GCD in nearly linear time

Ramprasad Saptharishi

August 25, 2025

Abstract

This is an exposition of the deterministic near-linear time algorithm for polynomials GCD of Schönhage.

Notation. For a polynomial $f(x)$, we will use $|f|$ to denote its degree.

1 Extended Euclid's Algorithm

Given a pair of polynomials f, g (with $|f| > |g|$) the Extended Euclid's algorithm provides a sequence of quotients and remainders. We will use the following notation throughout.

$$\begin{aligned} r_0 &:= f, \\ r_1 &:= g, \\ \text{For all } i \geq 1, \quad r_{i+1} &:= r_{i-1} - q_i r_i. \end{aligned}$$

We also have the corresponding Bézout coefficients (u_i, v_i) that for all $i \geq 0$ satisfy $u_i f + v_i g = r_i$. These also satisfy a similar relation:

$$\begin{aligned} (u_0, v_0) &:= (1, 0), \\ (u_1, v_1) &:= (0, 1), \\ \text{For all } i \geq 1, \quad u_{i+1} &= u_{i-1} - q_i u_i, \\ v_{i+1} &= v_{i-1} - q_i v_i. \end{aligned}$$

We will refer to the sequence $(r_0, r_1, r_2, \ldots)$ as the *remainder sequence*, $(q_1, q_2, \ldots)$ as the *quotient sequence* and $((u_0, v_0), (u_1, v_1), \ldots)$ as the *Bézout coefficient sequence*.

Base version: (2025-08-25 11:32:08 +0530) , 7280eef

1.1 Expressing as matrices

Each step of Extended Euclid's algorithm essentially replaces a pair of polynomials (r_{i-1}, r_i) with $(r_i, r_{i-1} - q_i r_i)$. This can be expressed conveniently in this matrix form:

$$\begin{bmatrix} r_i \\ r_{i+1} \end{bmatrix} = \begin{bmatrix} & 1 \\ 1 & -q_i \end{bmatrix} \begin{bmatrix} r_{i-1} \\ r_i \end{bmatrix}$$

Extending the above, we get the following.

Lemma 1.1 (Extended Euclid in matrix form)**.** *For all $i \geq 1$, we have*

$$\begin{bmatrix} r_i \\ r_{i+1} \end{bmatrix} = \begin{bmatrix} & 1 \\ 1 & -q_i \end{bmatrix} \cdots \begin{bmatrix} & 1 \\ 1 & -q_1 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \end{bmatrix}.$$

Consequently, for all $i \geq 1$, we have

$$\begin{bmatrix} u_i & v_i \\ u_{i+1} & v_{i+1} \end{bmatrix} = \begin{bmatrix} & 1 \\ 1 & -q_i \end{bmatrix} \cdots \begin{bmatrix} & 1 \\ 1 & -q_1 \end{bmatrix}$$

Observation 1.2 (Degrees of polynomials in the sequences)**.** *The remainder, quotient and Bézout coefficient sequence satisfy the following:*

1. *The degrees of r_i are strictly decreasing.*
2. *For each $i \geq 1$, $|q_i| = |r_{i-1}| - |r_i|$.*
3. *For each $i \geq 2$, $|u_i| = |q_2| + \cdots + |q_{i-1}| = |r_1| - |r_{i-1}|$ and $|v_i| = |q_1| + \cdots + |q_{i-1}| = |r_0| - |r_{i-1}|$*

Proof. The first two items are immediate from the definition. The third item follows from inspecting the matrix form in Lemma 1.1. □

2 GCD in near-linear time

The near-linear time algorithm for GCD really uses two key insights

Working with quotient sequences instead of remainder sequences: It is easy to construct simple examples of degree $\leq n$ polynomials[1] f, g such that $|r_0| + |r_1| + \cdots + |r_{t+1}| = \Omega(n^2)$ where $r_0, \ldots, r_{t+1}$ is the degree sequence. Thus, any algorithm that computes the complete sequence of remainders will inevitably take $\Omega(n^2)$ time.

On the other hand, $|q_1| + |q_2| + \cdots |q_t| = |f| - |r_{t+1}| = O(n)$. Thus, it is at least plausible to compute the complete quotient sequence in near linear time.

[1]Continuants: $f_0 = 1$, $f_1 = x$, $f_{i+1} = x f_i + f_{i-1}$ for all $i \geq 1$

Quotients are mostly determined by higher order parts: Suppose f and g are two polynomials with $|f| = n$ and $|g| = n - d$, then it turns out that the quotient of f and g is completely determined by the top d terms of f and g. This idea can be extended further to say that the first few terms of the quotient sequence is determined by the top few coefficients of f and g. This is formalised in the following lemma.

