Advanced Computer Algebra: Euclidean Methods

Marc Moreno Maza
University of Western Ontario
AM583 – Fall 2003

April 27, 2004

1 The Euclidean Algorithm

1.1 Euclidean Domains

Definition 1 *An integral domain* R *endowed with a function* $d : R \longmapsto \mathbb{N} \cup \{-\infty\}$ *is an* Euclidean domain *if the following two conditions hold*

- *for all* $a, b \in R$ *with* $a \neq 0$ *and* $b \neq 0$ *we have* $d(a\,b) \geq d(a)$,
- *for all* $a, b \in R$ *with* $b \neq 0$ *there exist* $q, r \in R$ *such that*

$$a \;=\; bq + r \quad \text{and} \quad d(r) < d(b). \tag{1}$$

The elements q *and* r *are called the* quotient *and the* remainder *of* a *w.r.t.* b *(although* q *and* r *may not be unique). The function* d *is called the* Euclidean size.

Example 1 Here are some classical examples.

- $R = \mathbb{Z}$ with $d(a) =\mid a \mid$ for $a \in \mathbb{Z}$. Here the quotient q and the remainder r of a w.r.t. b (with $b \neq 0$) can be made unique by requiring $r \geq 0$ (hence we have $0 \leq r < b$).

- $R = \mathbf{k}[x]$ where $\mathbf{k}$ is a field with $d(a) = \deg(a)$ the degree of a for $a \in R, a \neq 0$ and $d(0) = -\infty$. Uniqueness of the quotient and the remainder is easy to show in that case. Indeed

  $$a = b\,q_1 + r_1 = b\,q_2 + r_2 \text{ with } \deg(r_1) < \deg(b) \text{ and } \deg(r_2) < \deg(b) \tag{2}$$

  implies

  $$r_1 - r_2 = b\,(q_1 - q_2) \text{ with } \deg(r_1 - r_2) < \deg(b) \tag{3}$$

  Hence we must have $q_1 - q_2 = 0$ and thus $r_1 - r_2 = 0$.

- $R = \mathbf{k}$ is a field with $d(a) = 1$ for $a \in \mathbf{k}, a \neq 0$ and $d(0) = 0$. In this case the quotient q and the remainder r of a w.r.t. b are a/b and 0 respectively.

- Let R be the ring of the complex numbers whose real and imaginary parts are integer numbers. Hence

  $$R = \{x + iy \mid x, y \in \mathbb{Z}\} \tag{4}$$

  Consider as a map d from R to $\mathbb{N} \cup \{-\infty\}$ the norm of an element. Hence $d(x + iy) = x^2 + y^2$ with $x, y \in \mathbb{Z}$. It is easy to check that for every $a, b \in R$ with $a, b \neq 0$ we have $d(a\,b) \geq d(a)$. Indeed for $x, y, z, t \in \mathbb{Z}$ we have

  $$\begin{array}{rcl} d((x + iy)(z + it)) & = & d(x\ z - t\ y + (y\ z + t\ x)\ i) \\ & = & (x\,z - t\,y)^2 + (y\,z + t\,x)^2 \\ & = & x^2\,z^2 + t^2\,y^2 - 2x\,z\,t\,y + y^2\,z^2 + t^2\,x^2 + 2x\,z\,t\,y \\ & = & x^2\,(z^2 + t^2) + y^2\,(z^2 + t^2) \\ & = & (y^2 + x^2)\,(z^2 + t^2) \\ & = & d(x + iy)\,d(z + it) \end{array} \tag{5}$$

  Moreover for every $a \neq 0$ we have $d(a) \geq 1$. Therefore we have proved that $d(a\,b) \geq d(a)$ holds for every $a, b \in R$ with $a, b \neq 0$. Now given $a, b \in R$ with $b \neq 0$ we are looking for a quotient and a remainder of a w.r.t. b. Hence we are looking for q such that $d(a - b\,q) < d(b)$. Such a q can be constructed as follows. Let q' be such that $a - q'\,b = 0$ that is $q' = -a/b = -a\overline{b}/d(b)$ where $\overline{b}$ is the conjugate of b. Hence q' writes

$x' + iy'$ with $x', y' \in \mathbb{Q}$. Let $x, y \in \mathbb{Z}$ be such that $\mid x - x' \mid \leq 1/2$ and $\mid y - y' \mid \leq 1/2$. Then

$$
\begin{array}{rcl}
d(a - b\,q) & = & d(a - b\,q + b\,q' - b\,q') \\
& = & d(b(q - q')) \\
& = & d(b)\,(\mid x - x' \mid^2 + \mid y - y' \mid^2) \\
& \leq & d(b)/2 \\
& < & d(b).
\end{array}
\tag{6}
$$

It turns out that several q can be chosen. For instance with $a = 1 + i$ and $b = 2 - 2\,i$ we have $a - b\,q = -1 - i$ with $q = i$ and $a - b\,q = 1 + i$ with $q = 0$. In both cases $d(a - b\,q) = 2 < 4 = d(b)$. Finally this shows that a quotient and a remainder of a w.r.t. b may not be uniquely defined in R.

1.2 The Euclidean Algorithm

Remark 1 Let $a, b \in R$ with $b \neq 0$. Let r be the remainder of a w.r.t. b. Let $c \in R$. It is easy to see that

$$
\left\{ \begin{array}{ccc} c & \mid & a \\ c & \mid & b \end{array} \right. \iff \left\{ \begin{array}{ccc} c & \mid & b \\ c & \mid & r \end{array} \right.
\tag{7}
$$

where $x \mid z$ means that x divides z, that is there exists y such that $xy = z$.

Algorithm 1

Input: $a, b \in R$.

Output: *$g \in R$ a gcd of a and b.*

$r_0 := a$
$r_1 := b$
$i := 2$
while $r_{i-1} \neq 0$ **repeat**
  $r_i := r_{i-2}$ rem r_{i-1}
  $i := i + 1$
return r_{i-2}

Let k be the greatest value of i in Algorithm 1 such that $r_i \neq 0$. From Remark 1 we have

$$\left\{ \begin{array}{lcl} c & | & a \\ c & | & b \end{array} \right. \iff \left\{ \begin{array}{lcl} c & | & b \\ c & | & r_2 \end{array} \right. \iff \cdots \iff \left\{ \begin{array}{lcl} c & | & r_{k-1} \\ c & | & r_k \end{array} \right. \iff \left\{ \begin{array}{lcl} c & | & r_k \\ c & | & 0 \end{array} \right. \tag{8}$$

Hence the following properties hold:

- every divisor of a and b divides r_k,
- r_k divides a and b.

