

CS810A/MA810A Computer Algebra

Alex Myasnikov and Werner Backes

Stevens Institute of Technology

CS810A/MA810A 2008

- Modular algorithms
 - Evaluation and interpolation.
 - The Chinese Remainder algorithm.
 - Modular determinant computation.

Modern Computer Algebra (2nd edition), by Joachim von zur Gathen and Jürgen Gerhard, Pages 95 – 111

The slides are not a substitute for reading the book!

Evaluation and interpolation

- Simplest but most important change of representation: evaluation and interpolation.
- Evaluation of $f = \sum_{0 \leq i \leq n} f_i x^i \in F[x]$ at point $u \in F$ (field) using Horner's rule:

$$f(u) = (\dots (f_{n-1}u + f_{n-2})u + \dots + f_1)u + f_0$$

- Needs $n - 1$ multiplications and $n - 1$ additions in F
- Evaluate points $u_1, \dots, u_{n-1} \in F$ in $2n^2 - 2n$ operations in F .
- What about interpolation?

Evaluation and interpolation

- The Lagrange interpolant

$$l_i = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - u_j}{u_i - u_j} \in F[x]$$

- Property that $l_i(u_j)$ is 0 for $i \neq j$ and 1 when $i = j$.
- For arbitrary $v_0, \dots, v_{n-1} \in F$,

$$f = \sum_{0 \leq i < n} v_i l_i = \sum_{0 \leq i < n} v_i \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - u_j}{u_i - u_j}$$

is a polynomial with $\deg f < n$ such that $f(u_i) = v_i$ for all i .

- Interpolation polynomial is unique (given $\deg f < n$).

Theorem

Evaluating a polynomial $f \in F[x]$ of degree less than n at n distinct points $u_0, \dots, u_{n-1} \in F$ or computing an interpolating polynomial at these points can be performed with $O(n^2)$ operations in F .

More precisely, evaluation takes $2n^2 - 2n$ operations and Lagrange interpolation uses $7n^2 - 8n + 1$ operations.

- Later we see, that evaluation and interpolation can be done in $O(n \log^2 n \log \log n)$.

Evaluation and interpolation

Global view on evaluation and interpolation.

- Given n points $u_0, \dots, u_{n-1} \in F$, we define the evaluation map $\chi : F^n \rightarrow F^n$ by

$$\chi(f_0, \dots, f_{n-1}) = \left(\sum_{0 \leq j < n} f_j u_0^j, \dots, \sum_{0 \leq j < n} f_j u_{n-1}^j \right)$$

- Map corresponding to evaluation of polynomials f with $\deg f < n$ at points u_0, \dots, u_{n-1} .
- χ function of coefficients f_j ; different evaluation points u_i give different functions.

Evaluation and interpolation

- Map χ is F -linear and represented by Vandermonde matrix.

$$V = VDM(u_0, \dots, u_{n-1}) = \begin{pmatrix} 1 & u_0 & u_0^2 & \dots & u_0^{n-1} \\ 1 & u_1 & u_1^2 & \dots & u_1^{n-1} \\ 1 & u_2 & u_2^2 & \dots & u_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & u_{n-1} & u_{n-1}^2 & \dots & u_{n-1}^{n-1} \end{pmatrix}$$

- V singular, if $u_i = u_j$ for $i \neq j$ (row i equal to row j).
- u_i distinct, unique interpolation polynomial exists (deg $< n$), shows that χ (hence V) invertible.
- Evaluation and interpolation are linear maps between coefficients and value vectors (not linear in u_0, \dots, u_{n-1}).

The Chinese Remainder Algorithm

- R is Euclidean domain. Fix following notation:

$$\begin{aligned} m_0, \dots, m_{r-1} \in R \text{ pairwise coprime, s.t. } \gcd(m_i, m_j) = 1 \\ \text{for } 0 \leq i < j < r \text{ and } m = m_0 \cdots m_{r-1} \end{aligned} \quad (1)$$

- $m = \text{lcm}(m_0, \dots, m_{r-1})$. We have the canonical ring homomorphism for $1 \leq i < r$

$$\begin{aligned} \pi_i : R &\longrightarrow R/\langle m_i \rangle \\ f &\longmapsto f \bmod m_i \end{aligned}$$

