Abstract

Subquadratic divide-and-conquer algorithms for computing the greatest common divisor have been studied for a couple of decades. The integer case has been notoriously difficult, with the need for "backup steps" in various forms. One central idea is the "half-gcd" operation, HGCD. HGCD takes two n-bit numbers as inputs, and outputs two numbers of size $\approx n/2$ with the same GCD, together with a transformation matrix with elements also of size $\approx n/2$. This talk explains why backup steps are necessary for algorithms based directly on the quotient sequence, and proposes a robustness criterion that is used to construct a simpler HGCD algorithm without any backup steps.

Subquadratic GCD

Niels Möller

May 15, 2008

Outline

Background

History

- 300 BC (or even earlier): Euclid's algorithm.
- 1938: Lehmer's algorithm.
- 1961: Binary GCD described by Stein.
- 1994, 1995: Sorensson, Weber.
- 1970, 1971: Knuth and Schönhage, subquadratic computation of continued fractions.
- ca 1987: Schönhage's "controlled Euclidean descent", unpublished.
- 2004: Stéhle and Zimmermann, recursive binary GCD.
- 2005–2008: Möller. Left-to-right algorithm. Simpler and slightly faster than earlier algorithms.

Comparison of GCD algorithms

Algorithm	Time (ms)	# lines	
`mpn_gcd`	1440	304	GMP-4.1.4 (Weber)
`mpn_rgcd`	87	1967	"Classical" Schönhage GCD
`mpn_bgcd`	93	1348	Rec. bin. (Stehlé/Zimmermann)
`mpn_sgcd`	100	760	1987 alg. (Schönhage/Weilert)
`mpn_ngcd`	85	733	New algorithm for GMP-5

Comparison of GCD algorithms

Algorithm	Time (ms)	# lines	
mpn_gcd	1440	304	GMP-4.1.4 (Weber)
mpn_rgcd	87	1967	"Classical" Schönhage GCD
mpn_bgcd	93	1348	Rec. bin. (Stehlé/Zimmermann)
mpn_sgcd	100	760	1987 alg. (Schönhage/Weilert)
mpn_ngcd	85	733	New algorithm for GMP-5

- Benchmarked on 32-bit AMD, with inputs of 48 000 digits.
- Cross-over around 7 700 digits.

Questions

Q Where does the complexity come from?

A Accurate computation of the quotient sequence.

Q How to avoid that?

A Stop bothering about quotients.

What is HGCD?

Definition (Reduction)

$$\begin{pmatrix} A \\ B \end{pmatrix} = M \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

- Positive integers A, B, α, and β.
- Matrix M, non-negative integer elements.
- $\det M = 1$.

What is HGCD?

Definition (Reduction)

$$\begin{pmatrix} A \\ B \end{pmatrix} = M \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

- Positive integers A, B, α, and β.
- Matrix M, non-negative integer elements.
- $\det M = 1$.

Fact

For any reduction, $\text{GCD}(A, B) = \text{GCD}(\alpha, \beta)$

What is HGCD?

Definition (Reduction)

$$\begin{pmatrix} A \\ B \end{pmatrix} = M \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

- Positive integers A, B, α, and β.
- Matrix M, non-negative integer elements.
- $\det M = 1$.

Fact

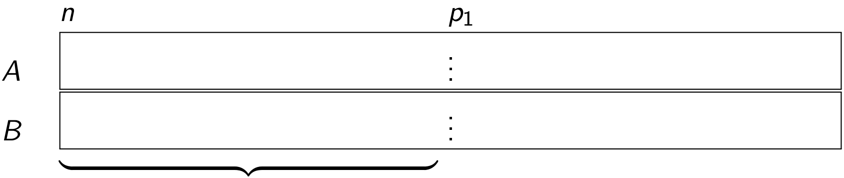
For any reduction, $\mathrm{GCD}(A, B) = \mathrm{GCD}(\alpha, \beta)$

Definition (HGCD, "half gcd")