Therefore Algorithm 1 computes a gcd of a and b. However this gcd may not be the nicest one.

```
(37) -> a:= (4*x-1/2) * (x+2) * (5*x+1) * (1/20*x+1)

          4   883  3   333  2   49
   (37)  x  + --- x  + --- x  + -- x - 1
               40        8       20

(38) -> b := (4*x-1/2) * (x+4) * (5*x-1) * (1/20*x+1)

          4   947  3   2889  2   127
   (38)  x  + --- x  + ---- x  - --- x + 2
               40        40        5

(39) -> r2 := a rem b

           8  3   153  2   557
   (39)  - - x  - --- x  + --- x - 3
           5        5        20

(40) -> r3 := b rem r2

         209  2   33231     209
   (40)  --- x  + ----- x - ---
          80        640      32

(41) -> r4 := r2 rem r3

   (41)  0
```

Remark 2 Let R be an Euclidean domain such for every $a \in R$ we can choose a *canonical associate* denoted by $\mathrm{normal}(a)$ and called the *normal form* of a. Because of the polynomial case, the unit u such that $a = u\,\mathrm{normal}(a)$ is denoted $\mathrm{lc}(a)$ and called the *leading coefficient* of a. Then $\mathrm{normal}(r_k)$ where r_k is the result of Algorithm 1 can be called THE GCD of a and b. This leads to the following AXIOM code.

```
euclideanGcd(a,b) ==
         a:=unitCanonical a
         b:=unitCanonical b
         while not zero? b repeat
            (a,b):= (b,a rem b)
            b:=unitCanonical b
         return a
```

where `unitCanonical:` `%` $\longmapsto$ `%` is the AXIOM function for computing the normal form of an element. Moreover normalizing the intermediate r_i make computations easier.

```
(2) -> a: P := (4*x-1/2) * (x+2) * (5*x+1) * (1/20*x+1)

        4   883  3   333  2   49
   (2)  x  + --- x  + --- x  + -- x - 1
              40       8       20

(3) -> b: P := (4*x-1/2) * (x+4) * (5*x-1) * (1/20*x+1)

        4   947  3   2889  2   127
   (3)  x  + --- x  + ---- x  - --- x + 2
              40       40        5

(4) -> r2 := unitCanonical(a rem b)

        3   153  2   557      15
   (4)  x  + --- x  - --- x + --
              8        32      8

(5) -> r3 := unitCanonical(b rem r2)
```

```
          2   159      5
   (5)   x  + --- x - -
               8       2

(6) -> r4 := unitCanonical(r2 rem r3)

   (6)  0
```

1.3 The Extended Euclidean Algorithm

Remark 3 Let $r_0 = a, r_1 = b, r_2 = r_0 \text{ rem } r_1, \ldots, r_i = r_{i-2} \text{ rem } r_{i-1}, \ldots,$ $\gcd(a, b) = r_k = r_{k-2} \text{ rem } r_{k-1}$ be as in Algorithm 1. For $i = 2 \cdots k$ let q_i be the quotient of r_{i-2} w.r.t. r_{i-1} that is

$$r_{i-2} \; = \; q_i \, r_{i-1} + r_i \tag{9}$$

Hence we have

$$\begin{array}{rcl} r_2 & = & r_0 - q_2 \, r_1 \\ r_3 & = & r_1 - q_3 \, r_2 \\ \vdots & \vdots & \vdots \\ r_i & = & r_{i-2} - q_i \, r_{i-1} \\ \vdots & \vdots & \vdots \\ r_k & = & r_{k-2} - q_k \, r_{k-1} \end{array} \tag{10}$$

Observe that each r_i writes $s_i \, a + t_i \, b$. Indeed we have

$$\begin{array}{llll} r_0 = s_0 \, a + t_0 \, b & \text{with} \;\; s_0 = 1 & \text{and} & t_0 = 0 \\ r_1 = s_1 \, a + t_1 \, b & \text{with} \;\; s_1 = 0 & \text{and} & t_1 = 1 \\ r_2 = s_2 \, a + t_2 \, b & \text{with} \;\; s_2 = s_0 - q_2 \, s_1 & \text{and} & t_2 = t_0 - q_2 \, t_1 \\ r_3 = s_3 \, a + t_3 \, b & \text{with} \;\; s_3 = s_1 - q_3 \, s_2 & \text{and} & t_3 = t_1 - q_2 \, t_2 \\ \vdots & \quad\;\; \vdots \;\;\; \vdots & \;\; \vdots & \vdots \\ r_i = s_i \, a + t_i \, b & \text{with} \;\; s_i = s_{i-2} - q_i \, s_{i-1} & \text{and} & t_i = t_{i-2} - q_i \, t_{i-1} \\ \vdots & \quad\;\; \vdots \;\;\; \vdots & \;\; \vdots & \vdots \\ r_k = s_k \, a + t_k \, b & \text{with} \;\; s_k = s_{k-2} - q_k \, s_{k-1} & \text{and} & t_k = t_{k-2} - q_k \, t_{k-1} \end{array} \tag{11}$$

The elements s_k and t_k are called the *Bézout coefficients* of $\gcd(a, b)$. In order to compute **a gcd** together with its Bézout coefficients Algorithm 1 needs to be transformed as follows. The resulting algorithm (Algorithm 2) is called the *Extended Euclidean Algorithm.* Finally Algorithm 3 shows how to compute **the gcd** together with its Bézout coefficients.

Algorithm 2

Input: $a, b \in R$.

Output: $g \in R$ *a gcd of* a *and* b *together with* $s, t \in R$ *such that* $g = s\,a + t\,b$.

$r_0 := a$; $s_0 := 1$; $t_0 := 0$
$r_1 := b$; $s_1 := 0$; $t_1 := 1$
$i := 2$
while $r_{i-1} \neq 0$ **repeat**
  $q_i := r_{i-2}$ quo r_{i-1}
  $r_i := r_{i-2}$ rem r_{i-1}
  $s_i := s_{i-2} - q_i\, s_{i-1}$
  $t_i := t_{i-2} - q_i\, t_{i-1}$
  $i := i + 1$
return$(r_{i-2}, s_{i-2}, t_{i-2})$

Algorithm 3

Input: $a, b \in R$.

Output: $g \in R$ *the gcd of* a *and* b *together with* $s, t \in R$ *such that* $g = s\,a + t\,b$.

$u_0 := \mathrm{lc}(a)$; $r_0 := \mathrm{normal}(a)$; $s_0 := u_0^{-1}$; $t_0 := 0$
$u_1 := \mathrm{lc}(b)$; $r_1 := \mathrm{normal}(b)$; $s_1 := 0$; $t_1 := u_1^{-1}$
$i := 2$
while $r_{i-1} \neq 0$ **repeat**
  $q_i := r_{i-2}$ quo r_{i-1}
  $r_i := r_{i-2}$ rem r_{i-1}
  $u_i := \mathrm{lc}(r_i)$
  $r_i := \mathrm{normal}(r_i)$
  $s_i := (s_{i-2} - q_i\, s_{i-1})/u_i$
  $t_i := (t_{i-2} - q_i\, t_{i-1})/u_i$
  $i := i + 1$
return$(r_{i-2}, s_{i-2}, t_{i-2})$

Proposition 1 *Let $a, b \in \mathbf{k}[x]$ where $\mathbf{k}$ is a field. Assume $\deg(a) = n \geq \deg(b) = m$.*