Lemma 2.1 (Quotient sequence of polynomials with large 'prefix')**.** *Suppose f, g are two polynomials with $|f| > |g|$, and suppose there exists polynomials $W, \hat{f}, \hat{g}, e_f, e_g$ such that*

$$\begin{aligned} f &= W \cdot \hat{f} + e_f \\ g &= W \cdot \hat{g} + e_g \end{aligned}$$

and assume that $|W| > |e_f|, |e_g|$.

Suppose $\{\hat{q}_1, \hat{q}_2, \ldots\}$ and $\{\hat{r}_0, \hat{r}_1, \ldots\}$ are the quotient and remainder sequence for $\hat{f}$ and $\hat{g}$. Let $t \geq 1$ be the first index satisfying $|\hat{r}_{t+1}| < \left|\hat{f}\right| / 2$.

Then, the first t terms of the quotient sequence of f and g is also $\{\hat{q}_1, \ldots, \hat{q}_t\}$. Furthermore, if $r_0 = f, r_1 = g, r_2, \ldots$ is the remainder sequence of f and g, then $|r_{t+1}| < |W| + \left|\hat{f}\right| / 2$.

Although the above lemma is more general, it would be convenient to just think of $W = x^k$ for an appropriate k and that's how we would actually end up using the lemma.

Proof. Define $r_0 = f, r_1 = g$ and $r_{i+1} = r_{i-1} - \hat{q}_i r_i$ be the purported remainder sequence of f and g assuming that $\{\hat{q}_*\}$ is indeed the quotient sequence. We will show that this is the right quotient sequence by exhibiting that the degrees of r_i are strictly decreasing.

Let $\{(\hat{u}_i, \hat{v}_i) \;:\; i \in \{0, \ldots, t\}\}$ be the Bézout coefficients associated with the quotient sequence. Then for any $i \in [t]$,

$$\begin{aligned} r_i &= u_i f + v_i g \\ &= W \cdot (\hat{u}_i \hat{f} + \hat{v}_i \hat{g}) + \hat{u}_i e_f + \hat{v}_i e_g \\ &= W \cdot \hat{r}_i + (\hat{u}_i e_f + \hat{v}_i e_g). \end{aligned}$$

Since $i \leq t$, we have

$$\begin{aligned} \left|\hat{u}_i \cdot e_f + \hat{v}_i \cdot e_g\right| &\leq \max\left(\left|\hat{u}_i e_f\right|, \left|\hat{v}_i e_g\right|\right) \\ &\leq \max\left(|\hat{u}_i|, |\hat{v}_i|\right) + \max\left(|e_f|, |e_g|\right) \\ &= \left(\left|\hat{f}\right| - |r_{i-1}|\right) + \max\left(|e_f|, |e_g|\right) && \text{(by Observation 1.2)} \\ &< \left|\hat{f}\right| / 2 + \max\left(|e_f|, |e_g|\right) && \because |r_{i-1}| > |r_t| \geq \left|\hat{f}\right| / 2 \\ &\leq |\hat{r}_i| + \max\left(|e_f|, |e_g|\right) && \because |r_i| \geq |r_t| \geq \left|\hat{f}\right| / 2 \\ &\leq |\hat{r}_i| + |W| = |W \cdot \hat{r}_i|. \end{aligned}$$

Therefore the degree of r_i is in fact the degree of $W \cdot \hat{r}_i$. Since $\hat{r}_i$ have monotonically decreasing degrees, so must be the degree of r_i. This shows that

$$|r_0| > \cdots > |r_{t-1}| > |r_t| \geq |W| + \left|\hat{f}\right| /2$$

which implies that the first $\hat{q}_1, \ldots, \hat{q}_{t-1}$ are the first $(t-1)$ terms of the quotient sequence of f, g.

For the last term,

$$\begin{aligned} r_{t+1} &= W \cdot \hat{r}_{t+1} + (\hat{u}_{t+1} \cdot e_f + \hat{v}_{t+1} \cdot e_g) \\ \implies |r_{t+1}| &\leq \max\left(|W| + |\hat{r}_{t+1}|, \left|\hat{u}_{t+1} \cdot e_f + \hat{v}_{t+1} \cdot e_g\right|\right) \end{aligned}$$

Since $|\hat{r}_{t+1}| < \left|\hat{f}\right| /2$, and $\max(|\hat{u}_{t+1}|, |\hat{v}_{t+1}|) = |\hat{v}_{t+1}| = \left|\hat{f}\right| - |r_t| < \left|\hat{f}\right| /2$, we have

$$|r_{t+1}| < |W| + \left|\hat{f}\right| /2 \leq |r_t| .$$

This implies that next term of the quotient sequence of f and g is indeed $\hat{q}_t$. □

3 The `HalfGCD` algorithm

The following subroutine for will take us "half-way" through the Extended Euclid Algorithm.

