

- There is no known deterministic algorithm for equal-degree factorization that runs in time polyonmial in n and $\log p$.
- In fact, there is no known deterministic polynomial time algorithm for factoring $x^2 - a$, i.e., computing $\sqrt{a} \in \mathbb{F}_p$, if $a \in \mathbb{F}_p$ is a square and $4 \mid (p-1)$.
- Quest for deterministic polynomial time factoring is of purely theoretical interest; the probabilistic algorithms are highly efficient in practice.

Special case: root finding

Input: $f \in \mathbb{F}_p[x] \setminus \{0\}$ monic, where $p > 2$ and $\deg f = n < p$
Output: the distinct roots $a_1, \ldots, a_r \in \mathbb{F}_p$ of f

1. $g \leftarrow \dfrac{f}{\gcd(f, f')}$
2. $a \leftarrow x^p \text{ rem } g$
3. $h \leftarrow \gcd(a - x, g)$
4. **call** the Cantor-Zassenhaus algorithm with input h and $i - 1$ and **return** its result

(Expected) cost: $O(\mathsf{M}(n) \log(pn))$ / $O(n^2 \log p)$ classical

Special case: irreducibility test

Input: $f \in \mathbb{F}_p[x] \setminus \{0\}$ monic, where $p > 2$ and $\deg f = n$
Output: *true* if f is irreducible and *false* otherwise

1. **if** $\gcd(f, f') \neq 1$ **then return** *false*
2. **if** $x^{p^n} \text{ rem } f \neq x$ **then return** *false*
3. **for** every prime divisor $d \in \mathbb{N}$ of n **do**
4. $\quad a_d \leftarrow x^{p^{n/d}} \text{ rem } f$
5. $\quad$ **if** $\gcd(a_d - x, f) \neq 1$ **then return** *false*
6. **return** *true*

Cost: $O(n\,\mathsf{M}(n)\log(pn))$ / $O(n^3 \log p)$ classical

State of the art: factoring in $\mathbb{F}_p[x]$

SQF + DDF + EDF	arithmetic RAM	word RAM
classical Cantor & Zassenhaus	$n^3 \log p$	$n^3 \log^3 p$
fast Cantor & Zassenhaus	$n^2 \log p$	$n^2 \log^2 p$
von zur Gathen & Shoup	$n^2 + n \log p$	$n^2 \log p + n \log^2 p$
Kaltofen & Shoup	$n^{1.815} \log^{0.407} p$	$n^{1.815} \log^{1.407} p$
Kedlaya & Umans		$n^{1.5} \log p + n \log^2 p$

Ignoring constants and factors $\log n$ and $\operatorname{loglog} p$

Main ingredients:
blocking strategy and fast *modular composition* $g(h) \operatorname{rem} f$

Outline

From $\mathbb{F}_p[x]$ to $\mathbb{Q}[x]$

$f \in \mathbb{Q}[x]$ monic nonconstant squarefree

Main idea:

- Choose ``small'' prime $p > 2$ and factor f in $\mathbb{F}_p[x]$
- *Lift* the factorization to one modulo p^k for k large enough
- Combine some modular factors to obtain factors in $\mathbb{Q}[x]$

- Need to choose a ``good'' prime p such that it does not divide the denominator of f and such that f remains squarefree in $\mathbb{F}_p[x]$
- How large is ``large enough''?
- How to determine the denominators of the factors?

From $\mathbb{F}_p[x]$ to $\mathbb{Q}[x]$

$f \in \mathbb{Q}[x]$ monic nonconstant squarefree

Main idea:

- Choose ``small'' prime $p > 2$ and factor f in $\mathbb{F}_p[x]$
- *Lift* the factorization to one modulo p^k for k large enough
- Combine some modular factors to obtain factors in $\mathbb{Q}[x]$

- Need to choose a ``good'' prime p such that it does not divide the denominator of f and such that f remains squarefree in $\mathbb{F}_p[x]$
- How large is ``large enough''?
- How to determine the denominators of the factors?

From $\mathbb{F}_p[x]$ to $\mathbb{Q}[x]$

$f \in \mathbb{Q}[x]$ monic nonconstant squarefree

Main idea:

- Choose ``small'' prime $p > 2$ and factor f in $\mathbb{F}_p[x]$
- *Lift* the factorization to one modulo p^k for k large enough
- Combine some modular factors to obtain factors in $\mathbb{Q}[x]$

- Need to choose a ``good'' prime p such that it does not divide the denominator of f and such that f remains squarefree in $\mathbb{F}_p[x]$
- How large is ``large enough''?
- How to determine the denominators of the factors?

From $\mathbb{F}_p[x]$ to $\mathbb{Q}[x]$

$f \in \mathbb{Q}[x]$ monic nonconstant squarefree

Main idea:

- Choose ``small'' prime $p > 2$ and factor f in $\mathbb{F}_p[x]$
- *Lift* the factorization to one modulo p^k for k large enough
- Combine some modular factors to obtain factors in $\mathbb{Q}[x]$

Remarks:

- Need to choose a ``good'' prime p such that it does not divide the denominator of f and such that f remains squarefree in $\mathbb{F}_p[x]$
- How large is ``large enough''?
- How to determine the denominators of the factors?

From $\mathbb{F}_p[x]$ to $\mathbb{Q}[x]$

$f \in \mathbb{Q}[x]$ monic nonconstant squarefree

Main idea:

- Choose ``small'' prime $p > 2$ and factor f in $\mathbb{F}_p[x]$
- *Lift* the factorization to one modulo p^k for k large enough
- Combine some modular factors to obtain factors in $\mathbb{Q}[x]$

Remarks:

- Need to choose a ``good'' prime p such that it does not divide the denominator of f and such that f remains squarefree in $\mathbb{F}_p[x]$
- How large is ``large enough''?
- How to determine the denominators of the factors?

From $\mathbb{F}_p[x]$ to $\mathbb{Q}[x]$

$f \in \mathbb{Q}[x]$ monic nonconstant squarefree

Main idea:

- Choose ``small'' prime $p > 2$ and factor f in $\mathbb{F}_p[x]$
- *Lift* the factorization to one modulo p^k for k large enough
- Combine some modular factors to obtain factors in $\mathbb{Q}[x]$

Remarks:

- Need to choose a ``good'' prime p such that it does not divide the denominator of f and such that f remains squarefree in $\mathbb{F}_p[x]$
- How large is ``large enough''?
- How to determine the denominators of the factors?