- *Algorithm 3 requires at most $m + 2$ inversions and $13/2m\,n + \mathcal{O}(n)$ additions and multiplications in $\mathbf{k}$.*
- *If we do not compute the coefficients s_i, t_i then Algorithm 3 requires at most $m+2$ inversions and $5/2m\,n+\mathcal{O}(n)$ additions and multiplications in $\mathbf{k}$.*

Proof $\triangleright$ See Theorem 3.11 in [vzGG99]. $\triangleleft$

Proposition 2 *Let $a, b \in \mathbb{Z}$ be multi-precision integers written with m and m words. Algorithm 3 can be performed in $\mathcal{O}(m\,n)$ word operations.*

Proof $\triangleright$ See Theorem 3.13 in [vzGG99]. $\triangleleft$

1.4 Modular Arithmetic

Let R be a (commutative) ring (with unity) and I be an ideal of R. For $a, b \in R$ the relation

$$a - b \in I \tag{12}$$

usually denoted by

$$a \equiv b \mod I \tag{13}$$

defines an equivalence relation. If we denote by $\overline{a}^I$ (or $\overline{a}$ if not ambiguous) the class of the element a, then the residue classes of this relation forms a (commutative) ring (with unity) denoted by R/I where addition and multiplication are defined by

$$\overline{a} + \overline{b} = \overline{a+b} \quad \text{and} \quad \overline{a}\overline{b} = \overline{ab} \tag{14}$$

Example 2 Let R be an Euclidean domain and let $p \in R$ with $p \neq 0$. We consider the ideal I generated by p. The residue class ring R/I is often denoted by R/p and the class of $a \in R$ in R/p by $a \mod p$. For $a, b \in R$ the relation $a - b \in I$ means that $a - b$ is a multiple of p. Let (q_a, r_a) and (q_b, r_b) be the quotient-remainder pairs of a and b w.r.t. p respectively. We have

$$p \mid a - b \iff p \mid (q_a - q_b)p + r_a - r_b \iff p \mid r_a - r_b \tag{15}$$

For $R = \mathbb{Z}$ with positive remainder or for $R = \mathbf{k}[x]$ we have in fact

$$a \ \equiv \ b \mod p \quad \Longleftrightarrow \quad r_a \ = \ r_b \tag{16}$$

Explain why!

Example 3 Let $R = \mathbf{k}[x]$ for a field $\mathbf{k}$. Let $u \in \mathbf{k}$ and let I be the ideal generated by the polynomial $x - u$. For every $a \in R$ there exists $q \in R$ such that

$$a \ = \ q(x-u) + a(u) \tag{17}$$

So in that case for every $a, b \in R$ we have

$$a \ \equiv \ b \mod p \quad \Longleftrightarrow \quad a(u) \ = \ b(u) \tag{18}$$

Proposition 3 *Let R be an Euclidean domain and let a, m be in R. Then $a \mod m$ is a unit of R/m iff $\gcd(a, m) = 1$. In this case the Extended Euclidean Algorithm can be used to compute the inverse of $a \mod m$.*

PROOF $\triangleright$ Indeed, let g be the gcd of a and m and let s, t be the corresponding Bézout coefficients. Hence we have

$$s\,a + t\,m = g \tag{19}$$

If $g = 1$ then $s \mod m$ is the inverse of $a \mod m$. Conversly if $a \mod m$ is invertible there exists $b \in R$ such that

$$a\,b \equiv 1 \mod m \tag{20}$$

That is there exists $c \in R$ such that $a\,b - 1 = c\,m$. If a and m could be divided by an element d which is not a unit then we would be led to a contradiction $(d(a'\,b + c\,m') = 1)$. Hence $\gcd(a, m) = 1$. $\triangleleft$

Remark 4 The above remarks and proposition lead us to the following ALDOR implementation of the residue class ring of an Euclidean domain R by one of its element.

```
ResidueClassRing(R:EuclideanDomain, p:R): Ring with {
         class: R -> %;
         sample: % -> R;
} == R add {
```

```
        Rep == R;
        import from R;

        1: % == per(1);
        0: % == per(0);
        zero?(x:%): Boolean == zero?(rep(x));
        (x:%) = (y:%) : Boolean == zero?(x-y);
        one?(x:%): Boolean == x = 1;
        class(a:R):% == {
            (q,r) := divide(a,p);
            per(r);
        }
        sample(x:%): R == rep(x);
        (x:%) + (y:%) : % == class(rep(x) + rep(y));
        -(x:%) : % == class(-rep(x));
        (x:%) * (y:%) : % == class(rep(x) * rep(y));
        (n:Integer) * (x:%): % == class(n * rep(x));
        (x:%) ^ (n:NonNegativeInteger) : % == class(rep(x) ^ n);
        recip(x:%): Partial(%) == {
                 import from Partial(%);
                 (u,v,g) := extendedEuclidean(rep(x),p);
                 h?: Partial(R) := recip(g);
                 failed? h? => failed();
                 h: R := retract(h?);
                 [class(h * u)];
        }
}
```

Proposition 4 *Let* $\mathbf{k}$ *be a field and* $f \in \mathbf{k}[x]$ *with degree* $n \in \mathbb{N}$. *One arithmetic operation in the residue class ring* $\mathbf{k}[x]/f$, *that is, addition, multiplication or division by an invertible element can be done using* $\mathcal{O}(n^2)$ *arithmetic operations in* $\mathbf{k}$.

Explain why!

Remark 5 Let m be a positive integer. The set

$$(\mathbb{Z}/m)^* \;=\; \{a \mod m \;\mid\; \gcd(a,m) = 1\} \tag{21}$$

is the group of units of the ring $\mathbb{Z}/m$. The Euler's totient function $m \longmapsto \phi(m)$ counts the number of elements of $(\mathbb{Z}/m)^*$. By convention $\phi(1) = 1$. Then if p is a prime we have $\phi(p) = p - 1$. If m is a power p^e of the prime p then we have

$$\phi(m) \;=\; (p-1)p^{e-1} \tag{22}$$

Explain why!

Theorem 1 (Fermat's little theorem) *If $p \in \mathbb{N}$ is a prime and $a \in \mathbb{Z}$ then we have*

$$a^p \;\equiv\; a \mod p \tag{23}$$

Moreover if p does not divide a then we have $a^{p-1} \;\equiv\; 1 \mod p$.

Proof $\triangleright$ It is sufficient to prove the claim for $a = 0 \cdots p-1$ which we do by induction on a. The cases $a = 0$ and $a = 1$ are trivial. For $a > 1$ we have

$$a^p = ((a-1)+1)^p \equiv (a-1)^p + 1^p \equiv (a-1)+1 = a \mod p \tag{24}$$

$\triangleleft$

Remark 6 For a prime $p \in \mathbb{N}$ and $a \in \mathbb{Z}$ such that $a \neq 0$ and such that p does not divide a. It follows from Theorem 1 that the inverse of $a \mod p$ can be computed by

$$a^{-1} \;\equiv\; a^{p-2} \mod p \tag{25}$$

Explain why this leads to an algorithm requiring $\mathcal{O}(\log_2^3(p))$ word operations. Is this better than the modular inversion via Euclide's algorithm?

