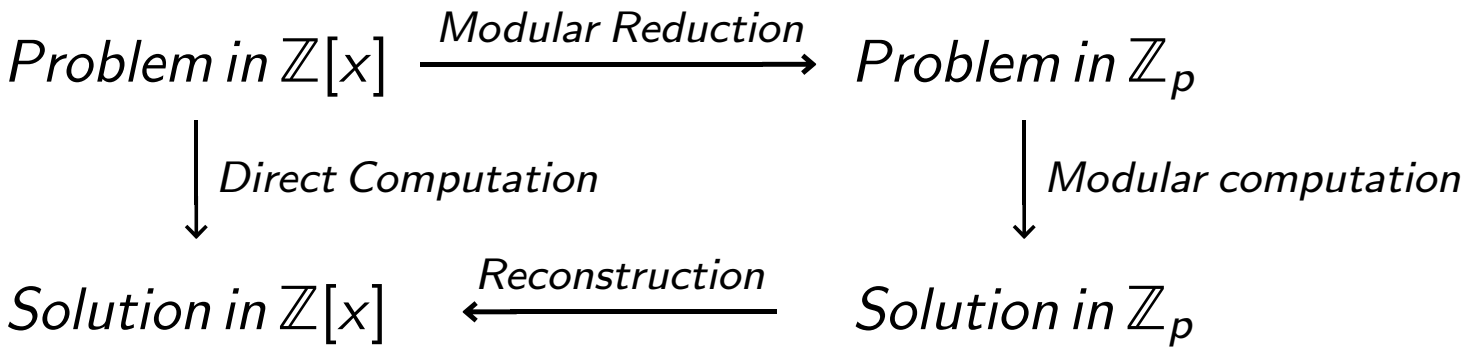
Computational Algebra. Lecture 6

M.Eulàlia Montoro

March 21, 2022

$gcd(f, g),\ f, g \in \mathbb{Z}[x]$



$gcd(f, g)$, $f, g \in \mathbb{Z}[x]$

- This presentation discusses a modular *gcd* algorithm for $\mathbb{Z}[x]$ by Char, Geddes & Gonnet (1989).
- The modulus is not a prime but a linear polynomial $x - u$.
- This is a heuristic procedure in the following sense: It's composed of a *solver* and a *checker*. The solver receives as input the description of a problem and a random number. The checker is a boolean function which receives as input the description of the problem and a proposed solution, and it determines whether the solution satisfies the problem.

Char, Geddes and Gonnet

$$\begin{array}{ccc} \textit{Problem in } \mathbb{Z}[x] & \xrightarrow{\textit{Evaluation}} & \textit{Problem in } \mathbb{Z}/\langle x-u\rangle \\ \Big\downarrow \textit{Direct Computation} & & \Big\downarrow \textit{gcd in } \mathbb{Z} \\ \textit{Solution in } \mathbb{Z}[x] & \xleftarrow{\textit{Special interpolation}} & \textit{Solution in } \mathbb{Z}/\langle x-u\rangle \end{array}$$

That is

$$\begin{array}{ccc} f, g & \xrightarrow{\textit{Evaluation}} & f(u), g(u) \\ \Big\downarrow \textit{Direct Computation} & & \Big\downarrow \textit{gcd in } \mathbb{Z} \\ h := gcd(f, g) & \xleftarrow{\textit{Special interpolation}} & h(u) | gcd(f(u), g(u)) =: c \end{array}$$

Char, Geddes and Gonnet

Let $f, g \in \mathbb{Z}[x]$ be nonzero and primitive of degree at most n, with max-norm at most A, $h = \gcd(f, g) \in \mathbb{Z}[x]$, and $u \in \mathbb{N}$ such that $u > 4A$.

- $h(u) \,|\, c := \gcd(f(u), g(u))$ in $\mathbb{Z}$ and $h(u) \neq 0$.

Proof.

Suppose $h(u) \neq 0$, then

$$\tilde{f}(x)h(x) = f(x) = f(u) + q(x)(x - u)$$

and

$$\tilde{g}(x)h(x) = g(x) = g(u) + \tilde{q}(x)(x - u)$$

therefore $h(u) \,|\, c = \gcd(f(u), g(u))$.

Char, Geddes and Gonnet

Proof.

Let us to see that $h(u) \neq 0$.
Note that $f(x) = h(x)\tilde{f}(x)$. If $h(u) = 0$, then $f(u) = 0$ and if we write

$$f(x) = x^n + \ldots + f_1 x + f_0$$

we have that $u|f_0$, but $|f_0| \leq A < \frac{u}{4}$, and therefore must be $f_0 = 0$.
Analougosly we prove that $f_1 = \ldots = f_{n-1} = 0$, and then
$f = 0$. □

Char, Geddes and Gonnet

- Let $v \in \mathbb{Z}[x]$ whose coefficients v_i satisfy $-u/2 < v_i \leq u/2$, and $v(u) = c$. Give an algorithm for computing v from c.