Hensel's lemma I

$f, g, h \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators.
Notation: write $f \equiv gh \bmod p$ to mean $f = gh$ in $\mathbb{F}_p[x]$, more precisely: $p \mid (f - gh)$.

Hensel's lemma If $\gcd(g, h) = 1$ in $\mathbb{F}_p[x]$, then for any $k \in \mathbb{N}$ there exist monic $g_k, h_k \in \mathbb{Q}[x]$ such that

$$g_k \equiv g \bmod p, \quad h_k \equiv h \bmod p, \text{ and } f \equiv gh \bmod p^k.$$

Moreover, g_k and h_k are unique modulo p^k.

Proof: Induction on k.

Hensel's lemma II

$k \in \mathbb{N}$, $f, g_k, h_k \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_k, h_k) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_k h_k \bmod p^k$.

Construction of g_{k+1}, h_{k+1}:

1. $e_k = f - g_k h_k$
2. EEA computes $s, t \in \mathbb{Z}[x]$ such that $sg_k + th_k = 1$ in $\mathbb{F}_p[x]$
3. $\bar{g} = g_k + te_k$ and $\bar{h} = h_k + se_k$

Hensel's lemma II

$k \in \mathbb{N}$, f, g_k, $h_k \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_k, h_k) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_k h_k \mod p^k$.

Construction of g_{k+1}, h_{k+1}:

1. $e_k = f - g_k h_k$
2. EEA computes $s, t \in \mathbb{Z}[x]$ such that $sg_k + th_k = 1$ in $\mathbb{F}_p[x]$
3. $\bar{g} = g_k + te_k$ and $\bar{h} = h_k + se_k$

Hensel's lemma II

$k \in \mathbb{N}$, f, g_k, $h_k \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_k, h_k) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_k h_k \mod p^k$.

Construction of g_{k+1}, h_{k+1}:

1. $e_k = f - g_k h_k$
2. EEA computes $s, t \in \mathbb{Z}[x]$ such that $sg_k + th_k = 1$ in $\mathbb{F}_p[x]$
3. $\bar{g} = g_k + te_k$ and $\bar{h} = h_k + se_k$

Then

$$f - \bar{g}\bar{h} = f - g_k h_k - g_k s e_k - h_k t e_k - ste_k^2 = (1 - sg_k - th_k)\,e_k - ste_k^2$$

By assumption, $p^k \mid e$ and $p \mid 1 - sg_k - th_k$, and hence $p^{k+1} \mid (f - \bar{g}\bar{h})$.

Example

$k = 1$, $f = x^3 + 14x^2 + 15x + 26$, $g_1 = x + 1$, $h_1 = x^2 + x + 2$, $p = 3$

1. $$\begin{aligned} e_1 &= x^3 + 14x^2 + 15x + 26 - (x+1)(x^2 + x + 2) \\ &= x^3 + 14x^2 + 15x + 26 - (x^3 + 2x^2 + 3x + 2) \\ &= 12x^2 + 12x + 24 = 3 \cdot (4x^2 + 4x + 8) \end{aligned}$$
2. $s = x$ and $t = 2$ work:
   $x \cdot (x+1) + 2 \cdot (x^2 + x + 2) = 3x^2 + 3x + 4 \equiv 1 \bmod 3$
3. $\bar{g} = (x+1) + 2 \cdot (12x^2 + 12x + 24) = 24x^2 + 25x + 49$,
   $\bar{h} = (x^2 + x + 2) + x \cdot (12x^2 + 12x + 24) = 12x^3 + 13x^2 + 25x + 2$

Check:

$$\begin{aligned} f - \bar{g}\bar{h} &= x^3 + 14x^2 + 15x + 26 \\ &\quad -(288x^5 + 612x^4 + 1513x^3 + 1310x^2 + 1275x + 98) \\ &= -9 \cdot (32x^5 + 68x^4 + 168x^3 + 144x^2 + 140x + 8) \end{aligned}$$

Issues

- $\bar{g}, \bar{h}$ are not monic and their degrees are too high.
  Resolution:

$$\begin{aligned} q_k &= te_k \text{ quo } g_k, \\ g_{k+1} &= g_k + (te_k \text{ rem } g_k) = g_k + te_k - q_k g_k, \\ h_{k+1} &= h_k + se_k + q_k h_k. \end{aligned}$$

  Then g_{k+1}, h_{k+1} are monic, $\deg g_{k+1} = \deg g_k$, $\deg h_{k+1} = \deg h_k$, and $g_{k+1}h_{k+1} \equiv \bar{g}\bar{h} \bmod p^{k+1}$.
- Coefficient growth.
  Resolution: reduce coefficients $\bmod\, p^{k+1}$

Issues

- $\bar{g}, \bar{h}$ are not monic and their degrees are too high.
  Resolution:

$$\begin{aligned} q_k &= te_k \text{ quo } g_k, \\ g_{k+1} &= g_k + (te_k \text{ rem } g_k) = g_k + te_k - q_k g_k, \\ h_{k+1} &= h_k + se_k + q_k h_k. \end{aligned}$$

  Then g_{k+1}, h_{k+1} are monic, $\deg g_{k+1} = \deg g_k$, $\deg h_{k+1} = \deg h_k$, and $g_{k+1} h_{k+1} \equiv \bar{g}\bar{h} \bmod p^{k+1}$.
- Coefficient growth.
  Resolution: reduce coefficients $\bmod\, p^{k+1}$

Example (continued)

$k = 1, f = x^3 + 14x^2 + 15x + 26,\ g_1 = x + 1,\ h_1 = x^2 + x + 2,\ p = 3$

1. $$\begin{aligned} e_1 &= x^3 + 14x^2 + 15x + 26 - (x+1)(x^2 + x + 2) \\ &= x^3 + 14x^2 + 15x + 26 - (x^3 + 2x^2 + 3x + 2) \\ &= 12x^2 + 12x + 24 \equiv \mathbf{3x^2 + 3x + 6 \bmod 3^2} \end{aligned}$$
2. $s = x$ and $t = 2$ work:
   $x \cdot (x+1) + 2 \cdot (x^2 + x + 2) = 3x^2 + 3x + 4 \equiv 1 \bmod 3$
3. $\mathbf{q_1 = 2 \cdot (3x^2 + 3x + 6) \text{ quo } x + 1 = 6x}$,
   $$\begin{aligned} g_2 &= (x+1) + 2 \cdot (3x^2 + 3x + 6) \mathbf{- 6x \cdot (x+1)} \\ &= x + 13 \equiv \mathbf{x + 4 \bmod 3^2}, \\ h_2 &= (x^2 + x + 2) + x \cdot (3x^2 + 3x + 6) \mathbf{+ 6x \cdot (x^2 + x + 2)} \\ &= 9x^3 + 10x^2 + 19x + 2 \equiv \mathbf{x^2 + x + 2 \bmod 3^2} \end{aligned}$$