2 Modular Algorithms and interpolation

2.1 Evaluation, interpolation

Let $\mathbf{k}$ be a field and let $u = (u_0, \ldots, u_{n-1})$ be a sequence of pairwise distinct elements of $\mathbf{k}$.

Remark 7 A polymomial in $\mathbf{k}[x]$ with degree $n-1$, say

$$f \;=\; f_0 + f_1 x + \cdots + f_{n-1}x^{n-1} \tag{26}$$

can be evaluated at $x = x_0$ using *Horner's rule*

$$f(x_0) \;=\; (\cdots(f_{n-1}x_0 + f_{n-2})x_0 + \cdots + f_1)x_0 + f_0 \tag{27}$$

with $n-1$ additions and $n-1$ multiplications leading to $2n-2$ operations in the base field $\mathbf{k}$. The proof is easy by induction on $n \geq 1$.

Definition 2 *For $i = 0 \cdots n-1$ the i-th* Lagrange interpolant *is the polynomial*

$$L_i(u) \;=\; \Pi_{\substack{0 \leq j < n \\ j \neq i}} \frac{x-u_j}{u_i-u_j} \tag{28}$$

with the property that

$$L_i(u_j) = \begin{cases} 0 & \text{if} i \neq j \\ 1 & \text{otherwise} \end{cases} \tag{29}$$

Proposition 5 *Let $v_0, \ldots, v_{n-1}$ be in $\mathbf{k}$. There is a* **unique** *polynomial $f \in \mathbf{k}[x]$ with degree less than n and such that*

$$f(u_i) = v_i \;\; \text{for} \;\; i = 0 \cdots n-1. \tag{30}$$

Moreover this polynomial is given by

$$f \;=\; \Sigma_{0 \leq i < n}\, v_i\, L_i(u). \tag{31}$$

PROOF $\triangleright$ Clearly the polynomial f of Relation (31) satisfies Relation (30). Hence the existence is clear. The unicity follows from the fact that the difference of two such polynomials has

- degree less than n and
- n roots

hence is the zero polynomial. $\triangleleft$

Proposition 6 *Evaluating a polynomial $f \in \mathbf{k}[x]$ of degree less than n at n distinct points $u_0, \ldots, u_{n-1}$ or computing an interpolating polynomial at these points can be done in $\mathcal{O}(n^2)$ operations in $\mathbf{k}$.*

PROOF $\triangleright$ We saw in Remark 7 that evaluating the polynomial f of degree n at one point costs $2n-2$ operations in $\mathbf{k}$. So evaluating f at $u_0, \ldots, u_{n-1}$ amounts to $2n^2 - 2n$. Let us prove now that interpolating a polynomial at $u_0, \ldots, u_{n-1}$ can be done in $\mathcal{O}(n^2)$ operations in $\mathbf{k}$. We first need to estimate the cost of computing the i-th interpolant $L_i(u)$. Consider $m_0 m_1$, $m_0 m_1 m_2$, $\ldots m = m_0 \cdots m_{n-1}$ where m_i is the monic polynomial $m_i = x - u_i$. Let $p_i = m_0 m_1 \cdots m_{i-1}$ and $q_i = m/m_i$ for $i = 1 \cdots n$. We have

$$L_i(u) \;=\; \frac{q_i}{q_i(u_i)} \tag{32}$$

To estimate the cost of computing the $L_i(u)$'s let us start with that of m. Computing the product of the monic polynomial $p_i = m_0 m_1 \cdots m_{i-1}$ of degree i by the monic polynomial $m_i = x\_u_i$ of degree 1 costs

- i multiplications (in the field $\mathbf{k}$) to get $-u_i\, p_i$ plus
- $i-1$ additions (in the field $\mathbf{k}$) to add $-u_i\, p_i$ (of degree i) to $x\, p_i$ (of degree $i+1$ but without constant term)

leading to $2\,i-1$. Hence computing $p_2, \ldots, p_n = m$ amounts to

$$\begin{array}{rcl} \Sigma_{2 \leq i \leq n}\, 2\,i - 1 & = & \Sigma_{1 \leq i \leq n-1}\, 2\,i + 1 \\ & = & 2\Sigma_{1 \leq i \leq n-1} i \;+\; n-1 \\ & = & 2\,n\,(n-1)/2 \;+\; n-1 \\ & = & (n-1)(n+1) \end{array} \tag{33}$$

Computing q_i implies a *division-with-remainder* of the polynomial m of degree n by the polynomial m_i of degree 1. This division will have $n-1+1$ steps, each step requiring 2 operations in $\mathbf{k}$. Hence computing all q_i's amounts to $n\,2\,n$. Since q_i has degree $n-1$ computing each $q_i(u_i)$'s amounts to $2n-2$ operations in the base field $\mathbf{k}$ (from Remark 7). Then computing all $q_i(u_i)$'s amounts to $2n^2 - 2n$. Then computing each $L_i(u) = q_i/q_i(u_i)$ from the q_i's and $q_i(u_i)$'s costs n. Therefore computing all $L_i(u)$'s from scratch amounts to $(n-1)(n+1) + 2\,n^2 + 2n^2 - 2n + n^2 = 6n^2 - 2n - 1$.

Computing f from the $L_i(u)$'s requires

- to multiply each $L_i(u)$ (which is a polynomial of degree $n-1$) by the number v_i leading to n^2 operations in $\mathbf{k}$ and
- to add these $v_i\, L_i(u)$ leading to $n-1$ additions of polynomials of degree at most $n-1$ costing $(n-1)n$ operations in $\mathbf{k}$

amounting to $2\,n^2 - n$. Finally the total cost is $6n^2 - 2n - 1 + 2\,n^2 - n = 8\,n^2 - 3\,n - 1$ $\triangleleft$

Remark 8 We consider the map

$$E : \begin{array}{ccc} \mathbf{k}^n & \to & \mathbf{k}^n \\ (f_0, \ldots, f_{n-1}) & \longmapsto & (\Sigma_{0 \leq j < n} f_j u_0^j, \ldots, \Sigma_{0 \leq j < n} f_j u_{n-1}^j) \end{array} \tag{34}$$

This is just the map corresponding to evaluation of polynomials of degree less than n at points $u_0, \ldots, u_{n-1}$. It is obvious that E is $\mathbf{k}$-linear and it can be represented by the *Vandermonde matrix*

$$VDM(u_0, \ldots, u_{n-1}) \;=\; \begin{pmatrix} 1 & u_0 & u_0^2 & \cdots & u_0^{n-1} \\ 1 & u_1 & u_1^2 & \cdots & u_1^{n-1} \\ 1 & u_2 & u_2^2 & \cdots & u_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & u_{n-1} & u_{n-1}^2 & \cdots & u_{n-1}^{n-1} \end{pmatrix} \tag{35}$$

From the above discussion, this matrix is invertible iff $u_i \neq u_j$ for all $0 \leq i < j \leq n-1$. To conclude observe that both *evaluation* and *interpolation* are linear maps between coefficients and value vectors.