- Combine all π_i we get ring homomorphism

$$\begin{aligned} \chi = \pi_0 \times \cdots \times \pi_{r-1} : R &\longrightarrow R/\langle m_0 \rangle \times \cdots \times R/\langle m_{r-1} \rangle \\ f &\longmapsto (f \bmod m_0, \dots, f \bmod m_{r-1}) \end{aligned}$$

The Chinese Remainder Algorithm

- Example: $R = \mathbb{Z}$, $r = 2$, $m_0 = 11$, $m_1 = 13$, $m = 143$, and

$$\begin{aligned}\chi(f) &= (f \bmod 11, f \bmod 13) \\ &= (2 \bmod 11, 7 \bmod 13) \in \mathbb{Z}_{11} \times \mathbb{Z}_{13}\end{aligned}$$

Does this define f uniquely?

Theorem

χ is surjective with kernel $\langle m \rangle$.

The Chinese Remainder Algorithm

Proof.

Let $f \in R$. Then

$$f \in \ker \chi \iff \chi(f) = (f \bmod m_0, \dots, f \bmod m_{r-1}) = (0, \dots, 0)$$

$$\iff m_i \mid f \text{ for } 0 \leq i < r$$

$$\iff \text{lcm}(m_0, \dots, m_{r-1}) \mid f \iff m \mid f$$

and $\ker \chi = \langle m \rangle$.

For surjectivity it is sufficient to show that a Lagrange interpolant $l_i \in R$ with $\chi(l_i) = e_i$ exists for $0 \leq i < r$, where $e_i = (0, \dots, 0, 1, 0, \dots, 0) \in R/\langle m_0 \rangle \times \dots \times R/\langle m_{r-1} \rangle$ denotes the i th unit vector. □

The Chinese Remainder Algorithm

Proof (continued).

Why is this enough?

$v = (v_0 \bmod m_0, \dots, v_{r-1} \bmod m_{r-1}) \in R/\langle m_0 \rangle, \dots, R/\langle m_{r-1} \rangle$ be arbitrary, with $v_0, \dots, v_{r-1} \in R$. Then

$$\begin{aligned}\chi\left(\sum_{0 \leq i < r} v_i l_i\right) &= \sum_{0 \leq i < r} \chi(v_i) \chi(l_i) \\ &= \sum_{0 \leq i < r} (v_i \bmod m_0, \dots, v_i \bmod m_{r-1}) \cdot e_i \\ &= \sum_{0 \leq i < r} (0, \dots, 0, v_i \bmod m_i, 0, \dots, 0) = v\end{aligned}$$



The Chinese Remainder Algorithm

Proof (continued).

Assume $i = 0$ for simplicity. The EEA applied to $m_1 \cdots m_r - 1 = m/m_0$ and m_0 computes $s, t \in R$ with $sm/m_0 + tm_0 = 1 = \gcd(m/m_0, m_0)$.

If we let $l_0 = sm/m_0$, then obviously $l_0 \equiv 0 \pmod{m_j}$ for $1 \leq j < r$, and

$$l_0 = s \frac{m}{m_0} \equiv s \frac{m}{m_0} + tm_0 = 1 \pmod{m_0},$$

so that $\chi(l_0) = e_0$ as claimed. □

The Chinese Remainder Algorithm

Corollary (Chinese Remainder Theorem - CRT)

We have the ring isomorphism

$$(1) \quad R/\langle m \rangle \cong R/\langle m_0 \rangle \times \cdots \times R/\langle m_{r-1} \rangle$$

and the group isomorphism of the multiplicative groups

$$(2) \quad (R/\langle m \rangle)^\times \cong (R/\langle m_0 \rangle)^\times \times \cdots \times (R/\langle m_{r-1} \rangle)^\times$$

Proof.

Earlier theorems (s.o. and intro) imply (1). For $f \in R$ we have

$$\begin{aligned} f \text{ invertible mod } m &\iff \gcd(f, m) = 1 \iff \gcd(f, m_i) = 1 \quad \forall i \\ &\iff f \text{ invertible mod } m_i \quad \forall i, 0 \leq i < r, \end{aligned}$$

and (2) follows. □

The Chinese Remainder Algorithm

- Proof for the theorem (s.o.) is constructive and leads to the following algorithm.

Algorithm (Chinese Remainder Algorithm - CRA)

INPUT: $m_0, \dots, m_{r-1} \in R$ pairwise coprime, $v_0, \dots, v_{r-1} \in R$,
 R a Euclidean domain.