Check: $$\begin{aligned} f - g_2 h_2 &= x^3 + 14x^2 + 15x + 26 - (x+4)(x^2 + x + 2) \\ &= x^3 + 14x^2 + 15x + 26 - (x^3 + 5x^2 + 6x + 8) \\ &= 9x^2 + 9x + 18 \end{aligned}$$

Hensel lifting

Input: $k \in \mathbb{N}$, $f, g_1, h_1 \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_1, h_1) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_1 h_1 \bmod p$
Output: $g_k, h_k \in \mathbb{Q}[x]$ monic, $g_k \equiv g_1 \bmod p$, $h_k \equiv h_1 \bmod p$, and $f \equiv g_k h_k \bmod p^k$

1. **call** EEA to compute $s, t \in \mathbb{Z}[x]$ such that $sg_1 + th_1 = 1$ in $\mathbb{F}_p[x]$
2. **for** $i = 1, \ldots, k-1$ **do**
3. $\quad e_i \leftarrow (f - g_i h_i) \text{ rem } p^{i+1}$
4. $\quad q_i \leftarrow (te_i \text{ quo } g_i) \text{ rem } p^{i+1}$
5. $\quad g_{i+1} \leftarrow (g_i + te_i - q_i g_i) \text{ rem } p^{i+1}$
6. $\quad h_{i+1} \leftarrow (h_i + se_i + q_i h_i) \text{ rem } p^{i+1}$
7. **return** g_k, h_k

Hensel lifting

Input: $k \in \mathbb{N}$, $f, g_1, h_1 \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_1, h_1) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_1 h_1 \bmod p$
Output: $g_k, h_k \in \mathbb{Q}[x]$ monic, $g_k \equiv g_1 \bmod p$, $h_k \equiv h_1 \bmod p$, and $f \equiv g_k h_k \bmod p^k$

```
call EEA to compute s, t ∈ Z[x] such that s g_1 + t h_1 = 1 in F_p[x]
for i = 1, ..., k − 1 do
    e_i ← (f − g_i h_i) rem p^{i+1}
    q_i ← (t e_i quo g_i) rem p^{i+1}
    g_{i+1} ← (g_i + t e_i − q_i g_i) rem p^{i+1}
    h_{i+1} ← (h_i + s e_i + q_i h_i) rem p^{i+1}
return g_k, h_k
```

Hensel lifting

Input: $k \in \mathbb{N}$, $f, g_1, h_1 \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_1, h_1) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_1 h_1 \bmod p$
Output: $g_k, h_k \in \mathbb{Q}[x]$ monic, $g_k \equiv g_1 \bmod p$, $h_k \equiv h_1 \bmod p$, and $f \equiv g_k h_k \bmod p^k$

```
call EEA to compute s, t ∈ Z[x] such that s g_1 + t h_1 = 1 in F_p[x]
for i = 1, ..., k − 1 do
    e_i ← (f − g_i h_i) rem p^{i+1}
    q_i ← (t e_i quo g_i) rem p^{i+1}
    g_{i+1} ← (g_i + t e_i − q_i g_i) rem p^{i+1}
    h_{i+1} ← (h_i + s e_i + q_i h_i) rem p^{i+1}
return g_k, h_k
```

Hensel lifting

Input: $k \in \mathbb{N}$, $f, g_1, h_1 \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_1, h_1) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_1 h_1 \bmod p$
Output: $g_k, h_k \in \mathbb{Q}[x]$ monic, $g_k \equiv g_1 \bmod p$, $h_k \equiv h_1 \bmod p$, and $f \equiv g_k h_k \bmod p^k$

```
call EEA to compute s, t ∈ ℤ[x] such that s g_1 + t h_1 = 1 in 𝔽_p[x]
for i = 1, ..., k − 1 do
    e_i ← (f − g_i h_i) rem p^{i+1}
    q_i ← (t e_i quo g_i) rem p^{i+1}
    g_{i+1} ← (g_i + t e_i − q_i g_i) rem p^{i+1}
    h_{i+1} ← (h_i + s e_i + q_i h_i) rem p^{i+1}
return g_k, h_k
```

Hensel lifting

Input: $k \in \mathbb{N}$, $f, g_1, h_1 \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_1, h_1) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_1 h_1 \bmod p$
Output: $g_k, h_k \in \mathbb{Q}[x]$ monic, $g_k \equiv g_1 \bmod p$, $h_k \equiv h_1 \bmod p$, and $f \equiv g_k h_k \bmod p^k$

```
call EEA to compute s, t ∈ Z[x] such that sg_1 + th_1 = 1 in F_p[x]
for i = 1, ..., k − 1 do
    e_i ← (f − g_i h_i) rem p^{i+1}
    q_i ← (t e_i quo g_i) rem p^{i+1}
    g_{i+1} ← (g_i + t e_i − q_i g_i) rem p^{i+1}
    h_{i+1} ← (h_i + s e_i + q_i h_i) rem p^{i+1}
return g_k, h_k
```

Hensel lifting

Input: $k \in \mathbb{N}$, $f, g_1, h_1 \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_1, h_1) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_1 h_1 \bmod p$
Output: $g_k, h_k \in \mathbb{Q}[x]$ monic, $g_k \equiv g_1 \bmod p$, $h_k \equiv h_1 \bmod p$, and $f \equiv g_k h_k \bmod p^k$

call EEA to compute $s, t \in \mathbb{Z}[x]$ such that $sg_1 + th_1 = 1$ in $\mathbb{F}_p[x]$
for $i = 1, \ldots, k-1$ **do**
    $e_i \leftarrow (f - g_i h_i) \text{ rem } p^{i+1}$
    $q_i \leftarrow (te_i \text{ quo } g_i) \text{ rem } p^{i+1}$
    $g_{i+1} \leftarrow (g_i + te_i - q_i g_i) \text{ rem } p^{i+1}$
    $h_{i+1} \leftarrow (h_i + se_i + q_i h_i) \text{ rem } p^{i+1}$
return g_k, h_k