2.2 The Chinese Remaindering Algorithm

We start with the elementary case of the Chinese Remaindering Theorem (Theorem 2). Then we give a much more abstract version (Theorem 3). Then we state the Chinese Remaindering Theorem and the Chinese Remaindering Algorithm in the context of Euclidean domains.

Theorem 2 (Sun-Tsu, first century AD) *Let m and n be two relatively prime integers. Let $s, t \in \mathbb{Z}$ be such that $s\,m + t\,n = 1$. For every $a, b \in \mathbb{Z}$ there exists $c \in \mathbb{Z}$ such that*

$$(\forall x \in \mathbb{Z}) \qquad \left\{ \begin{array}{ll} x \equiv a & \mod m \\ x \equiv b & \mod n \end{array} \right. \quad \Longleftrightarrow \quad x \equiv c \mod m\,n \tag{36}$$

where a convenient c is given by

$$c \;=\; a + (b-a)\,s\,m = b + (a-b)t\,n \tag{37}$$

Therefore for every $a, b \in \mathbb{Z}$ the system of equations

$$\left\{ \begin{array}{ll} x \equiv a \mod m \\ x \equiv b \mod n \end{array} \right. \tag{38}$$

has a solution.

PROOF $\triangleright$ First observe that Relation (37) implies

$$c \equiv a \mod m \qquad \text{and} \qquad c \equiv b \mod n. \tag{39}$$

Now assume that $x \equiv c \mod m\,n$ holds. This implies

$$x \equiv c \mod m \qquad \text{and} \qquad x \equiv c \mod n \tag{40}$$

Thus Relations (39) and (40) lead to

$$x \equiv a \mod m \qquad \text{and} \qquad x \equiv b \mod n \tag{41}$$

Conversly

- $x \equiv a \mod m$ implies $x \equiv c \mod m$ that is m divides $x - c$ and
- $x \equiv b \mod n$ implies $x \equiv c \mod n$ that is n divides $x - c$.

Since m and n are relatively prime it follows that $m\,n$ divides $x - c$. (Gauss Lemma). $\triangleleft$

Theorem 3 *Let R be a commutative ring with unity and $I_0, I_1, \ldots, I_{n-1}$ be ideals of R. We consider the direct product of residue class rings $R/I_0 \times \cdots \times R/I_{n-1}$ (additions and multiplications are computed componentwise) together with the homomorphism*

$$M : \begin{array}{lcl} R & \to & R/I_0 \times \cdots \times R/I_{n-1} \\ r & \longmapsto & (\overline{r}^{I_0}, \ldots, \overline{r}^{I_{n-1}}) \end{array} \tag{42}$$

Then we have

- *The kernel of M is $(\cap_{i=0}^{n-1} I_i)$.*
- *If the tuple $(\overline{r_0}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}}) \in R/I_0 \times \cdots \times R/I_{n-1}$ has a pre-image $r \in R$ then we have*

$$(\forall\ i, j)\ (0 \leq i < j \leq n-1) \implies \overline{r_i}^{I_i + I_j} = \overline{r_j}^{I_i + I_j} \tag{43}$$

- *M is surjective iff for every i, j such that $0 \leq i < j \leq n-1$ we have $I_i + I_j = R$.*

Proof $\triangleright$ The first statement is obvious. Let us prove the second one. Let $r, r_0, r_1, \ldots, r_{n-1}$ be in R. We look for the pre-image of $(\overline{r_0}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}})$. Hence we look for r such that

$$M(r) = (\overline{r_0}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}}) \tag{44}$$

that is

$$(\overline{r}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}}) \;=\; (\overline{r_0}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}}) \tag{45}$$

or

$$\overline{r - r_0}^{I_0} \;=\; \overline{r - r_1}^{I_1} \;=\; \cdots \;=\; \overline{r - r_{n-1}}^{I_{n-1}}. \tag{46}$$

At this point of the proof we need the following lemma. $\triangleleft$

Lemma 1 *Let a, b be two elements of the ring R (again commutative and with unity). Let I, J be two ideals of R such that $\overline{a}^I = \overline{b}^J$. Then we have*

$$\overline{a}^{I+J} = \overline{b}^{I+J} \tag{47}$$

Proof $\triangleright$ The equation $\overline{a}^I = \overline{b}^J$ means that the residue classe of a in R/I is equal to the residue classe of b in R/J. More formally we have

$$\{x \in R \;|\; a - x \in I\} \;=\; \{x \in R \;|\; b - x \in J\} \tag{48}$$

or equivalently

$$\{x \in R \;|\; (\exists u \in I) \;|\; x = a + u\} \;=\; \{x \in R \;|\; (\exists v \in I) \;|\; x = b + v\} \tag{49}$$

Since these sets are non empty this implies

$$(\exists (u, v) \in I \times J) \;|\; a + u = b + v. \tag{50}$$

Since J is an ideal we have $-v \in J$ and thus

$$a - b \in I + J \tag{51}$$

The lemma is proved. $\triangleleft$

Continuing theorem's Proof $\triangleright$ With the previous lemma Relation 46 leads to

$$\overline{r - r_i}^{I_i+I_j} = \overline{r - r_j}^{I_i+I_j} \tag{52}$$

for i, j such that $0 \le i < j \le n-1$ or equivalently

$$\overline{r_i}^{I_i+I_j} = \overline{r_j}^{I_i+I_j} \tag{53}$$

Relation (53) is a necessary condition for the existence of a pre-image of r via the homomorphism M. Therefore we proved the second statement of the theorem. Let us prove the third one.

Let us ssume first that M is surjective. Let i, j be such that $0 \le i < j \le n-1$. There exists $r \in R$ such that

$$\overline{r}^{I_i} = \overline{1}^{I_i} \quad \text{and} \quad \overline{r}^{I_j} = \overline{0}^{I_j} \tag{54}$$

Hence

$$1 - r \in I_i \quad \text{and} \quad r \in I_j \tag{55}$$

which implies $I_i + I_j = 1$. Conversly, let us assume that the ideals $I_0, \ldots, I_{n-1}$ are pairwise coprime. Let i be in the range $0 \cdots n-1$ and consider the product I of all ideals $I_0, \ldots, I_{n-1}$ except I_i. It is a classical result that I_i and I are coprime. Let $u_i \in I_i$ and $v_i \in I$ such that $u_i + v_i = 1$. Observe that for $j = 0 \cdots n-1$

$$\overline{v_i}^{I_i} = \overline{1}^{I_i} \quad \text{and} \quad j \neq i \Longrightarrow \overline{v_i}^{I_j} = \overline{0}^{I_j}. \tag{56}$$

Now consider $(\overline{r_0}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}}) \in R/I_0 \times \cdots \times R/I_{n-1}$. Then

$$r = v_0 r_0 + \cdots + v_{n-1} r_{n-1} \tag{57}$$

satisfies

$$M(r) = (\overline{r_0}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}}) \tag{58}$$