OUTPUT: $f \in R$, such that $f \equiv v_i \pmod{m_i}$ for $0 \leq i < r$

- (1) $m \leftarrow m_0 \cdots m_{r-1}$
- (2) **for** $(0 \leq i < r)$ **do**
- (3) compute $\frac{m}{m_i}$
- (4) call the EEA to compute $s_i, t_i \in R$ with $s_i \frac{m}{m_i} + t_i m_i = 1$
- (5) $c_i \leftarrow v_i s_i \pmod{m_i}$
- (6) **return** $\sum_{0 \leq i < r} c_i \frac{m}{m_i}$

The Chinese Remainder Algorithm

- c_i is the remainder in R on dividing $v_i s_i$ by m_i .
- $c_i m / m_i \equiv 0 \pmod{m_j}$ for all $j \neq i$
- $c_i m / m_i \equiv v_i s_i m / m_i \equiv v_i \pmod{m_i}$
- Hence $f \equiv c_i m_i \equiv v_i \pmod{m_i}$ for all $0 \leq i < r$ as claimed.

- Example: Let $R = \mathbb{Z}$, $m_i = p_i^{e_i}$ for $0 \leq i < r$, where $p_i \in \mathbb{N}$ are distinct primes and $e_i \in \mathbb{N}_{>0}$ for $0 \leq i < r$. Then

$$m = \prod_{0 \leq i < r} p_i^{e_i}$$

is the prime decomposition of $m \in \mathbb{Z}$. The CRT tells us that

The Chinese Remainder Algorithm

- Example (continued):

$$\mathbb{Z}/\langle m \rangle \cong \mathbb{Z}/\langle p_0^{e_0} \rangle \times \cdots \times \mathbb{Z}/\langle p_{r-1}^{e_{r-1}} \rangle$$

and for $v_0, \dots, v_{r-1} \in \mathbb{Z}$ the CRA computes solution $f \in \mathbb{Z}$ for

$$f \equiv v_i \pmod{p_i^{e_i}} \quad \text{for } 0 \leq i < r.$$

Concrete example:

Take $r = 2$, $m_0 = 11$, $m_1 = 13$, and $m = 11 \cdot 13 = 143$.

Find $f \in \mathbb{Z}$, $0 \leq f < m$ for $v_0 = 2$ and $v_1 = 7$.

The Chinese Remainder Algorithm

- Example (continued):

EEA gives us $6 \cdot 13 + (-7) \cdot 11 = 1$ with $s_0 = 6$ and $s_1 = -7$. Lagrange interpolants l_0 and l_1 do not occur explicitly in the CRA. $l_0 = 6 \cdot 13 = 78$ and $l_1 = (-7) \cdot 11 = -77$. In fact $l_0 \equiv 1 \pmod{11}$, $l_0 \equiv 0 \pmod{13}$, $l_1 \equiv 0 \pmod{11}$, $l_1 \equiv 1 \pmod{13}$.

$$c_0 = v_0 s_0 \pmod{m_0} = 2 \cdot 6 \pmod{11} = 1,$$

$$c_1 = v_1 s_1 \pmod{m_1} = 7 \cdot (-7) \pmod{13} = 3,$$

Finally we compute

$$f = c_0 \frac{m}{m_0} + c_1 \frac{m}{m_1} = 1 \cdot 13 + 3 \cdot 11 = 46,$$

and indeed $46 = 4 \cdot 11 + 2 = 3 \cdot 13 + 7$.

The Chinese Remainder Algorithm

- With $R = \mathbb{Z}$ and the m_i as in the example above, we obtain the following formula for Euler's totient function and the CRT.

Corollary

If $m = p_0^{e_0} \cdot p_{r-1}^{e_{r-1}}$, with distinct primes $p_0, \dots, p_{r-1} \in \mathbb{N}$ and $e_0, \dots, e_{r-1} \in \mathbb{N}_{>0}$, then

$$\begin{aligned}\varphi(m) &= (p_0 - 1)p_0^{e_0-1} \cdots (p_{r-1} - 1)p_{r-1}^{e_{r-1}-1} \\ &= m \cdot \prod_{p|m, p \text{ prime}} \left(1 - \frac{1}{p}\right).\end{aligned}$$

The Chinese Remainder Algorithm

- Example: $R = F[x]$, F field, $m_i = x - u_i$ for $0 \leq i < r$.
 u_0, \dots, u_{r-1} in F pairwise distinct. Then
 $f \equiv f(u_i) \pmod{(x - u_i)}$ for $1 \leq i < r$ and arbitrary $f \in F$, and
hence the ring homomorphism

$$\begin{aligned}\chi : F[x] &\longrightarrow F[x]/\langle x - u_0 \rangle \times \cdots \times F[x]/\langle x - u_{r-1} \rangle \cong F^r \\ f &\longmapsto (f(u_0), \dots, f(u_{r-1}))\end{aligned}$$

is just the evaluation homomorphism at u_0, \dots, u_{r-1} .