Hensel lifting

Input: $k \in \mathbb{N}$, $f, g_1, h_1 \in \mathbb{Q}[x]$ monic, $p \in \mathbb{N}$ prime not dividing any denominators, $\gcd(g_1, h_1) = 1$ in $\mathbb{F}_p[x]$, and $f \equiv g_1 h_1 \bmod p$
Output: $g_k, h_k \in \mathbb{Q}[x]$ monic, $g_k \equiv g_1 \bmod p$, $h_k \equiv h_1 \bmod p$, and $f \equiv g_k h_k \bmod p^k$

call EEA to compute $s, t \in \mathbb{Z}[x]$ such that $sg_1 + th_1 = 1$ in $\mathbb{F}_p[x]$
for $i = 1, \dots, k-1$ **do**
    $e_i \leftarrow (f - g_i h_i) \text{ rem } p^{i+1}$
    $q_i \leftarrow (te_i \text{ quo } g_i) \text{ rem } p^{i+1}$
    $g_{i+1} \leftarrow (g_i + te_i - q_i g_i) \text{ rem } p^{i+1}$
    $h_{i+1} \leftarrow (h_i + se_i + q_i h_i) \text{ rem } p^{i+1}$
return g_k, h_k

Example (continued)

$i = 2$, $f = x^3 + 14x^2 + 15x + 26$, $g_2 = x + 4$, $h_2 = x^2 + x + 2$, $p = 3$

1. $e_2 = x^3 + 14x^2 + 15x + 26 - (x+4)(x^2 + x + 2) = 9x^2 + 9x + 18$
2. $s = x$ and $t = 2$ still work:
   $x \cdot (x + 4) + 2 \cdot (x^2 + x + 2) = 3x^2 + 6x + 4 \equiv 1 \bmod 3$
3. $q_2 = 2 \cdot (9x^2 + 9x + 18)$ quo $x + 4 = 18x - 54 \equiv 18x \bmod 3^3$,
   $g_3 = (x + 4) + 2 \cdot (9x^2 + 9x + 18) - 18x \cdot (x + 4)$
   $= -55x + 40 \equiv x + 13 \bmod 3^3$,
   $h_3 = (x^2 + x + 2) + x \cdot (9x^2 + 9x + 18) + 18x \cdot (x^2 + x + 2)$
   $= 27x^3 + 28x^2 + 54x + 2 \equiv x^2 + x + 2 \bmod 3^3$

Check: $f - g_3 h_3 = x^3 + 14x^2 + 15x + 26 - (x + 13)(x^2 + x + 2) = 0$

Example (continued)

$i = 2$, $f = x^3 + 14x^2 + 15x + 26$, $g_2 = x + 4$, $h_2 = x^2 + x + 2$, $p = 3$

1. $e_2 = x^3 + 14x^2 + 15x + 26 - (x+4)(x^2+x+2) = 9x^2 + 9x + 18$
2. $s = x$ and $t = 2$ still work:
   $x \cdot (x+4) + 2 \cdot (x^2 + x + 2) = 3x^2 + 6x + 4 \equiv 1 \bmod 3$
3. $q_2 = 2 \cdot (9x^2 + 9x + 18)$ quo $x + 4 = 18x - 54 \equiv 18x \bmod 3^3$,
   $g_3 = (x+4) + 2 \cdot (9x^2 + 9x + 18) - 18x \cdot (x+4)$
   $= -55x + 40 \equiv x + 13 \bmod 3^3$,
   $h_3 = (x^2 + x + 2) + x \cdot (9x^2 + 9x + 18) + 18x \cdot (x^2 + x + 2)$
   $= 27x^3 + 28x^2 + 54x + 2 \equiv x^2 + x + 2 \bmod 3^3$

Check: $f - g_3 h_3 = x^3 + 14x^2 + 15x + 26 - (x+13)(x^2+x+2) = 0$
(In general, another stage is required after Hensel lifting.)

Cost analysis

$p \in \mathbb{N}$ prime, $k \in \mathbb{N}$, $f \in \mathbb{Q}[x]$ monic nonconstant, $\deg f = n$, numerators and denominators of f absolutely bounded by p^k, $g_1, h_1 \in \mathbb{Z}[x]$ with coefficients in $\{0, \ldots, p-1\}$

Counting word operations:

1. $O(nk \log^2 p)$ to reduce all coefficients of f modulo p
   EEA: $O(n^2 \log^2 p)$
2. $k-1$ iterations of:
3. $(f - g_i h_i)$ rem p^{i+1}: $O(n^2 k^2 \log^2 p)$
4. $(te_i$ quo $g_i)$ rem p^{i+1}: $O(n^2 k^2 \log^2 p)$
5. $(g_i + te_i - q_i g_i)$ rem p^{i+1}: $O(n^2 k^2 \log^2 p)$
6. $(h_i + se_i + q_i h_i)$ rem p^{i+1}: $O(n^2 k^2 \log^2 p)$

Total cost: $O(n^2 k^3 \log^2 p)$

Quadratic Hensel lifting

Main ingredient: lift from p^i tp p^{2i} in one step by also lifting s and t

Total cost:

- $O(\mathsf{M}(n)\log n \cdot \mathsf{M}(\log p) + n \cdot \mathsf{M}(\log p)\,\mathrm{loglog}\, p)$ for the EEA in $\mathbb{F}_p[x]$ and
- $O(\mathsf{M}(n)\mathsf{M}(k \log p)$ for the main loop, which is dominated by the cost for the last iteration

Ignoring constant and logarithmic factors, this corresponds to $nk \log p$ word operations, vs $n^2k^3 \log p$ for the classical Hensel lifting algorithm.

Factors with negative integer coefficients

Hensel lifting to order k will always produce factors with nonnegative integer coefficients less than p^k.

Solution: When reducing modulo p^k, use *symmetric* coefficients in $\{-\frac{p^k-1}{2}, \ldots, \frac{p^k-1}{2}\}$ instead of nonnegative coefficients in $\{0, \ldots, p^k - 1\}$.

Example: $x^3 + 14x^2 + 15x + 26 \equiv x^3 - 13x^2 - 12x - 1 \bmod 3^3$

Factors with rational coefficients

$f \in \mathbb{Q}[x]$ with $f \equiv f_1 \cdots f_r \bmod p^k$
Determine common denominator $d \in \mathbb{N}$ such that $df \in \mathbb{Z}[x]$
Let $g_i = df_i \operatorname{rem} p^k$ for $1 \leq i \leq r$. Then $d^r f \equiv g_1 \cdots g_r \bmod p^k$.