This concludes the proof of the theorem. $\triangleleft$

Corollary 1 *Let R be a commutative ring with unity and $I_0, \ldots, I_{n-1}$ be ideals of R such that we have $I_i + I_j = R$ for every i, j with $0 \le i < j \le n-1$. Let I be the product of the ideals $I_0, \ldots, I_{n-1}$. Then we have the ring isomorphism*

$$R/I \simeq R/I_0 \times \cdots \times R/I_{n-1} \tag{59}$$

and the group isomorphism of the multiplicative groups

$$(R/I)^* \simeq (R/I_0)^* \times \cdots \times (R/I_{n-1})^* \tag{60}$$

PROOF $\triangleright$ The following ring isomorphism follows from Theorem 3

$$R/(\cap_{0\leq i\leq n-1} I_i) \;\simeq\; R/I_0 \times \cdots \times R/I_{n-1} \tag{61}$$

It is a well known fact that if the ideals $I_0, \ldots, I_{n-1}$ are pairwise coprime ($I_i + I_j = R$ for every i, j with $0 \leq i < j \leq n-1$) then their product is equal to their intersection [van91]. Therefore we have the ring isomorphism of the statement. The group isomorphism follows from the previous ring isomorphism and the fact that the element

$$(\overline{r_0}^{I_0}, \ldots, \overline{r_{n-1}}^{I_{n-1}}) \;\in\; R/I_0 \times \cdots \times R/I_{n-1} \tag{62}$$

is a unit iff every $\overline{r_i}^{I_i}$ is a unit of R/I_i. $\triangleleft$

From now on R denotes an Euclidean domain.

Corollary 2 *Let $m_0, \ldots, m_{n-1}$ be n elements pairwise coprime in the Euclidean domain R. (Hence for $0 \leq i < j < n$ we have $\gcd(m_i, m_j) = 1$.) Then let $m = m_0 \cdots m_{r-1}$. We have the ring isomorphism*

$$R/m \;\simeq\; R/m_0 \times \cdots \times R/m_{n-1} \tag{63}$$

and the group isomorphism of the multiplicative groups

$$(R/m)^* \;\simeq\; (R/m_0)^* \times \cdots \times (R/m_{n-1})^* \tag{64}$$

PROOF $\triangleright$ These are well known facts in an Euclidean domain R.

- Every ideal I of R is generated by a single element. (That is there exists an element $r \in R$ such that I is the set of the multiples of r denoted by $\langle r \rangle$.)

- Two ideals $\langle r \rangle$ and $\langle s \rangle$ are coprime iff $\gcd(r, s) = 1$.

Then the ring isomorphism and the groups isomorphism follow from Corollary 1 by considering the ideals $I_0 = \langle m_0 \rangle, \ldots I_{n-1} = \langle m_{n-1} \rangle$. $\triangleleft$

Algorithm 4

Input: $m_0, \ldots, m_{n-1} \in R$ *pairwise coprime and* $r_0, \ldots r_{n-1} \in R$.

Output: $r \in R$ *such that* $r \equiv r_i \mod m_i$ *for* $i = 0 \cdots n-1$.

$m := m_0 \cdots m_{n-1}$.
$r := 0$
for $i = 0 \cdots n-1$ **repeat**
  $(u_i, v_i, g_i) :=$ *extendedEuclidean*$(m_i, \frac{m}{m_i})$
  if $g_i \neq 1$ *then* **error**
  $c_i := r_i v_i \text{ rem } m_i$
  $r := r + c_i \frac{m}{m_i}$
return r

PROOF ▷ Assume that the algorithm terminates without error, that is the case if every g_i is **the** gcd of m_i and $\frac{m}{m_i}$ (which are assumed to be coprime). Then, for $i = 0 \cdots n-1$ we have

$$u_i\, m_i + v_i \frac{m}{m_i} \;=\; 1 \tag{65}$$

Hence

$$r_i v_i \frac{m}{m_i} \quad \equiv \quad r_i \quad \mod m_i \tag{66}$$

and for $j = 0 \cdots n-1$ with $j \neq i$ we have

$$r_i v_i \frac{m}{m_i} \quad \equiv \quad 0 \quad \mod m_j \tag{67}$$

The specification of Algorithm 4 follow easily from Relation (66) and (67). ◁

Remark 9 It is important to observe that Algorithm 4 computes a solution r of the system of equations given by

$$r \equiv r_i \mod m_i \;\text{ for } i = 0 \cdots n-1 \tag{68}$$

Any other solution r' of (68) satisfies $r \equiv r' \mod m$ where m is the product of the moduli $m_0, \ldots, m_{r-1}$. This follows from the fact the m_i's are pairwise coprime and from Corollary 2. Therefore the set all solutions of (68) is of the form

$$\{r + k\, m \ \mid \ k \in R\} \tag{69}$$

However, in practice, we need only one solution. In the next two results (Theorem 4 and Theorem 5) by imposing

$$d(r) \ < \ d(m) \tag{70}$$

where d is the Euclidean size of R, we manage to restrict to a unique solution.

Theorem 4 *Let $R = \mathbf{k}[x]$ for a field $\mathbf{k}$. Let $m_0, \ldots, m_{r-1} \in R$ be polynomials pairwise coprime ($\gcd(m_i, m_j) = 1$ for $0 \leq i < j \leq r-1$). Let m be their product. For $0 \leq i \leq r-1$ let $d_i \geq 1$ be the degree of m_i and $n = \Sigma_{i=0}^{r-1} d_i$ be the degree of m. For $0 \leq i \leq r-1$ let $f_i \in \mathbf{k}[x]$ be a polynomial with degree $\deg(f_i) < d_i$. Then there is a unique polynomial $f \in \mathbf{k}[x]$ such that*

$$\deg(f) < n \quad \text{ and } \quad f \equiv f_i \mod m_i \ \text{ for } \ i = 0 \cdots r-1. \tag{71}$$

Moreover it can be computed in $\mathcal{O}(n^2)$ operations in $\mathbf{k}$.

PROOF $\triangleright$ Except for the complexity result (that can be found in [vzGG99] as Theorem 5.7) and the uniqueness, this theorem follows from Algorithm 4. The uniqueness follows from the constraint $\deg(f) < n$. Indeed, assume that there are two polynomials f and g solutions of (71). Then we have

$$f \equiv g \mod m_i \ \text{ for } \ i = 0 \cdots r-1. \tag{72}$$

and thus

$$f \equiv g \mod m \tag{73}$$

Hence m divides $f - g$ although $\deg(m) = n > \deg(f - g)$ holds. Therefore $f = g$. $\triangleleft$

Theorem 5 *Let $m_0, \ldots, m_{r-1}, m$ be in $R = \mathbb{Z}$ such that the m_i's are pairwise coprime and m is their product. Let n be the word length of m. Let $a_0, \ldots, a_{r-1} \in R$ be such that $0 \leq a_i < m_i$ for $i = 0 \cdots r-1$. Then there is a unique $a \in R$ such that*

$$0 \leq a < m \quad \text{ and } \quad a \equiv a_i \mod m_i \ \text{ for } \ i = 0 \cdots r-1. \tag{74}$$

Moreover it can be computed in $\mathcal{O}(n^2)$ word operations.