Input: A, B, of size n

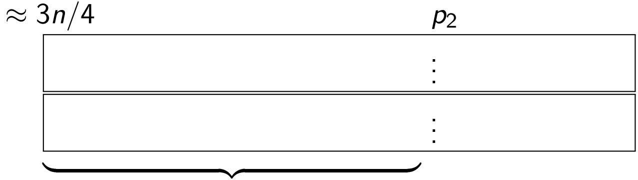
Output: M, with size of α, β and M elements $\approx n/2$

Main idea of subquadratic HGCD



$$M_1 \leftarrow \mathrm{HGCD}(\lfloor 2^{-p_1} A \rfloor, \lfloor 2^{-p_1} B \rfloor)$$

$$\begin{pmatrix} A \\ B \end{pmatrix} \leftarrow M_1^{-1} \begin{pmatrix} A \\ B \end{pmatrix}$$



$$M_2 \leftarrow \mathrm{HGCD}(\lfloor 2^{-p_2} A \rfloor, \lfloor 2^{-p_2} B \rfloor)$$

$$M \leftarrow M_1 \cdot M_2$$

Asymptotic running time

GCD(A, B)

while $\#(A, B) >$ GCD-THRESHOLD

  do

    $n \leftarrow \#(A, B),\ p \leftarrow \lfloor n/2 \rfloor$

    $M \leftarrow$ HGCD$(\lfloor 2^{-p} A \rfloor, \lfloor 2^{-p} B \rfloor)$

    $(A; B) \leftarrow M^{-1}(A; B)$

return GCD-BASE(A, B)

Running times for operations on n-bit numbers

Multiplication: $M(n) = O(n \log n \log\log n)$

HGCD: $H(n) = O(M(n) \log n)$

GCD: $G(n) \approx 2H(n)$

Quotient based HGCD

Definition (Quotient sequence)

For any positive integers a, b, the quotient sequence q_j and remainder sequence r_j are defined by

$$r_0 = a \qquad\qquad r_1 = b$$

$$q_j = \lfloor r_{j-1}/r_j \rfloor \qquad\qquad r_{j+1} = r_{j-1} - q_j r_j$$

Definition (Quotient sequence)

For any positive integers a, b, the quotient sequence q_j and remainder sequence r_j are defined by

$$\begin{aligned} r_0 &= a & r_1 &= b \\ q_j &= \lfloor r_{j-1}/r_j \rfloor & r_{j+1} &= r_{j-1} - q_j r_j \end{aligned}$$

Fact

$$\begin{pmatrix} a \\ b \end{pmatrix} = M \begin{pmatrix} r_j \\ r_{j+1} \end{pmatrix}$$

with

$$M = \begin{pmatrix} q_1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q_2 & 1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} q_j & 1 \\ 1 & 0 \end{pmatrix}$$

Theorem (Jebelean's criterion)

Let $a > b > 0$*, with remainders* r_j *and* r_{j+1}*, and*

$$\begin{pmatrix} a \\ b \end{pmatrix} = \underbrace{\begin{pmatrix} u & u' \\ v & v' \end{pmatrix}}_{=M} \begin{pmatrix} r_j \\ r_{j+1} \end{pmatrix}$$

Let $p > 0$ *be arbitrary,* $0 \le A', B' < 2^p$*, and define*

$$\begin{pmatrix} A \\ B \end{pmatrix} = 2^p \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} A' \\ B' \end{pmatrix}$$
$$\begin{pmatrix} R_j \\ R_{j+1} \end{pmatrix} = 2^p \begin{pmatrix} r_j \\ r_{j+1} \end{pmatrix} + M^{-1} \begin{pmatrix} A' \\ B' \end{pmatrix}$$

For even j*, the following two statements are equivalent:*

(i) $r_{j+1} \ge v$ *and* $r_j - r_{j+1} \ge u + u'$

(ii) *For any* p *and any* A', B'*, the* j*th remainders of* A *and* B *are* R_j *and* R_{j+1}*. The quotient sequences are the same.*

