Computer Algebra

Sebastian Iwanowski
FH Wedel

3. Modular Arithmetic
3.1 Computation of modular functions

References for repetition and deepening your knowledge:

Köpf 4 (except for 4.3, 4.6) (in German)
von zur Gathen 4.1, 4.2

most of the slides are translations of a seminar presentation of Hendrik Annuth

Computer Algebra 3

Residue classes

The most used residue class ever:

There are 12 equivalence classes for time:

$$\mathbb{Z}_{12} = \{[0]_{12};[1]_{12};[2]_{12};[3]_{12};[4]_{12};[5]_{12};\\ [6]_{12};[7]_{12};[8]_{12};[9]_{12};[10]_{12};[11]_{12}\}$$

The time 0:00, 12:00 and 24:00 are called
by the same spoken time (12 o´clock).
They are in the same equivalence class forming a residue class in the ring of residues.

Computer Algebra 3

Residue classes

Computing with residue classes

$|\mathbb{Z}_{12}|=12$

$\mathbb{Z}_{12} = \{[0]_{12};[1]_{12};[2]_{12};[3]_{12};[4]_{12};[5]_{12};$
$[6]_{12};[7]_{12};[8]_{12};[9]_{12};[10]_{12};[11]_{12}\}$

$[0]_{12} = \{...;-24;-12;0;12;24;...\}$
$[8]_{12} = \{...;-16;-4;8;20;32;...\}$

$[8]_{12}+[11]_{12}=[7]_{12}$, because $8 + 11 = 19 = 12*1 + 7 \Rightarrow 19 \in [7]_{12}$

$[4]_{12}*[8]_{12}=[8]_{12}$, because $4 * 8 = 32 = 12*2 + 8 \Rightarrow 32 \in [8]_{12}$

Computer Algebra 3

Residue classes

$$[x]_{12} * [2]_{12} = [10]_{12}$$

Search for $[„10/2“]_{12}$

$[2]_{12}*[0]_{12}= [0]_{12}$
$[2]_{12}*[1]_{12}= [2]_{12}$
$[2]_{12}*[2]_{12}= [4]_{12}$
$[2]_{12}*[3]_{12}= [6]_{12}$
$[2]_{12}*[4]_{12}= [8]_{12}$
$[2]_{12}*[5]_{12}= [10]_{12}$
$[2]_{12}*[6]_{12}= [0]_{12}$
$[2]_{12}*[7]_{12}= [2]_{12}$
$[2]_{12}*[8]_{12}= [4]_{12}$
$[2]_{12}*[9]_{12}= [6]_{12}$
$[2]_{12}*[10]_{12}=[8]_{12}$
$[2]_{12}*[11]_{12}=[10]_{12}$

Two solutions found, but only by testing:

$$[2]_{12}*[x]_{12} = [7]_{12}$$

Modular division is not known to be solved efficiently. And the operation „*2“ is not invertable for all operands in $\mathbb{Z}_{12}$

Computer Algebra 3

Residue classes

Example for a residue class set $\mathbb{Z}_p$
with unique and well-defined division:

$(\mathbb{Z}_7, +)$	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

$(\mathbb{Z}_7, *)$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

In $\mathbb{Z}_7$ the operation „*“ is invertable for all operands except for 0, but is this also possible in an efficient way, i.e. other than testing ?

Computer Algebra 3

Residue classes

Division via determining the inverse element:

$(\mathbb{Z}_7, *)$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

$$2 * x \equiv 10 \pmod 7$$ x = ??