If $f = h_1 \cdots h_r$ is the monic irreducible factorization in $\mathbb{Q}[x]$, then also $f \equiv h_1 \cdots h_r \bmod p^k$, and the uniqueness of factorization modulo p and of Hensel lifting implies that $h_i \equiv \frac{g_i}{d} \equiv f_i \bmod p^k$, for all i, up to reordering.

Thus $g_i \equiv dh_i \bmod p^k$. It follows from a Lemma by Gauß that $dh_i \in \mathbb{Z}[x]$, and if k is large enough, then $|g - dh_i| < p^k$, which implies that $g = dh_i$ for all i.

Example

$f = x^2 - \frac{3}{2}x - 1$, $p = 3$, $k = 2$
Then $f \equiv (x+1)(x-1) \bmod 3$ and Hensel lifting yields
$f \equiv f_1 f_2 \equiv (x-2)(x+4) \bmod 3^2$.
Choosing $d = 2$, we obtain $g_1 = 2(x-2) \equiv 2x - 4 \bmod 3^2$ and
$g_2 = 2(x+4) \equiv 2x - 1 \bmod 3^2$.
Indeed, the factors of f in $\mathbb{Q}[x]$ are $h_1 = x - 2 = \frac{g_1}{2}$ and
$h_2 = x + \frac{1}{2} = \frac{g_2}{2}$.

This is not the most efficient solution for rational coefficients; a better way is to use *rational number reconstruction* (the equivalent of Padé approximation in $\mathbb{Z}$, using the EEA)

How large is large enough?

$f, f_1, \ldots, f_r \in \mathbb{Z}[x]$ nonconstant squarefree, $\deg f = n$, $f = f_1 \cdots f_r$

Mignotte's factor bound: $\|f_i\|_\infty \leq \sqrt{n+1} \cdot 2^n \|f\|_\infty$ for $1 \leq i \leq r$

Corollary: If

- $p \in \mathbb{N}$ prime and $k = 1 + \lfloor \log_p(\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
- $g_1, \ldots, g_r \in \mathbb{Z}[x]$ with symmetric coefficients such that $f \equiv g_1 \cdots g_r \bmod p^k$

Then $\frac{f_i}{\mathrm{lc}(f_i)} = \frac{g_i}{\mathrm{lc}(g_i)}$ for $1 \leq i \leq k$, up to reordering.

Swinnerton-Dyer polynomials

Can irreducible $f \in \mathbb{Q}[x]$ be reducible in $\mathbb{F}_p[x]$?

Yes, this is quite common. Actually, there are examples that are reducible modulo *every* prime $p \in \mathbb{N}$:

$f = x^4 + 1$; its complex roots are the primitive 8th roots of unity $\varphi = e^{i\pi/4}, \varphi^3, \varphi^5, \varphi^7$. Note that $\varphi^2 = i = \sqrt{-1}$ and $\varphi + \varphi^7 = \sqrt{2}$.

- $p = 2$: $f \equiv (x + 1)^4 \bmod 2$
- $4 \mid (p - 1)$: then there exists $a \in \mathbb{F}_p[x]$ such that $a^2 = -1$. Thus $f \equiv (x^2 + a)(x^2 - a) \bmod p$
- $4 \nmid (p - 1)$: then either 2 or -2 is a square in $\mathbb{F}_p$. In the first case, $f \equiv (x^2 + bx + 1)(x^2 - bx + 1) \bmod p$, where $b^2 = 2$, and similarly in the second case.

Factor combination

p prime, $f \equiv g_1 \cdots g_s \bmod p^k$, $k \in \mathbb{N}$ large enough

Irreducible factor f_i of f in $\mathbb{Q}[x]$ may split into a $\prod_{j \in S} g_j$ for some subset $S \subset \{1, \ldots, s\}$

Factor combination: Try all possible such subsets S until all factors of f in $\mathbb{Q}[x]$ are found.
Worst case: f irreducible in $\mathbb{Q}[x] \rightarrow 2^s - 2$ trials

More efficient polynomial-time methods based on *lattice reduction* exist (Lenstra, Lenstra & Lovacz, van Hoeij, ...)

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow d \prod_{j \in S} g_j \operatorname{rem} p^k$, $v \leftarrow d \prod_{j \notin S} g_j \operatorname{rem} p^k$
   (using symmetric coefficients)
8. $\quad$ **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow d \prod_{j \in S} g_j \operatorname{rem} p^k$, $v \leftarrow d \prod_{j \notin S} g_j \operatorname{rem} p^k$
   (using symmetric coefficients)
8. $\quad$ **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1}\|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow d \prod_{j \in S} g_j \text{ rem } p^k$, $v \leftarrow d \prod_{j \notin S} g_j \text{ rem } p^k$
   (using symmetric coefficients)
8. $\quad$ **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow d \prod_{j \in S} g_j \mathrm{~rem~} p^k$, $v \leftarrow d \prod_{j \notin S} g_j \mathrm{~rem~} p^k$ (using symmetric coefficients)
8. $\quad$ **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow d \prod_{j \in S} g_j \operatorname{rem} p^k$, $v \leftarrow d \prod_{j \notin S} g_j \operatorname{rem} p^k$
   (using symmetric coefficients)
8. $\quad$ **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \dots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \dots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \dots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \dots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $u \leftarrow d \prod_{j \in S} g_j \ \mathrm{rem}\ p^k$, $v \leftarrow d \prod_{j \notin S} g_j \ \mathrm{rem}\ p^k$
   (using symmetric coefficients)
8. **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow d \prod_{j \in S} g_j \operatorname{rem} p^k$, $v \leftarrow d \prod_{j \notin S} g_j \operatorname{rem} p^k$ (using symmetric coefficients)
8. $\quad$ **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$
Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
$k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
call Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
$T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
for all subsets $S \subset T$ by increasing cardinality **do**
  $u \leftarrow d \prod_{j \in S} g_j \ \mathrm{rem}\ p^k$, $v \leftarrow d \prod_{j \notin S} g_j \ \mathrm{rem}\ p^k$ (using symmetric coefficients)
  if $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
return L

Zassenhaus' algorithm

Input: $f \in \mathbb{Z}[x]$ squarefree of degree $n > 0$, $d = \mathrm{lc}(f)$

Output: $\{f_1, \ldots, f_r\} \subset \mathbb{Q}[x]$, monic irreducible, with $f = df_1 \cdots f_r$

1. Choose a prime $p \in \mathbb{N}$ not dividing d and such that f remains squarefree in $\mathbb{F}_p[x]$
2. $k \leftarrow 1 + \lfloor \log_p(d\sqrt{n+1} \cdot 2^{n+1} \|f\|_\infty) \rfloor$
3. Factor f modulo p, yielding monic $h_1, \ldots, h_s \in \mathbb{Z}[x]$ with $f \equiv dh_1 \cdots h_s \bmod p$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in \mathbb{Z}[x]$ with $f \equiv dg_1 \cdots g_s \bmod p^k$
5. $T \leftarrow \{1, \ldots, s\}, \quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow d \prod_{j \in S} g_j \operatorname{rem} p^k$, $v \leftarrow d \prod_{j \notin S} g_j \operatorname{rem} p^k$ (using symmetric coefficients)
8. $\quad$ **if** $df = uv$ in $\mathbb{Z}[x]$ **then** $T \leftarrow T \setminus S, \quad L \leftarrow L \cup \{\frac{u}{d}\}$
9. **return** L

How many bad primes?