Moreover, the l_i satisfying $l_i \equiv l_i(u_i) = 1 \pmod{(x - u_i)}$, and
 $l_i \equiv l_i(u_j) = 0 \pmod{(x - u_j)}$ for $j \neq i$, and $\deg l_i < r$ are the
Lagrange interpolants

$$l_i = \prod_{\substack{0 \leq j < r \\ j \neq i}} \frac{x - u_j}{u_i - u_j}.$$

The Chinese Remainder Algorithm

- Example (continued): If $v_0, \dots, v_{r-1} \in F$ are scalars, then $c_i = v_i s_i$ in the CRA, and

$$f = \sum_{0 \leq i < r} c_i \frac{m}{m_i} = \sum_{0 \leq i < r} v_i l_i$$

is the Lagrange interpolation polynomial satisfying $f(u_i) = v_i$ for $0 \leq i < r$. Thus Chinese remaindering for r distinct monic linear polynomials is the same as interpolation at r points.

- CRT tells us once more that interpolation polynomial is unique modulo m . Exactly one polynomial $f \in F[x]$ with $\deg f < r$ solves interpolation problem.
- It is useful to think of CRT as generalization of interpolation.

The Chinese Remainder Algorithm

Theorem

Let $R = F[x]$ for a field F , $m_0, \dots, m_{r-1}, m \in R$ (pairwise coprime, $m = m_0 \cdots m_{r-1}$), $d_i = \deg m_i \geq 1$ for $0 \leq i < r$, $n = \deg m - \sum_{0 \leq i < r} d_i$ and $v_i \in R$ with $\deg v_i < d_i$. Then the unique solution $f \in F[x]$ with $\deg f < n$ of the Chinese Remainder Problem

$$f \equiv v_i \pmod{m_i}, \quad \text{for } 0 \leq i < r$$

for polynomials can be computed in $O(n^2)$ operations in F .

Proof.

Idea: Check every step of the CRA. □

The Chinese Remainder Algorithm

- The following theorem is the integer analog:

Theorem

Let $R = \mathbb{Z}$, $m_0, \dots, m_{r-1}, m \in N$ (pairwise coprime, $m = m_0 \cdots m_{r-1}$), $n = \lfloor \log_2 m / 64 \rfloor + 1$ the word length of m , and $v_i \in Z$ such that $0 \leq v_i < m_i$ for $0 \leq i < r$. Then the unique solution $f \in \mathbb{Z}$ with $0 \leq f < m$ of the Chinese Remainder Problem

$$f \equiv v_i \pmod{m_i}, \quad \text{for } 0 \leq i < r$$

for integers can be computed using $O(n^2)$ word operations.

Modular determinant computation

- Use CRT to compute the determinant $\det A \in \mathbb{Z}$ for $n \times n$ matrix $A = (a_{ij})_{1 \leq i, j \leq n} \in \mathbb{Z}^{n \times n}$.
- Can be solved by Gaussian elimination over \mathbb{Q} which costs $2n^3$ operations in \mathbb{Q} .
- Polynomial in input size, but the number of word operations depends on numerators and denominators of intermediate results.
- How large can these grow?
- (Length of interm. results and number of word operations for Gaussian elimination over \mathbb{Q} in fact polynomial in input size)

Modular determinant computation

- Simplest way to obtain polynomial time for $d = \det A$ of matrix $A \in \mathbb{Z}^{n \times n}$ is choosing $p > 2|d|$ prime and perform Gaussian elimination on $A \bmod p \in \mathbb{Z}_p^{n \times n}$ to calculate $d \bmod p$ and represent this value in symmetric system.

$$-\frac{p-1}{2}, \dots, \frac{p-1}{2}$$

of representatives. If $r \in \mathbb{Z}$ is this representative, then

$$r \equiv d \pmod{p}, \quad -\frac{p}{2} < r < \frac{p}{2}.$$

Modular determinant computation

- Congruence holds since any polynomial expression like the determinant commutes with the canonical homomorphism $\mathbb{Z} \rightarrow \mathbb{Z}_p$, so that the determinant taken modulo p , of A equals the determinant in \mathbb{Z}_p of the matrix $(A \bmod p) \in \mathbb{Z}_p^{n \times n}$. It follows that p divides $d - r$,

$$|d - r| \leq |d| + |r| < \frac{p}{2} + \frac{p}{2} = p,$$

and hence $d = r$.