Determining the inverse element of 2:

$$a \equiv 2 \pmod 7$$

$$a^{-1} \equiv 4 \pmod 7$$

$$x \equiv 10 * a^{-1} \pmod 7$$

$$x \equiv 10 * 4 \equiv 40 \equiv 5 \pmod 7$$

Computer Algebra 3

Residue classes

Division via determining the inverse element:

- Inverse elements may be determined via the extended Euclidean algorithm:

  computes a^{-1} mod n, whenever gcd(a,n) = 1
  $\Rightarrow$ works for all $a \in \mathbb{Z}_n$, if n is a prime number

  run time: $O(\#n^2)$

Summary:

- Modular multiplication is always efficient.
- Modular division is efficient for prime modules
- Modular division for composed modules $n = p \cdot q$:
  gcd(a,n) = 1: efficient
  gcd(a,n) > 1: no efficient algorithm known!

Computer Algebra 3

Other discrete functions

The following operations are analysed in detail:

Taking powers

Computing square roots

Computing logarithms

Computer Algebra 3

Taking powers

Powers in $\mathbb{Z}_7$

(mod 7) is not mentioned for the sake of readability

$1^1 \equiv 1$	$2^1 \equiv 2$	$3^1 \equiv 3$	$4^1 \equiv 4$	$5^1 \equiv 5$	$6^1 \equiv 6$
$1^2 \equiv 1$	$2^2 \equiv 4$	$3^2 \equiv 2$	$4^2 \equiv 2$	$5^2 \equiv 4$	$6^2 \equiv 1$
$1^3 \equiv 1$	$2^3 \equiv 1$	$3^3 \equiv 6$	$4^3 \equiv 1$	$5^3 \equiv 6$	$6^3 \equiv 6$
$1^4 \equiv 1$	$2^4 \equiv 2$	$3^4 \equiv 4$	$4^4 \equiv 4$	$5^4 \equiv 2$	$6^4 \equiv 1$
$1^5 \equiv 1$	$2^5 \equiv 4$	$3^5 \equiv 5$	$4^5 \equiv 2$	$5^5 \equiv 3$	$6^5 \equiv 6$
$1^6 \equiv 1$	$2^6 \equiv 1$	$3^6 \equiv 1$	$4^6 \equiv 1$	$5^6 \equiv 1$	$6^6 \equiv 1$

We do not get all residues as a result

generating elements

appearently, $a^{(7-1)}=1$

Computer Algebra 3

Taking powers

Fermat‘s little theorem

Let p $\in \mathbb{P}$, then:

$$x^{(p-1)} \equiv 1 \,(\mathrm{mod}\ p)$$

Proof by induction over x (taking an arbitrary p):

1. $x^{(p-1)} \equiv 1 \,(\mathrm{mod}\ p) \,|\, {*x}$

   Assertion: $x^{p} \equiv x \,(\mathrm{mod}\ p)$

2. Base: Let x be 0 $\quad 0^{p} \equiv 0 \,(\mathrm{mod}\ p)$

3. Conclusion: $x^{p} \equiv x \,(\mathrm{mod}\ p) \Rightarrow (x+1)^{p} \equiv x+1 \,(\mathrm{mod}\ p)$

Computer Algebra 3

Taking powers

to prove: $(x+1)^p \equiv x+1$

$$(x+1)^p \equiv x^p + \binom{p}{1} x^{p-1} + \binom{p}{2} x^{p-2} + \ldots + \binom{p}{p-1} x^1 + 1 \pmod{p}$$

$$\binom{p}{k} = \frac{p!}{(p-k)!*k!} = \frac{p*(p-1)!}{(p-k)!*k!}$$

$$(x+1)^p \equiv x^p + p*\left(\frac{(p-1)!}{(p-1)!*1!} x^{p-1} + \frac{(p-1)!}{(p-2)!*2!} x^{p-2} + \ldots + \frac{(p-1)!}{1!*(p-1)!} x^1 \right) + 1 \pmod{p}$$

$$(x+1)^p \equiv x^p + 1 \pmod{p} \wedge x^p \equiv x \pmod{p} \Rightarrow \underline{\underline{(x+1)^p \equiv x+1 \pmod{p}}}$$

q.e.d.