$f = a_n x^n + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$ nonconstant squarefree, $\deg f = n$

$$
\begin{aligned}
p \in \mathbb{N} \textit{ bad prime} \quad :\Longleftrightarrow \quad & p \mid \mathrm{lc}(f) \text{ or } f \text{ is not squarefree in } \mathbb{F}_p[x] \\
\Longleftrightarrow \quad & p \mid \mathrm{lc}(f) \text{ or } \gcd(f, f') \neq 1 \text{ in } \mathbb{F}_p[x] \\
\Longleftrightarrow \quad & p \mid \det \mathrm{Syl}(f, f'), \text{ where}
\end{aligned}
$$

$$
\mathrm{Syl}(f, f') = \begin{pmatrix}
a_n & \dots & a_1 & a_0 & & \\
& \ddots & \ddots & \ddots & \ddots & \\
& & a_n & \dots & a_1 & a_0 \\
na_n & \dots & a_1 & & & \\
& na_n & \dots & a_1 & & \\
& & \ddots & \ddots & \ddots & \\
& & & na_n & \dots & a_1
\end{pmatrix} \in \mathbb{Z}^{(2n-1)\times(2n-1)}
$$

Sylvester matrix

Hadamard's inequality

$f = a_n x^n + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$ nonconstant squarefree, $\deg f = n$

$$|\det \mathrm{Syl}(f, f')| \leq \underbrace{(n^2 + n)^n \|f\|_\infty^{2n-1}}_{B}$$

So there are at most $\log_2 B \in O(n \log(n\|f\|_\infty))$ many bad primes.

Outline

Bivariate GCDs

F field, $f, g \in F[x, y]$, $\deg_x \leq n$, $\deg_y \leq m$
Main idea: Evaluation/interpolation

1. Choose evaluation points $a_0, \ldots, a_{2m} \in F$ such that $\mathrm{lc}_x(f), \mathrm{lc}_x(g) \in F[y]$ do not vanish at $y = a_i$ for any i
2. **for** $i = 0, \ldots, 2m$ **do** $h_i \leftarrow \gcd(f(x, a_i), g(x, a_i)) \in F[x]$
3. Compute interpolating polynomial $h \in \mathbb{F}[x, y]$ with $\deg_y f \leq 2m$ and $h(x, a_i) = h_i$ for all i
4. Using the EEA, perform rational reconstruction to compute $H \in \mathbb{F}(y)[x]$ with numerator and denominator degrees $\leq m$ and $H(x, a_i) = h(x, a_i)$ for all i
5. **return** H

Note: we arbitrarily chose x as the main variable and return a GCD that is monic in x

Does this work?

There may be bad evaluation points such that degree of h_i is too high.

Solution: Choose $4m$ instead of $2m$ evaluation points at random and discard any h_i whose degree is too high.

How many bad evaluation points are there?

$$\begin{aligned} a \in F \text{ bad} \quad &\iff \quad \mathrm{lc}_x(fg)(a) = 0 \\ &\qquad\quad \text{or } \deg_x \gcd(f(x,a), g(x,a)) > \deg_x \gcd(f,g) \\ &\iff \quad \mathrm{lc}_x(fg)(a) = 0 \text{ or } \det S_d(f(x,a), g(x,a)) = 0 \\ &\iff \quad \det S_d(f,g)(a) = 0, \end{aligned}$$

where S_d is a certain square submatrix of $\mathrm{Syl}_x(f, g)$.

Since every row in $\mathrm{Syl}_(f, g)$ has $\deg_y \leq m$, the degree of the determinant of a submatrix is at most $2nm$, and this is the maximal number of bad evaluation points.

Bivariate factorization

F field, $f \in F[x, y]$, $\deg_x f = n$, $\deg_y f = m$

Evaluation/interpolation does not work well because we do not know which factors at $y = a_i$ correspond to which factors at $y = a_j$

Similar to the $\mathbb{Z}[x]$ case, we choose a single evaluation point a, say $a = 0$, and use Hensel lifting and factor combination

Algorithm
Input: $f \in F[x, y]$ squarefree with $n = \deg_x f = n > 0$, $d = \mathrm{lc}_x(f)$, $m = \deg_y f$
Output: $\{f_1, \ldots, f_r\} \subset F(y)[x]$, monic irreducible, with $f = df_1 \cdots f_r$

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x,a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y+a)$, $\quad d^* \leftarrow d(y+a)$
2. $k \leftarrow 2m+1$
3. Factor $f^*(x,0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x,0) = d^*(0)h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y+a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x, a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y + a)$, $\quad d^* \leftarrow d(y + a)$
2. $k \leftarrow 2m + 1$
3. Factor $f^*(x, 0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x, 0) = d^*(0) h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y + a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x,a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y+a)$, $\quad d^* \leftarrow d(y+a)$
2. $k \leftarrow 2m+1$
3. Factor $f^*(x,0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x,0) = d^*(0) h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y+a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x, a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y + a)$, $\quad d^* \leftarrow d(y + a)$
2. $k \leftarrow 2m + 1$
3. Factor $f^*(x, 0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x, 0) = d^*(0) h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y + a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x, a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y + a)$, $\quad d^* \leftarrow d(y + a)$
2. $k \leftarrow 2m + 1$
3. Factor $f^*(x, 0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x, 0) = d^*(0)h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y + a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x, a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y + a)$, $\quad d^* \leftarrow d(y + a)$
2. $k \leftarrow 2m + 1$
3. Factor $f^*(x, 0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x, 0) = d^*(0) h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y + a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x,a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y+a)$, $\quad d^* \leftarrow d(y+a)$
2. $k \leftarrow 2m+1$
3. Factor $f^*(x,0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x,0) = d^*(0) h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \text{ rem } y^k$, $v \leftarrow \prod_{j \notin S} g_j \text{ rem } y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y+a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1 Choose $a \in F$ such that $d(a) \neq 0$ and $f(x, a)$ squarefree in $F[x]$
$f^* \leftarrow f(x, y + a)$, $\quad d^* \leftarrow d(y + a)$