Theorem (Jebelean's simplified criterion)

Let $a > b > 0$, with remainders r_j, r_{j+1} and r_{j+2}, and

$$\begin{pmatrix} a \\ b \end{pmatrix} = M \begin{pmatrix} r_j \\ r_{j+1} \end{pmatrix}$$

Assume that $\#r_{j+2} > \lceil n/2 \rceil$, with $n = \#a$. Let $p > 0$ be arbitrary, $0 \leq A', B' < 2^p$, and define

$$\begin{pmatrix} A \\ B \end{pmatrix} = 2^p \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} A' \\ B' \end{pmatrix}$$

$$\begin{pmatrix} R_j \\ R_{j+1} \end{pmatrix} = 2^p \begin{pmatrix} r_j \\ r_{j+1} \end{pmatrix} + M^{-1} \begin{pmatrix} A' \\ B' \end{pmatrix}$$

Then the jth remainders of A and B are R_j and R_{j+1}. The quotient sequences are the same.

Quotient based HGCD

A generalization of Lehmer's algorithm

Define $\textsc{hgcd}(a, b)$ to return an M satisfying Jebelean's criterion.

Example (Recursive computation)

$$
\begin{aligned}
(a; b) &= (858\,824; 528\,747) \\
M_1 &= (13, 8; 8, 5) && \text{No difficulties} \\
(c; d) &= M_1^{-1}(a; b) = 16\,(4009; 194) + (0; 15) \\
M_2 &= \textsc{hgcd}(4009, 194) = (21, 20; 1, 1) \\
M_2^{-1}(4009; 194) &= (129; 65) && \text{Satisfies Jebelean} \\
M &= M_1 \cdot M_2 = (281, 268; 173, 165) \\
M^{-1}(a; b) &= (1764; 1355)
\end{aligned}
$$

Backup step

Example (Continued)

$$
\begin{aligned}
(a; b) &= (858\,824; 528\,747) \\
M &= M_1 \cdot M_2 = (281, 268; 173, 165) \\
M^{-1}(a; b) &= (1764; 1355) && \text{Violates Jebelean}
\end{aligned}
$$

M corresponds to quotients $1, 1, 1, 1, 1, 1, 20, 1$.
E.g., $(A; B) = 8\,(a; b) + (1; 7)$ has quotient sequence starting with $1, 1, 1, 1, 1, 1, 20, 2$.

Backup step

Example (Continued)

$$
\begin{aligned}
(a; b) &= (858\,824; 528\,747) \\
M &= M_1 \cdot M_2 = (281, 268; 173, 165) \\
M^{-1}(a; b) &= (1764; 1355)
\end{aligned}
$$

Violates Jebelean

M corresponds to quotients $1, 1, 1, 1, 1, 1, 20, 1$.
E.g., $(A; B) = 8\,(a; b) + (1; 7)$ has quotient sequence starting with $1, 1, 1, 1, 1, 1, 20, 2$.

Conclusion

- The quotients are correct for $(a; b)$, but not robust enough.
- Must drop final quotient before returning $\textsc{hgcd}(a, b)$.

Robust HGCD

A robustness condition

Definition (Robust reduction)

A reduction M of $(A; B)$ is robust iff

$$M^{-1}\left\{\begin{pmatrix} A \\ B \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix}\right\} > 0$$

for all "small" $(x; y)$. More precisely, for all $(x; y) \in S$, where

$$S = \{(x; y) \in \mathbb{R}^2, |x| < 2, |y| < 2, |x - y| < 2\}$$

A robustness condition

Definition (Robust reduction)