Computer Algebra 3

Computing square roots

Square roots in $\mathbb{Z}_7$

$$\sqrt{4} \equiv 2 (\mathrm{mod}\ 7) \qquad \wedge \sqrt{4} \equiv (7-2) \equiv 5 (\mathrm{mod}\ 7)$$

$$5^2 = 25 = 3 * 7 + \underline{\underline{4}}$$

$$-2 \in [5]_7$$

$$x^2 \equiv (x-p)^2 \equiv x^2 - 2xp + p^2 \equiv x^2 - p(2x-p) \equiv x^2 (\mathrm{mod}\ p)$$

$$\sqrt{1} \equiv \{1;6\}; \sqrt{2} \equiv \{3;4\}; \sqrt{3} \equiv ?; \sqrt{4} \equiv \{2;5\}; \sqrt{5} \equiv ?; \sqrt{6} \equiv ? (\mathrm{mod}\ 7)$$

Computer Algebra 3

Computing square roots

Square roots in $\mathbb{Z}_7$

(mod 7) is not mentioned for the sake of readability

$1^1 \equiv 1$	$2^1 \equiv 2$	$3^1 \equiv 3$	$4^1 \equiv 4$	$5^1 \equiv 5$	$6^1 \equiv 6$
$1^2 \equiv 1$	$2^2 \equiv 4$	$3^2 \equiv 2$	$4^2 \equiv 2$	$5^2 \equiv 4$	$6^2 \equiv 1$
$1^3 \equiv 1$	$2^3 \equiv 1$	$3^3 \equiv 6$	$4^3 \equiv 1$	$5^3 \equiv 6$	$6^3 \equiv 6$
$1^4 \equiv 1$	$2^4 \equiv 2$	$3^4 \equiv 4$	$4^4 \equiv 4$	$5^4 \equiv 2$	$6^4 \equiv 1$
$1^5 \equiv 1$	$2^5 \equiv 4$	$3^5 \equiv 5$	$4^5 \equiv 2$	$5^5 \equiv 3$	$6^5 \equiv 6$
$1^6 \equiv 1$	$2^6 \equiv 1$	$3^6 \equiv 1$	$4^6 \equiv 1$	$5^6 \equiv 1$	$6^6 \equiv 1$

$$\sqrt{1} \equiv \{1;6\}; \sqrt{2} \equiv \{3;4\}; \sqrt{3} \equiv ?; \sqrt{4} \equiv \{2;5\}; \sqrt{5} \equiv ?; \sqrt{6} \equiv ? \pmod 7$$

$a^{(p-1)/2} \equiv 1 \pmod p \Rightarrow a$ has got a square root?

Computer Algebra 3

Computing square roots

Square roots in $\mathbb{Z}_7$

(mod 7) is not mentioned for the sake of readability

$1^1 \equiv 1$	$2^1 \equiv 2$	$3^1 \equiv 3$	$4^1 \equiv 4$	$5^1 \equiv 5$	$6^1 \equiv 6$
$1^2 \equiv 1$	$2^2 \equiv 4$	$3^2 \equiv 2$	$4^2 \equiv 2$	$5^2 \equiv 4$	$6^2 \equiv 1$
$1^3 \equiv 1$	$2^3 \equiv 1$	$3^3 \equiv 6$	$4^3 \equiv 1$	$5^3 \equiv 6$	$6^3 \equiv 6$
$1^4 \equiv 1$	$2^4 \equiv 2$	$3^4 \equiv 4$	$4^4 \equiv 4$	$5^4 \equiv 2$	$6^4 \equiv 1$
$1^5 \equiv 1$	$2^5 \equiv 4$	$3^5 \equiv 5$	$4^5 \equiv 2$	$5^5 \equiv 3$	$6^5 \equiv 6$
$1^6 \equiv 1$	$2^6 \equiv 1$	$3^6 \equiv 1$	$4^6 \equiv 1$	$5^6 \equiv 1$	$6^6 \equiv 1$

$$\sqrt{1} \equiv \{1;6\}; \sqrt{2} \equiv \{3;4\}; \sqrt{3} \equiv ?; \sqrt{4} \equiv \{2;5\}; \sqrt{5} \equiv ?; \sqrt{6} \equiv ? \pmod 7$$

$a^{(p-1)/2} \equiv (p-1) \pmod p \Rightarrow a$ has not got a square root?