2 $k \leftarrow 2m + 1$

3 Factor $f^*(x, 0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
$f^*(x, 0) = d^*(0)h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$

4 **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
$f^* \equiv d^* g_1 \cdots g_s \bmod y^k$

5 $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$

6 **for** all subsets $S \subset T$ by increasing cardinality **do**

7 $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$

8 $\quad$ Compute $U, V \in F(y)[x]$ with
numerators and denominators of degrees at most m such
that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)

9 $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y + a)\}$

10 **return** L

Bivariate Zassenhaus algorithm

1. Choose $a \in F$ such that $d(a) \neq 0$ and $f(x, a)$ squarefree in $F[x]$
   $f^* \leftarrow f(x, y + a)$, $\quad d^* \leftarrow d(y + a)$
2. $k \leftarrow 2m + 1$
3. Factor $f^*(x, 0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
   $f^*(x, 0) = d^*(0)h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$
4. **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
   $f^* \equiv d^* g_1 \cdots g_s \bmod y^k$
5. $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$
6. **for** all subsets $S \subset T$ by increasing cardinality **do**
7. $\quad u \leftarrow \prod_{j \in S} g_j \operatorname{rem} y^k$, $v \leftarrow \prod_{j \notin S} g_j \operatorname{rem} y^k$
8. $\quad$ Compute $U, V \in F(y)[x]$ with
   numerators and denominators of degrees at most m such
   that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)
9. $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y + a)\}$
10. **return** L

Bivariate Zassenhaus algorithm

1 Choose $a \in F$ such that $d(a) \neq 0$ and $f(x, a)$ squarefree in $F[x]$
$f^* \leftarrow f(x, y + a)$, $\quad d^* \leftarrow d(y + a)$

2 $k \leftarrow 2m + 1$

3 Factor $f^*(x, 0)$, yielding monic $h_1, \ldots, h_s \in F[x]$ with
$f^*(x, 0) = d^*(0) h_1 \cdots h_s$, equivalently, $f^* \equiv d^* h_1 \cdots h_s \bmod y$

4 **call** Hensel lifting to obtain monic $g_1, \ldots, g_s \in F[x]$ with
$f^* \equiv d^* g_1 \cdots g_s \bmod y^k$

5 $T \leftarrow \{1, \ldots, s\}$, $\quad L \leftarrow \varnothing$

6 **for** all subsets $S \subset T$ by increasing cardinality **do**

7 $\quad u \leftarrow \prod_{j \in S} g_j \text{ rem } y^k$, $v \leftarrow \prod_{j \notin S} g_j \text{ rem } y^k$

8 $\quad$ Compute $U, V \in F(y)[x]$ with
numerators and denominators of degrees at most m such
that $U \equiv u \bmod y^k$ and $V \equiv v \bmod y^k$ (Padé approximation)

9 $\quad$ **if** $f^* = d^* uv$ **then** $T \leftarrow T \setminus S$, $L \leftarrow L \cup \{u(x, y + a)\}$

10 **return** L

Hilbert's irreducibility theorem

Main idea for more than two variables: reduce to bivariate case by substituting values for all but two variables

F field, $f \in F[x_1, x_2, \ldots, x_k]$ irreducible, $a_3, \ldots, a_k \in F$ random.
Then $f(x_1, x_2, a_3, \ldots, a_k) \in F[x_1, x_2]$ irreducible with high probability.

Consequently: no factor combination necessary for factorization.

Multivariate GCDs

F field, $f, g \in F[x_1, x_2, \ldots, x_k]$

1. Viewing f, g as polynomials in x_1, recursively compute the $\gcd$ c of all coefficients of f and g, and let $f^* = \frac{f}{c}$ and $g^* = \frac{g}{c}$
2. Recursively compute $d \leftarrow \gcd(\mathrm{lc}_{x_1}(f^*), \mathrm{lc}_{x_1}(g^*)) \in F[x_2, \ldots, x_n]$
3. Choose many evaluation vectors $a_i = (a_{i3}, \ldots, a_{ik}) \in F^{k-2}$ such that $\deg_{x_1}$ does not drop when $x_3, \ldots, x_k$ are evaluated at a_i, for any i
4. **for** all i **do**
   $h_i \leftarrow \gcd(f^*(x_1, x_2, a_{i3}, \ldots), g^*(x_1, x_2, a_{i3}, \ldots) \in \mathbb{F}(x_2)[x_1]$
5. Compute interpolating polynomial $h \in \mathbb{F}[x_1, \ldots, x_n]$ with $h(x_1, x_2, a_3, \ldots) = d(x_2, a_3, \ldots)h_i$ for all i
6. Viewing h as a polynomial in x_1, recursively compute the $\gcd$ e of all coefficients of h, and let $h^* = \frac{h}{e}$
7. **return** h^*

Bad evaluation points

F field, $f, g \in F[x_1, x_2, \ldots, x_k]$
As in the bivariate case, an evaluation point $(a_{i3}, \ldots, a_{ik})$ can be bad for two reasons:

- The degree in x_1 drops in step 3.
- The degree of the GCD is too high in step 4.

Solution: as in the bivariate case, double the number of a_i and choose them at random.

How many evaluation points?

- It is possible to give a sufficient but generally much to large upper bound based on the degrees of the input polynomials in all variables.
- Generally, multivariate problems tend to be sparse, and a bound depending on the nonzero terms of the input polynomials can be determined.
- In the sparse case, sparse interpolation should be used as well.
- A heuristic alternative is to also interpolate the cofactors $u = f/h$ and $v = g/h$ and adaptively add more points until $f = uh$ and $g = vh$.