A reduction M of $(A; B)$ is robust iff

$$M^{-1}\left\{\begin{pmatrix} A \\ B \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix}\right\} > 0$$

for all "small" $(x; y)$. More precisely, for all $(x; y) \in S$, where

$$S = \{(x; y) \in \mathbb{R}^2, |x| < 2, |y| < 2, |x - y| < 2\}$$

Theorem

The reduction

$$\begin{pmatrix} A \\ B \end{pmatrix} = \underbrace{\begin{pmatrix} u & u' \\ v & v' \end{pmatrix}}_{=M} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

is robust iff $\alpha \geq 2\max(u', v')$ *and* $\beta \geq 2\max(u, v)$

HGCD based on robustness

HGCD(A, B)

$n \leftarrow \#(A, B)$
$p_1 \leftarrow \lfloor n/2 \rfloor$
$M_1 \leftarrow \textsc{hgcd}(\lfloor 2^{-p_1} A \rfloor, \lfloor 2^{-p_1} B \rfloor)$
$(C; D) \leftarrow M_1^{-1}(A; B)$ $\quad \triangleright \ \#|C - D| \approx 3n/4$
One subtraction and one division step on $(C; D)$. Update M_1.
$p_2 \leftarrow \# M_1 + 2$
$M_2 \leftarrow \textsc{hgcd}(\lfloor 2^{-p_2} C \rfloor, \lfloor 2^{-p_2} D \rfloor)$
return $M_1 \cdot M_2$

HGCD based on robustness

HGCD(A, B)

$n \leftarrow \#(A, B)$
$p_1 \leftarrow \lfloor n/2 \rfloor$
$M_1 \leftarrow$ HGCD$(\lfloor 2^{-p_1} A \rfloor, \lfloor 2^{-p_1} B \rfloor)$
$(C; D) \leftarrow M_1^{-1}(A; B)$ $\qquad \triangleright \ \#|C - D| \approx 3n/4$
One subtraction and one division step on $(C; D)$. Update M_1.
$p_2 \leftarrow \# M_1 + 2$
$M_2 \leftarrow$ HGCD$(\lfloor 2^{-p_2} C \rfloor, \lfloor 2^{-p_2} D \rfloor)$
return $M_1 \cdot M_2$

$$c = \lfloor 2^{-p_2} C \rfloor \qquad \qquad \widetilde{c} = 2^{-p_2} C - c$$

$$M^{-1} \left\{ \begin{pmatrix} A \\ B \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \right\} = 2^{p_2} M_2^{-1} \left\{ \begin{pmatrix} c \\ d \end{pmatrix} + \underbrace{\begin{pmatrix} \widetilde{c} \\ \widetilde{d} \end{pmatrix} + 2^{-p_2} M_1^{-1} \begin{pmatrix} x \\ y \end{pmatrix}}_{\text{disturbance } \in S} \right\}$$

Strong robustness

Definition (Strong robustess)

Let $n = \#(A, B)$ denote the bitsize of the larger of A and B. If $\# \min(\alpha, \beta) > \lfloor n/2 \rfloor + 1$, then M is strongly robust.

Lemma

If a reduction M is strongly robust, then it is robust.

Strong robustness

Definition (Strong robustess)

Let $n = \#(A, B)$ denote the bitsize of the larger of A and B. If $\# \min(\alpha, \beta) > \lfloor n/2 \rfloor + 1$, then M is strongly robust.

Lemma

If a reduction M is strongly robust, then it is robust.

Theorem (Schönhage-Weilert reduction)

For arbitrary $A, B > 0$, let $n = \#(A, B)$ and $s = \lfloor n/2 \rfloor + 1$. Assume $\# min(A, B) > s$. There exists a unique strongly robust M such that $\# \min(\alpha, \beta) > s$ and $\#|\alpha - \beta| \leq s$.