Algorithm 1: `HalfGCD`

Input: $f(x), g(x)$: two polynomials with $n = \deg(f(x)) > \deg(g(x))$

Output: The initial prefix of the quotient sequence $q_1, \ldots, q_t$ where r_{t+1} is the first remainder with $|r_{t+1}| < n/2$

if $|g| < n/2$ **then**
  return *empty sequence*
end
Write f, g as $f = x^m \cdot \hat{f} + e_f$ and $g = x^m \cdot \hat{g} + e_g$ with $m = n/2$, and $\left|e_f\right|, \left|e_g\right| < m$.
Recursively compute `HalfGCD`$(\hat{f}, \hat{g}) = (q_1, \ldots, q_a)$.
Compute the matrix $M = \begin{bmatrix} & 1 \\ 1 & -q_a \end{bmatrix} \cdots \begin{bmatrix} & 1 \\ 1 & -q_1 \end{bmatrix}$.
Compute f', g' defined as $\begin{bmatrix} f' \\ g' \end{bmatrix} := M \begin{bmatrix} f \\ g \end{bmatrix}$. (Note that $|g'| < 3n/4$, but no bound on $|f'|$)
Compute the quotient and remainder for f' divided by g' to get $f' = g' \cdot q_{a+1} + h'$
Set $k = n/4$ and write g', h' as $g' = x^k \tilde{g} + e'_g$ and $h' = x^k \tilde{h} + e'_h$.
Compute `HalfGCD`$(\tilde{g}, \tilde{h}) = (q_{a+2}, \ldots, q_b)$.
return $(q_1, \ldots, q_b)$.

Running time bound: Using standard near-linear time polynomial multiplication subroutines, it is easy to see that Line 4, Line 7, Line 8, Line 9 can all be performed in deterministic $\tilde{O}(n)$ time. Line 6 can be computed via a balanced-tree-like multiplication in $\tilde{O}(|q_1| + \cdots + |q_a|) = \tilde{O}(n)$ time.

The remaining steps are the recursive calls in Line 5 and Line 10. The polynomials $\hat{f}, \hat{g}$ in Line 5 both have degree at most $n/2$. The polynomials $\tilde{g}, \tilde{h}$ in Line 10 have degree at most $|g'| - (n/4) < n/2$ since $|g'| < 3n/4$. Therefore, both these steps are recursive calls with input polynomials of half the degree. Thus, if $T(n)$ denotes the time complexity of the above algorithm when run in polynomials f, g satisfying $n = |f| > |g|$, then

$$T(n) = 2 \cdot T(n/2) + \tilde{O}(n) \implies T(n) = \tilde{O}(n).$$

Proof of correctness: By Lemma 2.1, the quotient sequence obtained in Line 5 is the first few terms of the quotient sequence of f and g. Therefore, if $r_0 = f, r_1 = g, r_2, \ldots$ was the remainder sequence of f and g, then we have that $r_a = f'$ and $r_{a+1} = g'$ with $|g'| < m + \left|\hat{f}\right|/2 \leq 3n/4$. Thus by Line 7, we have jumped to until the a-th term in the remainder sequence. Clearly, the next quotient in the sequence is q_{a+1} computed in Line 8. Again by Lemma 2.1, we have that $(q_{a+2}, \ldots, q_b)$ are the first few terms of the quotient sequence of g', h' with the last remainder having degree less than $k + |\tilde{g}|/2 \leq (n/4) + (n/4) = n/2$. Therefore $(q_1, \ldots, q_b)$ is indeed the initial prefix of the quotient sequence of f and g until the remainder has degree smaller than $n/2$.

3.1 The final gcd algorithm

Algorithm 2: `GCD`

Input: $f(x), g(x)$: two polynomials with $n = \deg(f(x)) > \deg(g(x))$

Output: The entire quotient sequence, and the gcd of f and g

Compute `HalfGCD`$(f, g) = (q_1, \ldots, q_r)$.

Compute the matrix product

$$M_Q = \begin{bmatrix} & 1 \\ 1 & -q_a \end{bmatrix} \cdots \begin{bmatrix} & 1 \\ 1 & -q_1 \end{bmatrix}$$

Compute $\begin{bmatrix} f' \\ g' \end{bmatrix} \leftarrow M_Q \begin{bmatrix} f \\ g \end{bmatrix}$. (At this point, $|g'| < |f|/2$ but no bound on $|f'|$.)

Run one step of Euclidian division to write $f' = g'q + h'$.