Multivariate factorization

F field, $f \in F[x_1, \ldots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \ldots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \ldots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \ldots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \ldots)}{c(x_2, x_3 + a_3, \ldots)}, \quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \ldots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \ldots, 0)$
5. **for** $1 \leq j \leq s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \ldots, 0) h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \ldots, 0) h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \ldots, x_k^{d_k} >$ in $F[x_1, \ldots, x_k]$
8. **return** $c_1, \ldots, c_s, f_1(x_1, x_2, x_3 - a_3, \ldots), \ldots, f_r(x_1, x_2, x_3 - a_3, \ldots)$

Multivariate factorization

F field, $f \in F[x_1, \ldots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \ldots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \ldots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \ldots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \ldots)}{c(x_2, x_3 + a_3, \ldots)}, \quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \ldots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \ldots, 0)$
5. **for** $1 \leq j \leq s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \ldots, 0)h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \ldots, 0)h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \ldots, x_k^{d_k} >$ in $F[x_1, \ldots, x_k]$
8. **return** $c_1, \ldots, c_s, f_1(x_1, x_2, x_3 - a_3, \ldots), \ldots, f_r(x_1, x_2, x_3 - a_3, \ldots)$

Multivariate factorization

F field, $f \in F[x_1, \ldots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \ldots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \ldots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \ldots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \ldots)}{c(x_2, x_3 + a_3, \ldots)}$, $\quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \ldots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \ldots, 0)$
5. **for** $1 \leq j \leq s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \ldots, 0)h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \ldots, 0)h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \ldots, x_k^{d_k} >$ in $F[x_1, \ldots, x_k]$
8. **return** $c_1, \ldots, c_s, f_1(x_1, x_2, x_3 - a_3, \ldots), \ldots, f_r(x_1, x_2, x_3 - a_3, \ldots)$

Multivariate factorization

F field, $f \in F[x_1, \dots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \dots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \dots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \dots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \dots)}{c(x_2, x_3 + a_3, \dots)}$, $\quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \dots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \dots, 0)$
5. **for** $1 \le j \le s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \dots, 0) h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \dots, 0) h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \dots, x_k^{d_k} >$ in $F[x_1, \dots, x_k]$
8. **return** $c_1, \dots, c_s, f_1(x_1, x_2, x_3 - a_3, \dots), \dots, f_r(x_1, x_2, x_3 - a_3, \dots)$

Multivariate factorization

F field, $f \in F[x_1, \ldots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \ldots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \ldots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \ldots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \ldots)}{c(x_2, x_3 + a_3, \ldots)}$, $\quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \ldots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \ldots, 0)$
5. **for** $1 \leq j \leq s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \ldots, 0)h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \ldots, 0)h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \ldots, x_k^{d_k} >$ in $F[x_1, \ldots, x_k]$
8. **return** $c_1, \ldots, c_s, f_1(x_1, x_2, x_3 - a_3, \ldots), \ldots, f_r(x_1, x_2, x_3 - a_3, \ldots)$

Multivariate factorization

F field, $f \in F[x_1, \ldots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \ldots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \ldots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \ldots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \ldots)}{c(x_2, x_3 + a_3, \ldots)}$, $\quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \ldots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \ldots, 0)$
5. **for** $1 \leq j \leq s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \ldots, 0)h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \ldots, 0)h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \ldots, x_k^{d_k} >$ in $F[x_1, \ldots, x_k]$
8. **return** $c_1, \ldots, c_s, f_1(x_1, x_2, x_3 - a_3, \ldots), \ldots, f_r(x_1, x_2, x_3 - a_3, \ldots)$

Multivariate factorization

F field, $f \in F[x_1, \ldots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \ldots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \ldots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \ldots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \ldots)}{c(x_2, x_3 + a_3, \ldots)}, \quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \ldots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \ldots, 0)$
5. **for** $1 \leq j \leq s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \ldots, 0)h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \ldots, 0)h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \ldots, x_k^{d_k} >$ in $F[x_1, \ldots, x_k]$
8. **return** $c_1, \ldots, c_s, f_1(x_1, x_2, x_3 - a_3, \ldots), \ldots, f_r(x_1, x_2, x_3 - a_3, \ldots)$

Multivariate factorization

F field, $f \in F[x_1, \ldots, x_k]$ nonconstant squarefree, $d_i = 1 + \deg_{x_i} f$

1. Compute the GCD $c \in F[x_2, \ldots, x_k]$ of all coefficients w.r.t. x_1 of f, and factor it recursively as $c = c_1 \cdots c_s$
2. Choose evaluation values $a = (a_3, \ldots, a_k) \in F^{k-1}$ such that $\deg_{x_1}$ does not drop and f^* remains squarefree when $x_3, \ldots, x_k$ are evaluated at a
3. $f^* \leftarrow \frac{f(x_1, x_2, x_3 + a_3, \ldots)}{c(x_2, x_3 + a_3, \ldots)}, \quad d \leftarrow \mathrm{lc}_{x_1}(f^*)$
4. **call** the bivariate Zassenhaus algorithm to compute the monic irreducible factors $h_1, \ldots, h_r \in F(x_2)[x_1]$ of $f^*(x_1, x_2, 0, \ldots, 0)$
5. **for** $1 \le j \le s$ **do**
6. Compute the GCD c_j of all coefficients w.r.t. x_1 of $d(x_2, 0, \ldots, 0)h_j \in F[x_2][x_1]$, as well as $g_j = \frac{d(x_2, 0, \ldots, 0)h_j}{c_j}$
7. Hensel lifting, variable by variable, yields $f^* \equiv f_1 \cdots f_r \bmod < x_3^{d_3}, \ldots, x_k^{d_k} >$ in $F[x_1, \ldots, x_k]$
8. **return** $c_1, \ldots, c_s, f_1(x_1, x_2, x_3 - a_3, \ldots), \ldots, f_r(x_1, x_2, x_3 - a_3, \ldots)$

Remarks

- This is a heuristic algorithm based on the assumption that the bivariate factors correspond uniquely to the multivariate ones. Solution: Verify the final result by multiplying all factors
- The shift $x_3 \mapsto x_3 + a_3, \dots$ in step 3 can be avoided; it is done here to simplify the presentation.
- There are also multivariate GCD algorithms based on Hensel lifting instead of interpolation.

Remarks

- This is a heuristic algorithm based on the assumption that the bivariate factors correspond uniquely to the multivariate ones. Solution: Verify the final result by multiplying all factors
- The shift $x_3 \mapsto x_3 + a_3, \ldots$ in step 3 can be avoided; it is done here to simplify the presentation.
- There are also multivariate GCD algorithms based on Hensel lifting instead of interpolation.

Remarks

- This is a heuristic algorithm based on the assumption that the bivariate factors correspond uniquely to the multivariate ones. Solution: Verify the final result by multiplying all factors
- The shift $x_3 \mapsto x_3 + a_3, \ldots$ in step 3 can be avoided; it is done here to simplify the presentation.
- There are also multivariate GCD algorithms based on Hensel lifting instead of interpolation.