Algorithm 1 $v(x)$

---

procedure POLY1(u, c)

$v := 0$;

$a := c$;

$i := 0$;

while $a \neq 0$ **do**

$v_i := a \bmod u$, $v := v + v_i x^i$, $a = \frac{(a - v_i)}{u}$, $i = i + 1$

return $(v(x))$

In MATHEMATICA: $Mod[m, n, d]$ gives a result x such that $d \leq x < d + n$ and $x \bmod n = m \bmod n$.

Char, Geddes and Gonnet

- Now assume that $pp(v)$ divides both f and g. Writting $h = pp(v)w$ with a primitive $w \in \mathbb{Z}[x]$, prove that $w(u)$ divides $cont(v)$.

Proof.

$cw(u) = v(u)w(u) = cont(v)pp(v(u))w(u) = cont(v)h(u)$ and the result follows because $h(u)|c$ □

Char, Geddes and Gonnet

- Show that $u/2 \geq |w(u)| \geq |lc(w)|(u-2A)^{\deg w} > (u/2)^{\deg w}$ if w is non constant, and conclude that $h = \pm pp(v)$.

Proof.

- $u/2 \geq |w(u)|$:
  $w(u)|cont(v) \Rightarrow |w(u)| \leq |cont(v)| \leq \|v\|_\infty \leq \frac{u}{2}$
- $|w(u)| \geq |lc(w)|(u-2A)^{\deg w} > (u/2)^{\deg w}$: write $w = lc(w)\Pi_{1\leq i\leq \deg(w)}(x-\alpha_i)$, since $w|h|f$, the $\alpha_i's$ are roots of f and $|\alpha_i| \leq 2A < \frac{u}{2}$, so

$$|w(u)| = |lc(w)||\Pi_{1\leq i\leq \deg(w)}(u-\alpha_i)| \geq |lc(w)|(u-2A)^{\deg(w)} \geq$$

$$\geq |lc(w)|(u/2)^{\deg w} > (u/2)^{\deg w}$$

Char, Geddes and Gonnet

Algorithm 2 $GCD(f, g)$

procedure $\mathrm{GCD}(f, g, u)$
$A := \max(\|f\|, \|g\|)$;
$h(u) := \gcd(f(u), g(u))$;
Compute $v(x)$;
Compute $pp(v(x))$
if $rem(f(x), pp(v(x)) = 0$ and $rem(f(x), pp(v(x)) = 0$ **then**
Return$(pp(v(x))$
else return (Failed)

Homework

Trace and implement the analogue of the algorithm above for polynomials $f, g \in \mathbb{Z}[x, y]$ We just need implement the same in $\mathbb{Z}[y][x]$ since $\mathbb{Z}[y]$ is an Euclidean algortithm (with norm N). We generalize the algorithm conditions in the following manner:

1. $A = \max(N(f), N(g))$.
2. $N(u) > 4A$.
3. $|N(v_i)| \leq \frac{N(u)}{2}$
4. $c(y) = GCD(f(u, y), g(u, y))$

Homework

- Compute $h(u, y) = gcd(f, g)$.
- Now $c(y) = h(u, y) = \sum c_i y^i$.
- $v2(x, y) = \sum v_i(x) y^i \in \mathbb{Z}[x, y]$ where $v_i(x) \in \mathbb{Z}[x]$ is such that $v_i(u) = c_i$.

Homework

Algorithm 3 $v2(x, y)$

procedure POLY2$(u, c(y))$

$v2 := 0;$

$a := CoefficientList(c(y), y);$

for $i = 1,\ i \leq lenght(a),\ i++$ **do**

$v2 := v2 + poly1(u, a[i])y^{i-1},$

return $(v2(x, y))$

Homework

- Compute $v_2(x, y) := \sum v_i(y)x^i$
- Compute $cont1(y) := gcd(v_i(y))$
- Compute $cont := cont(cont1(y))$
- $pp(v_2(x, y)) := v_2(x, y)/cont$

Homework

Algorithm 4 $GCD2(f, g)$

procedure $\mathrm{GCD2}(f, g, u)$

$A := \max(\|f\|, \|g\|)$;$c(y) := \gcd(f(u, y), g(u, y))$;

Compute $v2(x, y)$;

$list = CoefficientList[v2(x, y), x]$;

$cont = Apply[PolynomialGCD, list]$;

$cont = Apply[GCD, CoefficientList[cont, y]]$;

$list = list/cont$;

$list = Reverse[list]$;

$ppv2 = Expand[FromDigits[list, x]]$;

if $CoeffList(PolynomialRemainder(f(x, y), ppv2, y), x) = 0$ and $CoeffList(PolynomialRemainder(g(x, y), ppv2, y), x) = 0$ **then**, **return** (p, u $\geq 4A$),

else, **return** (False);