Compute $Q', d = \texttt{GCD}(g', h')$. Set $Q = (q_1, \ldots, q_a, q) + Q'$ (concatenating lists).

return Q, d

It can be easily seen that the time complexity of the above algorithm can be computed as

$$\begin{aligned} T_{\texttt{GCD}}(n) &= T_{\texttt{HalfGCD}} + \tilde{O}(n) + T_{\texttt{GCD}}(n/2) \\ &= \tilde{O}(n). \end{aligned}$$

4 Other applications

The intermediate terms of the Extended Euclid Algorithm have other applications as well. Recall that for any $i \geq 1$, we have

$$u_i f + v_i g \leq r_i$$

and we have that $|u_i| = |g| - |r_{i-1}| < |g| - |r_i|$ and $|v_i| = |f| - |r_{i-1}| < |f| - |r_i|$. The following lemma essentially provides a "converse" for any such equation.

Lemma 4.1. *Suppose f, g, u, v, r are polynomials with $|f| > |g|$ and satisfy $u \cdot f + v \cdot g = r$ and $|u| + |r| < |g|$. If r_t is the first element of the remainder sequence of f and g with $|r_t| \leq |r|$ and u_t, v_t are the corresponding Bézout coefficients, then there is some nonzero polynomial α such that $r = \alpha \cdot r_t$, $u = \alpha \cdot u_t$ and $v = \alpha \cdot v_t$.*

In other words, any equation of the form $uf + vg = r$ that satisfy the degree constraints must essentially be one of the Bézout equations possibly scaled by a nonzero polynomial overall.

Proof. Consider the two equations:

$$\begin{aligned} r &= u \cdot f + v \cdot g, \\ r_t &= u_t \cdot f + v_t \cdot g. \end{aligned}$$

Eliminating f from the above two equations yields

$$r \cdot u_t - r_t \cdot u = g \cdot (u_t \cdot v - u \cdot v_t) = 0 \bmod g$$

Note that $|r| + |u_t| = |r| + |g| - |r_{t-1}| < |g|$ since $|r_{t-1}| > |r|$, and similarly $|r_t| + |u| \leq |r| + |u| < |g|$. Therefore, the degree of $r \cdot u_t - r_t \cdot u$ is less than the degree of g. This forces $ru_t = r_t u$ and hence $u_t \cdot v = u \cdot v_t$. However, since $\gcd(u_t, v_t) = 1$, it must be that u_t divides u and v_t divides v with the ratios being the same. Hence, there is some nonzero polynomial α such that $u = u_t \cdot \alpha$, $v = \alpha \cdot v_t$ and $r = \alpha \cdot r_t$. □

The following is a slight variant of the above lemma (with basicall the same proof).

Lemma 4.2. *Suppose f, g, u, v, r are polynomials with $|f| > |g|$ and satisfy $u \cdot f + v \cdot g = r$. Suppose we additionally also have a parameter $s \in \mathbb{N}$ such that $|r| < s$ and $|u| + s \leq |g|$.*

If r_t is the first element of the remainder sequence of f and g with $|r_t| < s$ and u_t, v_t are the corresponding Bézout coefficients, then there is some nonzero polynomial α such that $r = \alpha \cdot r_t$, $u = \alpha \cdot u_t$ and $v = \alpha \cdot v_t$. □

4.1 Decoding Reed-Solomon codes

A cool corollary of the above lemma is the following near-linear time decoding algorithm for Reed-Solomon codes (due to Shuhong Gao). Let us assume that we are dealing with message polynomials $m(x)$ of degree at most k, and are evaluating on points $\alpha_1, \ldots, \alpha_n$. Suppose we are given a received word $(\beta_1, \ldots, \beta_n)$ that is within distance less than $(n-k)/2$.

Let $f(x)$ be the unique polynomial of degree at most $n-1$ such that $f(\alpha_i) = \beta_i$ for all $i \in [n]$ and let $g(x) = (x - \alpha_1) \cdots (x - \alpha_n)$ which we have access to. Let $E(x) = \prod_{i : m(\alpha_i) \neq \beta_i} (x - \alpha_i)$, the error locator polynomial (which we do not have access to). Then note that $E(x) \cdot f(x) = E(x) \cdot m(x) \mod g(x)$. Therefore, there is some polynomial $c(x)$ such that

$$E(x) \cdot f(x) + c(x) \cdot g(x) = E(x) \cdot m(x).$$

Note that $|E| < (n-k)/2$ and $|E \cdot m| < (n-k)/2 + k = (n+k)/2$. Thus, the above equation is of the form in Lemma 4.2 with $E(x)$ playing the role of u and $E(x) \cdot m(x)$ playing the role of r, and $(n+k)/2$ playing the role of s. Thus, the above equation must be a scaling of a Bézout equation for the polynomials $f(x)$ and $g(x)$. This yields the following algorithm.

1. Compute the polynomial $f(x)$ such that $|f| \leq n-1$ and $f(\alpha_i) = \beta_i$. Compute $g(x) = (x - \alpha_1) \cdots (x - \alpha_n)$.

2. Using the quotient sequence, compute the first t such that $u_t f + v_t g = r_t$ with $|r_t| < (n+k)/2$.

3. Return r_t / u_t as the message polynomial.