PROOF $\triangleright$ Except for the complexity result (that can be found in [vzGG99] as Theorem 5.8) and the uniqueness, this theorem follows from Algorithm 4. The proof of the uniqueness is quite easy to establish. $\triangleleft$

We reproduce below the ALDOR code for the Chinese Remaindering Algorithm. More precisely, the operation `interpolate` satisfies exactly the specification of Algorithm 4. After the definition of the `ChineseRemaindering` domain we prove that its operation `combine` satisifies the specification of Algorithm 4 for the case of two moduli m_1 and m_2. The reader is left with the proof of the operation `interpolate` which implements the general case (with $r \geq 2$ moduli).

```
ChineseRemaindering(E: EuclideanDomain): with {
        combine:  (E, E) -> (E, E) -> E;
        interpolate:  (List E, List E) -> E;
} == add {
        combine(m1: E, m2: E): (E, E) -> E == {
                local u1: E;
                assert(m1 = unitCanonical m1);
                assert(m2 = unitCanonical m2);
                fn(r1: E, r2: E): E == {
                        r1__new := r1 rem m1;
                        r := (r2 - r1__new) rem m2 ;
                        r :=  (r * u1) rem m2 ;
                        r := r1__new +  r * m1;
                }
                (u1, u2, g) := extendedEuclidean(m1, m2);
                fn
        }
        interpolate(lm: List E, lr: List E): E == {
                m := first lm;
                r := first lr rem m;
                for mi in rest lm for ri in rest lr repeat {
                        r := combine(m, mi)(r, ri);
                        m := m * mi;
                }
                return r;
        }
}
```

PROOF $\triangleright$ We need to prove that `combine(m1,m2)(r1,r2)` returns a qunatity congruent to that computed by Algorithm 4 modulo $m_1\, m_2$.

According to Algorithm 4 the case of two moduli m_1 and m_2 requires the following computations

- $r := 0$
- $(u_1, v_1, g_1) :=$ extendedEuclidean(m_1, m_2)
- $c_1 := r_1 v_1$ rem m_1
- $r := r + c_1 m_2$
- $(u_2, v_2, g_2) :=$ extendedEuclidean(m_2, m_1)
- $c_2 := r_2 v_2$ rem m_2
- $r := r + c_2 m_1$

This can be simplified as follows:

- $(u_1, v_1, g_1) :=$ extendedEuclidean(m_1, m_2)
- $c_1 := r_1 v_1$ rem m_1
- $c_2 := r_2 u_1$ rem m_2
- $r := c_1 m_2 + c_2 m_1$

From the relations

$$\begin{array}{l} c_1 m_2 \equiv r_1 v_1 m_2 \mod m_1 m_2, \\ c_2 m_1 \equiv r_2 u_1 m_1 \mod m_1 m_2, \\ u_1 m_1 + v_1 m_2 \ = \ 1, \end{array} \tag{75}$$

we deduce the following congruences $\mod m_1 m_2$

$$\begin{array}{rcl} r & \equiv & c_1 m_2 + c_2 m_1 \\ & \equiv & r_1 v_1 m_2 + r_2 u_1 m_1 \\ & \equiv & r_1 (1 - u_1 m_1) + r_2 u_1 m_1 \\ & \equiv & r_1 + (r_2 - r_1) u_1 m_1 \end{array} \tag{76}$$

This last expression is exactly the quantity computed by `combine(m1,m2)(r1,r2)`. This proves the implementation of the above operation `combine`. $\triangleleft$

2.3 Modular computation of the determinant

Consider a square matrix A of order n with coefficients in $\mathbb{Z}$. It is known that $\det(A)$, the determinant of A, can be computed in at most $2n^3$ operations in $\mathbb{Q}$ by means of Gaussian elimination. Let us estimate the growth of the coefficients. For simplicity, assume

- A is not singular,
- no row or column permutations are necessary,

After $k-1$ pivoting stages the current matrxi $A^{(k)}$ looks like

$$\begin{pmatrix} * & & & & & & & \\ 0 & * & & & & & & \\ & & \ddots & & & & & \\ & & 0 & * & & & & \\ & & & 0 & a_{kk}^{(k)} & \cdots & a_{kj}^{(k)} & \cdots \\ & & & \vdots & \vdots & & \vdots & \\ & & & 0 & a_{ik}^{(k)} & \cdots & a_{ij}^{(k)} & \cdots \\ & & & \vdots & \vdots & & \vdots & \end{pmatrix} \tag{77}$$

The entries of the matrix for $k < i \leq n$ and $k \leq j \leq n$ change according to the formula

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}. \tag{78}$$

We consider the following numbers.

- Let b_k be an upper bound for the absolute value of the numerators and the denominators of all $a_{ij}^{(k)}$.
- In particular for $1 \leq i, j \leq n$ we have $\mid a_{ij} \mid \leq \; b_0$.

From Relation (78) we obtain

$$b_k \;\leq\; 2b_{k-1}^4 \;\leq\; 4b_{k-2}^{4^2} \;\leq\; \cdots \;\leq\; 2^k {b_0}^{4^k} \tag{79}$$

This shows an exponential upper bound. (However a polynomial bound in b_0, n can be established but the proof is far from trivial and the formula still not very encouraging.)

In what follows, we present an approach whose goal is control the growth of the intermediate computations when calculating the determinant of A. Let d be this determinant. Let us choose a prime number $p \in \mathbb{Z}$ such that

$$p \; > \; 2 \mid d \mid \tag{80}$$

Let r be the determinant of A regarded as a matrix over $\mathbb{Z}/p\mathbb{Z}$ and let us choose for representing $\mathbb{Z}/p\mathbb{Z}$ the integers in the symmetric range

$$-\frac{p-1}{2} \cdots \frac{p-1}{2} \tag{81}$$

Hence we have

$$-\frac{p}{2} < r < \frac{p}{2} \quad \text{and} \quad -\frac{p}{2} < d < \frac{p}{2} \tag{82}$$

leading to

$$-p < d - r < p \tag{83}$$

Observe that $\det(A)$ is a polynomial in the coefficients of A. For instance with $n = 2$ we have

$$\det(A) \; = \; a_{11}\, a_{22} - a_{12}\, a_{21} \tag{84}$$

which shows that $\det(A)$ (for $n = 2$) is a polynomial in a_{11}, a_{22}, a_{12}, a_{21}. Observe also the map

$$h : \begin{array}{ccc} \mathbb{Z} & \longrightarrow & \mathbb{Z}/p\mathbb{Z} \\ x & \longrightarrow & x \mod p = \overline{x}^p \end{array} \tag{85}$$

is a ring homomorphism. In other words for every $x, y \in \mathbb{Z}$ we have

$$\overline{x+y}^p \; = \; \overline{x}^p + \overline{y}^p \quad \text{and} \quad \overline{xy}^p \; = \; \overline{x}^p \overline{y}^p \tag{86}$$

Hence for $n = 2$ we have

$$\overline{\det(A)}^p \; = \; \overline{a_{11}}^p \; \overline{a_{22}}^p - \overline{a_{12}}^p \; \overline{a_{21}}^p \tag{87}$$