HGCD with strong robustness

HGCD(A, B)

$n \leftarrow \#(A, B)$
$s \leftarrow \lfloor n/2 \rfloor + 1$
Split: $p_1 \leftarrow \lfloor n/2 \rfloor$, $A = 2^{p_1} a + A'$, $B = 2^{p_1} b + B'$
$(\alpha, \beta, M_1) \leftarrow$ HGCD(a, b)
$(A; B) \leftarrow 2^{p_1}(\alpha; \beta) + M_1^{-1}(A'; B')$ $\quad \triangleright\ \#|A - B| \approx 3n/4$
One subtraction and one division step on $(A; B)$. Update M_1.
Split: $p_2 \leftarrow 2s - \#(A, B) + 1$, $A = 2^{p_2} a + A'$, $B = 2^{p_2} b + B'$
$(\alpha, \beta, M_2) \leftarrow$ HGCD(a, b)
$(A; B) \leftarrow 2^{p_2}(\alpha; \beta) + M_2^{-1}(A'; B')$
$M \leftarrow M_1 \cdot M_2$
while $\#|A - B| > s$ $\quad \triangleright$ At most four times
    One division step on $(A; B)$. Update M.
return (A, B, M)

Base case HGCD

- HGCD2: Special case HGCD with two-limb inputs, and an M with single-limb elements.
- Repeat: extract top two limbs, call HGCD2, apply resulting M to bignums.
- Essentially Lehmer's algorithm, with a different stop condition.
- Quadratic running time.

Further work

Matrix multiplication

$$M_1 \cdot M_2 \quad 2 \times 2 \text{ matrices}$$

Assume FFT and sizes such that transforms and pointwise multiplication take equal time.

	FFT	IFFT	Pointwise	Saving
Naive	16	8	8	0%
Schönhage-Strassen	14	7	7	12%
Invariance	8	4	8	37%
S.-S. + invariance	8	4	7	40%

Matrix-vector multiplication

- If α, β are returned: M of size $n/4$, A', B' of size $n/2$.

$$M^{-1} \cdot \begin{pmatrix} A \\ B \end{pmatrix} = 2^p \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + M^{-1} \cdot \begin{pmatrix} A' \\ B' \end{pmatrix}$$

	#Mults.	Prod. size	
Naive	4	$3n/4$	Wins in FFT range
Block	8	$n/2$	Can use invariance
S.-S.	7	$n/2$	Wins in Karatsuba range

Matrix-vector multiplication

- If α, β are returned: M of size $n/4$, A', B' of size $n/2$.

$$M^{-1} \cdot \begin{pmatrix} A \\ B \end{pmatrix} = 2^p \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + M^{-1} \cdot \begin{pmatrix} A' \\ B' \end{pmatrix}$$

	#Mults.	Prod. size	
Naive	4	$3n/4$	Wins in FFT range
Block	8	$n/2$	Can use invariance
S.-S.	7	$n/2$	Wins in Karatsuba range

- If only matrix is returned: M of size $n/4$, A, B of size n.

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} A \\ B \end{pmatrix}$$

α, β are of size $3n/4$ (cancellation!). Compute $\bmod(2^k \pm 1)$, with transform size $\approx 3n/4$.

Matrix-vector multiplication

- If α, β are returned: M of size $n/4$, A', B' of size $n/2$.

$$M^{-1} \cdot \begin{pmatrix} A \\ B \end{pmatrix} = 2^p \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + M^{-1} \cdot \begin{pmatrix} A' \\ B' \end{pmatrix}$$

	#Mults.	Prod. size	
Naive	4	$3n/4$	Wins in FFT range
Block	8	$n/2$	Can use invariance
S.-S.	7	$n/2$	Wins in Karatsuba range

- If only matrix is returned: M of size $n/4$, A, B of size n.

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} A \\ B \end{pmatrix}$$

  α, β are of size $3n/4$ (cancellation!). Compute $\bmod(2^k \pm 1)$, with transform size $\approx 3n/4$.

- Same transform size, $3n/4$, no matter if reduced numbers are available or not!