- Trivial but useful fact used:
If $a \mid b$ and $|b| < |a|$, then $b = 0$, for $a, b \in \mathbb{Z}$.

Modular determinant computation

- Calculation of $\det(A \bmod p)$ essentially the same as Gaussian elimination over \mathbb{Q} .
- Dividing by a pivot element a we compute inverse modulo p i (Extended Euclidean Algorithm).
- We do not have to worry about the size of intermediate results.
- Hadamard's inequality gives us bound on $\det A$:

$$|\det A| \leq n^{\frac{n}{2}} B^n,$$

where $B = \max_{1 \leq i, j \leq n} |a_{ij}| \in \mathbb{N}$ the maximal absolute value of an entry in A .

Modular determinant computation

- Example: Let $A = \begin{pmatrix} 4 & 5 \\ 6 & -7 \end{pmatrix}$. After Gaussian elimination it has the form $\begin{pmatrix} 4 & 5 \\ 0 & \frac{-29}{2} \end{pmatrix}$ so that $\det A = -58$.

Hadamard's bound is $|\det A| \leq 2^1 7^2 = 98$. We choose $p = 199 > 2 \cdot 98$ and perform Gaussian elimination modulo p (inverse of pivot element 4 is 50) and

$$\det(A \bmod 199) = \det \begin{pmatrix} 4 & 5 \\ 0 & 85 \end{pmatrix} = 141 = -58 \in \mathbb{Z}_{199}.$$

- Idea: Not compute with single modulus, but with several moduli at a time: small primes modular computation.

Modular determinant computation

Algorithm (Small primes modular determinant computation)

INPUT: $A = (a_{ij})_{1 \leq i, j \leq n} \in \mathbb{Z}^{n \times n}$, with $|a_{ij}| \leq B$ for all i, j

OUTPUT: $\det A \in \mathbb{Z}$

- (1) $C \leftarrow n^{\frac{n}{2}} B^n$, $r \leftarrow \lceil \log_2(2C + 1) \rceil$
- (2) choose r distinct prime numbers $m_0, \dots, m_{r-1} \in \mathbb{N}$
- (3) **for** $(0 \leq i < r)$ **do**
- (4) compute $A \bmod m_i$
- (5) **for** $(0 \leq i < r)$ **do**
- (6) compute $d_i \in \{0, \dots, m_i - 1\}$ such that $d_i \equiv \det A \bmod m_i$
 using Gaussian elimination over \mathbb{Z}_{m_i}
- (7) call CRA to determine $d \in \mathbb{Z}$ of least absolute value with $d \equiv d_i \bmod m_i$ for $0 \leq i < r$
- (8) **return** d

Modular determinant computation

- Example: Let $A = \begin{pmatrix} 4 & 5 \\ 6 & -7 \end{pmatrix}$. First four prime moduli:

$$\det A \equiv 0 \pmod{2}$$

$$\det A \equiv 2 \pmod{5}$$

$$\det A \equiv 2 \pmod{3},$$

$$\det A \equiv -2 \pmod{7}.$$

We have $m = 2 \cdot 3 \cdot 5 \cdot 7 = 210$. Solutions to Chinese Remainder Problem $d \equiv d_i \pmod{m_i}$ for $1 \leq i \leq 4$ are $d \in -58 + 210\mathbb{Z} = \{\dots, -268, -58, 152, 362, \dots\}$.

Correct solution is -58 (least absolute value).

Theorem

The determinant of a matrix $A \in \mathbb{Z}^{n \times n}$ with all entries less than B in absolute value can be computed deterministically with

$$O((n^4 \log(nB) + n^3 \log^2(nB))(\log^2 n + (\log \log B)^2)) \text{ or} \\ \tilde{O}(n^4 \log B + n^3 \log^2 B)$$

word operations.

- Computing determinants for matrices with entries in $F[x]$, where F is a field, can be done similarly to the integer case.