Computer Algebra 3

Computing square roots

$$a^{(p-1)/2} \equiv 1 (\mathrm{mod}\ p) \Rightarrow a$$ has got a square root

$$a^{(p-1)/2} \equiv (p-1) (\mathrm{mod}\ p) \Rightarrow a$$ has not got a square root

How come ?

Element 1 is a result in powers of each element,
because: $1 \equiv 2^6 \equiv 3^6 \equiv 4^6 \equiv 5^6 \equiv 6^6 (\mathrm{mod}\ 7)$

Since exponent 6 is even, we may compute the square root:

$$\sqrt{1} \equiv \{1;-1\} \wedge -1 \in [p-1]_p$$

Computer Algebra 3

Computing square roots

May generating elements never have square roots ?

By definition, the even powers of generating elements do have square roots:

$$\sqrt{2} \equiv \{3;4\};$$
$$\sqrt{4} \equiv \{2;5\};$$
$$\sqrt{1} \equiv \{1;6\};$$

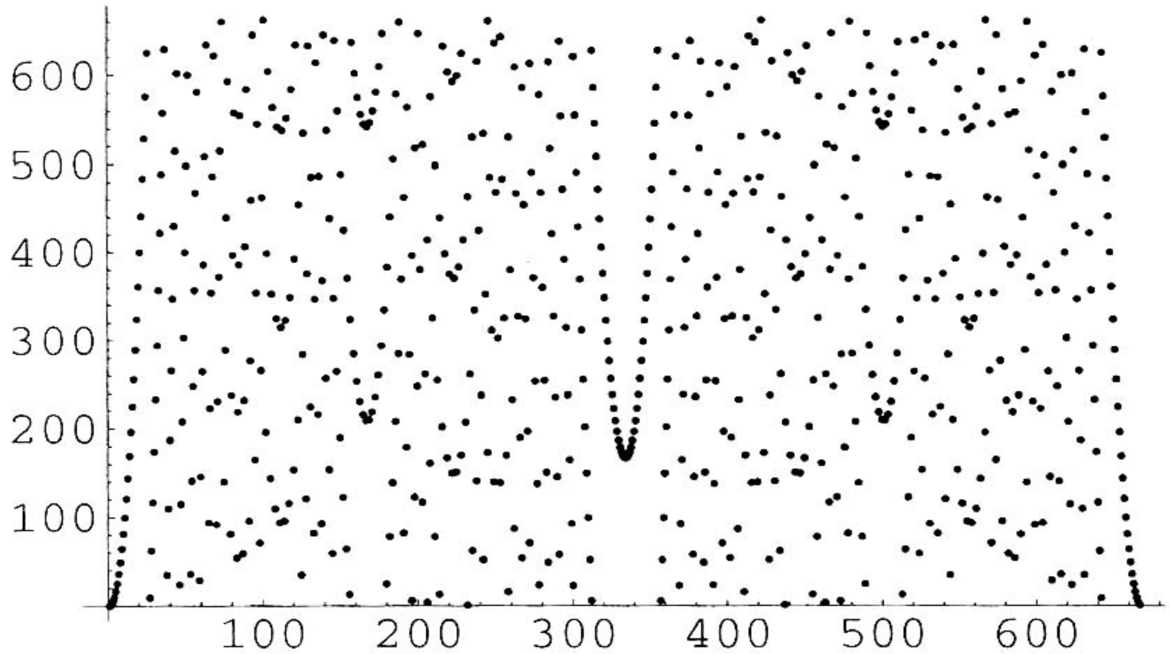
$$3^1 \equiv 3$$
$$3^2 \equiv 2$$
$$3^3 \equiv 6$$
$$3^4 \equiv 4$$
$$3^5 \equiv 5$$
$$3^6 \equiv 1$$

The powers of generating elements themselves as well as of the other generating elements must be odd if the modulus is a prime number.