Base case optimizations

- Optimizing HGCD2 attacks the linear term in the running time.
- The quadratic term is the computation

$$M^{-1}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} v'a - u'b \\ -va + ub \end{pmatrix}$$

  Using `mpn_mul_1` and `mpn_submul_1` uses four loops. Try writing a single loop to compute $v'a - u'b$.
- Or try writing a loop that computes two products $v'a$ and va.
- The matrix elements have high bit clear. May simplify sign or carry handling.
- If we have efficient `mpn_mul_2` and `mpn_submul_2`, implement HGCD4, as two calls to HGCD2. Then apply an M with two-limb elements to the bignums.

Recursive binary GCD

Binary (2-adic) division

Notation

$v(x)$ denotes the number of trailing zeros: $2^{-v(x)}x$ is an odd integer.

Assume that $v(a) < v(b)$. Put

$$a' = 2^{-v(a)}a \qquad b' = 2^{-v(b)}b \qquad k = v(b) - v(a)$$

Define a quotient

$$q = -a'(b')^{-1} \pmod{2^{k+1}}$$

and represent it as an integer in the symmetric interval $|q| < 2^k$.
Define the remainder

$$r = a + 2^{-k}qb$$

Then

$$v(r) > v(b) \qquad |r| < |a| + |b| \qquad \mathrm{GCD}(b, r) = 2^k\,\mathrm{GCD}(a, b)$$

Binary quotient sequence

Definition (Binary quotient sequence)

For odd a and even b, define a binary quotient and remainder sequence by

$$r_0 = a \qquad r_1 = b$$
$$q_j = \mathrm{bdiv}(r_{j-1}, r_j) \qquad r_{j+1} = r_{j-1} + 2^{v(r_{j-1})-v(r_j)} q_j r_j$$

Theorem

The sequence terminates with $r_j = 0$ for some finite j.

Proof.

Assume as $r_j \neq 0$. Then since 2^j divides r_j, we have

$$2^j \leq |r_j| \leq \max(|a|, |b|)\, F_{j+1}$$

□

Binary HGCD

Definition (BHGCD)

Input: Size n, odd A, even B, with $|A|, |B| < 2^n$.

Output: Matrix M, integer v, odd a, even b, such that

$$\begin{pmatrix} a \\ b \end{pmatrix} = 2^{-v} \begin{pmatrix} r_j \\ r_{j+1} \end{pmatrix} = 2^{-2v} M \begin{pmatrix} A \\ B \end{pmatrix}$$

and $v = v(r_j) < \lfloor (n-1)/2 \rfloor \leq v(r_{j+1})$

Fact

$$\textsc{gcd}(a, b) = \gcd(A, B)$$

Binary recursive algorithm

BHGCD(A, B, n)

$k \leftarrow \lfloor (n-1)/2 \rfloor$
if $v(B) \geq k$ **return** $0, A, B, I$
Split: $n_1 = k + 1$, $A = 2^{n_1} A' + a$, $B = 2^{n_1} B' + b$
$(j_1, \alpha, \beta, M) \leftarrow$ BHGCD(a, b, n_1)
$(A; B) \leftarrow (\alpha, \beta) + 2^{n_1 - 2j_1} M(A'; B')$
$v_1 \leftarrow v(B)$
if $j_1 + v_1 \geq k$ **return** j_1, A, B, M
$q \leftarrow \mathrm{bdiv}(A, B)$
$(A, B) \leftarrow 2^{-v_1}(B, A + 2^{-v_1} q B)$
$M \leftarrow (0, 2^{v_1}; 2^{v_1}, q) \cdot M$
if $j_1 + v_1 + v(B) \geq k$ **return** j_1, A, B, M
Split: $n_2 \leftarrow 2(k - j_1 - v_1) + 1$, $A = 2^{n_2} A' + a$, $B = 2^{n_2} B' + b$
$(j_2, \alpha, \beta, M') \leftarrow$ BHGCD(a, b, n_2)
$(A; B) \leftarrow (\alpha, \beta) + 2^{n_2 - 2j_2} M'(A'; B')$
$M \leftarrow M' \cdot M$
return $j_1 + v_1 + j_2, A, B, M$