More generally we have

$$\overline{\det(A)}^p \; = \; \det(A \mod p) \tag{88}$$

that is

$$d \equiv r \mod p \tag{89}$$

which means that p divides $d - r$. This with Relation (83) leads to

$$d = r \tag{90}$$

Hence the determinant of A as a matrix over $\mathbb{Z}$ is equal to the determinant of A regarded as a matrix over $\mathbb{Z}/p\mathbb{Z}$. Therefore the computation of the determinant of A as a matrix over $\mathbb{Z}$ can be done modulo p, which provides a control on the intermediate computations. Now we have to answer the following questions:

1. How to choose p?

2. What do we win?

For choosing p we need an a-priori bound for the determinant of A. This is given by the following *Hadamard's inequality.*

Theorem 6 *Let B be the maximal absolute value of an entry of A. Then we have*

$$\mid d \mid \leq \; n^{n/2}\, B^n \tag{91}$$

Example 4 Consider

$$A \;=\; \left(\begin{array}{cc} 4 & 5 \\ 6 & -7 \end{array} \right) \tag{92}$$

Gaussian elimination leads to

$$A \;=\; \left(\begin{array}{cc} 4 & 5 \\ 0 & -29/2 \end{array} \right) \tag{93}$$

Hence $\det(A) = -58$. The Hadamard's inequality gives

$$\mid \det(A) \mid \leq \; 2^1\, 7^2 = 98 \tag{94}$$

The number $p = 199$ is prime and satisfies $p > 2 \times 98$. Gaussian elimination $\bmod\ p$ leads to

$$A \;=\; \left(\begin{array}{cc} 4 & 5 \\ 0 & 85 \end{array} \right) \tag{95}$$

So $\det(A \mod p) = 141 = -58$ in $\mathbb{Z}/199\mathbb{Z}$.

Let us study what is the cost of this approach. Let us denote by C the determinant bound of Hadamard's inequality. Assume that our machine words are N-bit long. We make the following observations.

- The word length of C is in the order of magnitude of

$$\ell \quad = \quad \lceil \frac{1}{N} \log_2(C) \rceil + 1 \quad = \quad \lceil \frac{1}{N}\, n\, (\frac{1}{2} \log_2 n + \log_2 B) \rceil + 1. \qquad (96)$$

- Prime numbers are frequent enough to find one with a word length in the same order of magnitude as C.

- So each element of $\mathbb{Z}/p\mathbb{Z}$ can be coded by an array with at most $\mathcal{O}(\ell)$ words.

- Hence, each operation (like addition, multiplication, inverse computation) in $\mathbb{Z}/p\mathbb{Z}$ costs at most $\mathcal{O}(\ell^2)$ word operations.

- Gaussian elimination $\mod p$ will require $\mathcal{O}(n^3)$ operations in $\mathbb{Z}/p\mathbb{Z}$.

Therefore we have proved the following theorem

Theorem 7 *The determinant of a square matrix with order n, coefficients in $\mathbb{Z}$ and B as the maximal absolute value of a coefficient can be computed in $\mathcal{O}(n^3\, n^2\, (\log n + \log B)^2)$ word operations.*

This is not in fact a big progress w.r.t. Gaussian elimination over $\mathbb{Q}$. But this can be improved using a *small primes modular computation* as follows.

Algorithm 5

Input: $A = (a_{ij})$ a square matrix over $\mathbb{Z}$ of order n with $\mid a_{ij} \mid \leq B$ for $1 \leq i, j \leq n$.

Output: $\det(A) \in \mathbb{Z}$.

$C := n^{n/2}\, B^n$
$r := \lceil \log_2(2\, C + 1) \rceil$
choose r distinct prime numbers $m_0, \ldots, m_{r-1} \in \mathbb{N}$
for $i = 0 \cdots r-1$ **repeat**
  $d_i := \det(A \mod m_i)$
$d := \textit{interpolate}([m_0, \ldots, m_{r-1}], [d_0, \ldots, d_{r-1}])$
$d := d \text{ rem } m$
if $d \geq \frac{m}{2}$ **then**
  $d := d - m$
return(d)

Proof $\triangleright$ Recall that $\det(A)$ is a polynomial expression in the coefficients of A. Hence by using the ring homomorphism between $\mathbb{Z}$ and $\mathbb{Z}/m_i\mathbb{Z}$ for $i = 0 \cdots r-1$ we have

$$\det(A) \equiv d_i \mod m_i \tag{97}$$

Using the ring isomorphism of Corollary 2

$$\mathbb{Z}/m \;\simeq\; \mathbb{Z}/m_0 \times \cdots \times \mathbb{Z}/m_{r-1} \tag{98}$$

we deduce

$$\det(A) \equiv d \mod m \tag{99}$$

where m is the product of the moduli $m_0, \ldots, m_{r-1}$. Now observe that

$$\begin{array}{rcl} m & = & m_0 \cdots m_{r-1} \\ & \geq & 2^r \\ & \geq & 2\,C + 1 \\ & \geq & 2\,n^{n/2}\,B^n \\ & \geq & 2\mid d \mid \end{array} \tag{100}$$

Hence actually we have $\det(A) = d$. $\triangleleft$

Example 5 Consider again

$$A \;=\; \left(\begin{array}{cc} 4 & 5 \\ 6 & -7 \end{array} \right) \tag{101}$$

We choose the four primes $2, 3, 5, 7$ so that $m = 210$. We get

$$\begin{array}{cc} \det(A) \equiv 0 \mod 2 & \det(A) \equiv 2 \mod 3 \\ \det(A) \equiv 2 \mod 5 & \det(A) \equiv -2 \mod 7 \end{array} \tag{102}$$

The solutions of the system $d \equiv d_i \mod m_i$ for $1 \leq i \leq 4$ are in

$$-58 + 210\mathbb{Z} \;=\; \{\ldots, -268, -58, 152, 362, \ldots\} \tag{103}$$

Finally $\det(A) = -58$ again.

Based on this small primes modular computation approach one could prove the following theorem

Theorem 8 *The determinant of a square matrix with order n, coefficients in $\mathbb{Z}$ and B as the maximal absolute value of a coefficient can be computed in $\mathcal{O}(n^4 \log^2(n\,B)(\log^2 n + \log^2 B))$ word operations.*

PROOF $\triangleright$ See Theorem 5.12 in [vzGG99] $\triangleleft$

Remark 10 With Algorithm 5 we achieve the following goals.

- All intermediate computations can be made modulo small prime numbers. In practice these small primes are machine integers allowing fast computations.

- The only possible large value is the determinant itself.

- The space and the time required for the whole computation can be estimated in advance.

Moreover the computations of the modular determinants (the d_i's) are pairwise independent and thus can be distributed.

References

[BL82] Bruno Buchberger and Rüdiger Georg Konrad Loos. *Algebraic Simplification*, pages 11–43. Springer Verlag, Wien–New York, 1982.

[DST93] J. Davenport, Y. Siret, and E. Tournier. *Calcul Formel*. Masson, 1993. 2^e édition révisée.

[GCL92] K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluver Acadenic Publishers, 1992.

[van91] B.L. van der Waerden. *Algebra*. Springer-Verlag, 1991. seventh edition.

[vzGG99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.