Computer Algebra 3

Computing square roots

Squares in $\mathbb{Z}_{667}$ $f(x)=x^2$



For small numbers known parabolic form

Symmetry by

$$x^2 \equiv (x-p)^2 (\mathrm{mod}\ p)$$

There are also more solutions than 2,
if $\mathbb{Z}_n$ is composed (n=p*q)
667=23*29

$$\sqrt{506} = \{62;315;352;602\}$$

Summary for square roots

How to compute a square root:

Let a $\in \mathbb{Z}_n$ be known
Let x $\in \mathbb{Z}_n$ be searched

In a residue set with prime modulus n, there are efficient methods to compute square roots of the form $\sqrt{a} \equiv x \pmod{n}$

If n=p*q is sufficiently large (more than 200 digits), even modern computers need years. However, $x^2>n$ should hold, i.e. results between 0 and $\sqrt{n}$ have to be avoided. The reason is that for real numbers, numerical solutions are available.

But the inverse, the square is computable easily: $a^2 \equiv x \pmod{n}$

Computer Algebra 3

Discrete Logarithms

Logarithms in $\mathbb{Z}_7$

Query: Which is the element I have to exponentiate 2 with in order to get 4? $2^x \equiv 4 (\mathrm{mod}\ 7)$

Answer: $\log_2 4 \equiv 2 (\mathrm{mod}\ 7)$ because $2^2 \equiv 4 (\mathrm{mod}\ 7)$

and $\log_2 4 \equiv 5 (\mathrm{mod}\ 7)$ because $2^5 \equiv 32 \equiv 4 * 7 + 4 \equiv 4 (\mathrm{mod}\ 7)$

Query: $2^x \equiv 5 (\mathrm{mod}\ 7)$

Answer: $\log_2 5 \equiv ? (\mathrm{mod}\ 7)$ no solution

Computer Algebra 3

Discrete Logarithms

Logarithms in $\mathbb{Z}_7$

(mod 7) is not mentioned for the sake of readability

$1^1 \equiv 1$	$2^1 \equiv 2$	$3^1 \equiv 3$	$4^1 \equiv 4$	$5^1 \equiv 5$	$6^1 \equiv 6$
$1^2 \equiv 1$	$2^2 \equiv 4$	$3^2 \equiv 2$	$4^2 \equiv 2$	$5^2 \equiv 4$	$6^2 \equiv 1$
$1^3 \equiv 1$	$2^3 \equiv 1$	$3^3 \equiv 6$	$4^3 \equiv 1$	$5^3 \equiv 6$	$6^3 \equiv 6$
$1^4 \equiv 1$	$2^4 \equiv 2$	$3^4 \equiv 4$	$4^4 \equiv 4$	$5^4 \equiv 2$	$6^4 \equiv 1$
$1^5 \equiv 1$	$2^5 \equiv 4$	$3^5 \equiv 5$	$4^5 \equiv 2$	$5^5 \equiv 3$	$6^5 \equiv 6$
$1^6 \equiv 1$	$2^6 \equiv 1$	$3^6 \equiv 1$	$4^6 \equiv 1$	$5^6 \equiv 1$	$6^6 \equiv 1$

Generating elements have a unique solution.

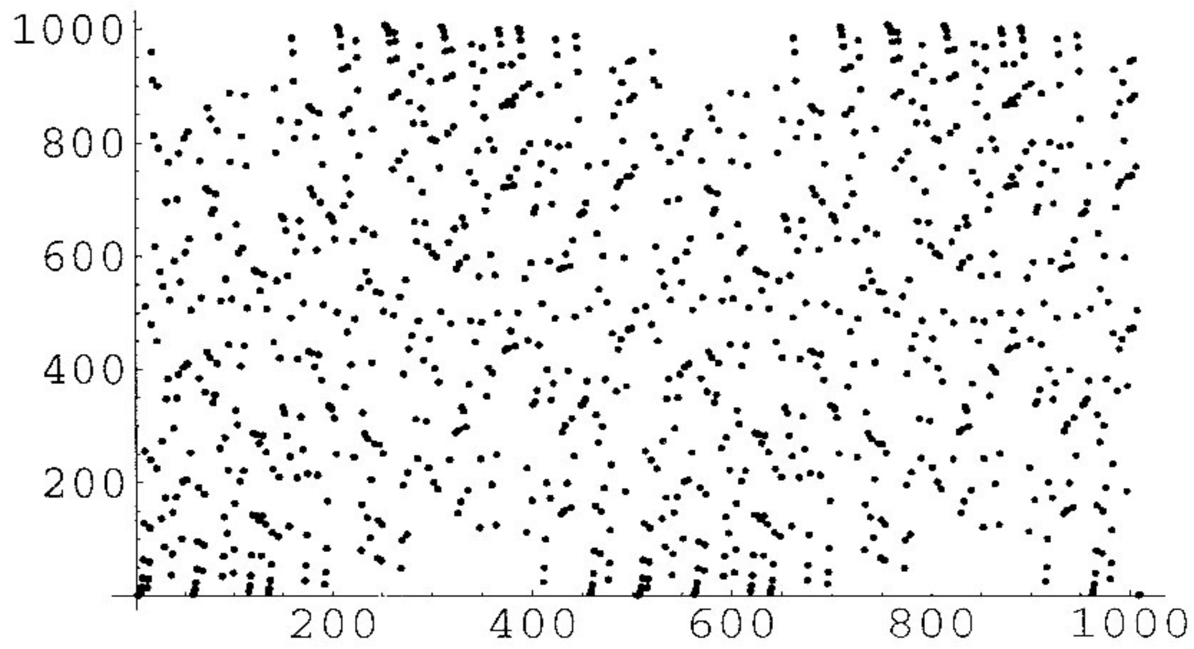
Not all elements of $\mathbb{Z}_7$ are reached.
Some elements are not unique.

$$\log_2 5 \equiv ?$$

Computer Algebra 3

Discrete Logarithms

Powers in $\mathbb{Z}_{1009}$ for base 2, $f(x)=2^x$



Initially the known exponential function

Graph is always periodic for a divisor of n-1.
Here: (n-1)=(1009-1)
1008/2=504

Computer Algebra 3

Summary for discrete Logarithms

How to compute discrete logarithms

Let a $\in \mathbb{Z}_n$ be known
Let x $\in \mathbb{Z}_n$ be searched

If n is suffieciently large (2007: at least 200 digits), $\log_b a \equiv x (\mathrm{mod}\ n)$
Can only be computed within years even on modern computers.
However, $b^x > n$ should hold. Otherwise a solution may be obtained using a numerical method.

On the other hand, the invers function
may be computed easily: $b^a \equiv x (\mathrm{mod}\ n)$

Computer Algebra 3

Summary: Computation of modular functions

- Find the multiplicative inverses b/a mod n (modular division):

  (Multiplication is efficiently solvable for each n.
  Algorithms for division:
  for prime moduli n: O(#n^2) using the extended Euclidean Algorithm
  for composed moduli n, gcd(a,n) = 1: see above
  für composed moduli n, gcd(a,n) > 1: no efficient algorithm known

- Find the square root mod n:

  Inverse of squaring which is efficiently solvable.
  Algorithm for square root:
  for prime moduli: there are polynomial methods (not trivial)
  for composed moduli n: no efficient algorithm known

- Find the logarithm mod n

  Inverse of exponentiation which is efficiently solvable.
  Algorithms for the logarithm:
  no efficient algorithm known for any n.