Multiplication of Polynomials in $\mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$ using the Fast Fourier Transform

Cory Ahn
Supervisor: Michael Monagan

Simon Fraser University

June 27, 2010

Motivation

Suppose that $f(x)$ and $g(x)$ are dense polynomials in $K[x]$ where $K = \mathbb{Q}(\alpha_1, \ldots, \alpha_t)$ is an algebraic number field and $\{\alpha_1, \ldots, \alpha_t\}$ is an algebraically independent set over $\mathbb{Q}$.

Question

How can we compute $h(x) = f(x) \cdot g(x)$ efficiently?

Representing $f(x) \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$

First, we must find a way to represent a polynomial $f \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$ in a computer.

Fact

$K(\alpha_1, \ldots, \alpha_t)[x] \cong K[x, u_1, \ldots, u_t]/\langle m_1, \ldots, m_t \rangle$, where $m_i := m_i(u_i)$ is the minimal polynomial for α_i over K.

Thus we can consider f as a $(t+1)$-variate polynomial in $\mathbb{Q}[x, u_1, \ldots, u_t]/\langle m_1, \ldots, m_t \rangle$.

Representing $f(x) \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$

We also need to choose a data structure to represent the polynomials. We will use a **recursive dense** data structure (`recden` in Maple).

- **recursive** $\Rightarrow$ nested list
- **dense** $\Rightarrow$ terms with zero coefficients are stored in the list

The `recden` data structure

Example

Let $f(x,y) = 13 + 8x^2y - 4\sqrt{2}y^2 \in \mathbb{Z}_7(\sqrt{2})[x,y]$ with $x >_{lex} y$.

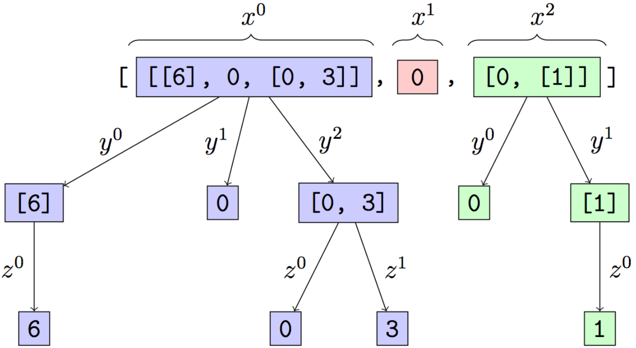
```
> f:= 13 + 8*x^2*y - 4*z*y^2 :
> F:= rpoly(f,[x,y,z],7, z=RootOf(a^2-2));
                   2              2        2
           F := (x  y + 6 + 3 z y ) mod <z  + 5, 7>
> lprint(F);
POLYNOMIAL([7, [x, y, z], [[5, 0, 1]]],[[[6], 0, [0, 3]], 0, [0, [1]]])
```

- `[7, [x, y, z], [[5, 0, 1]]]` $\Longleftrightarrow \mathbb{Z}_7[x,y]/\langle z^2 - 2\rangle$.
- `[[[6], 0, [0, 3]], 0, [0, [1]]]`: `recden` representation of $f(x,y)$.

The `recden` date structure

$$f(x,y) = 13 - 4y^2z + 8x^2y$$

$$\equiv \quad \boxed{(6y^0 + 0y^1 + (0z^0 + 3z^1)y^2)}\, x^0 + \boxed{0}\, x^1 + \boxed{(0y^0 + 1y^1)}\, x^2 \mod 7$$



Naïve Multiplication Strategy

1. convert $f(x), g(x) \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$ to `recden` polynomials $F(x), G(x) \in \mathbb{Q}[u_1, \ldots, u_t, x]/\langle m_1, \ldots, m_t \rangle$ where $m_i = m_i(u_i)$ is the minimal polynomial of α_i over $\mathbb{Q}$.
2. multiply $F(u_1, \ldots, u_t, x)$ and $G(u_1, \ldots, u_t, x)$ "naively", i.e., multiply each term in F by each term in G, etc.
   Note:
   $$F \cdot G = \left(\sum_{j=0}^{m} \sum_{i=1}^{t} \sum_{k=0}^{d_i - 1} \left(c_{i,k} u_i^k \right) x^j \right) \cdot \left(\sum_{j=0}^{n} \sum_{i=1}^{t} \sum_{k=0}^{d_i - 1} \left(\tilde{c}_{i,k} u_i^k \right) x^j \right),$$
   where $d_i = \deg(\alpha_i)$.

$\Rightarrow \mathcal{O}(mn \prod_{i=1}^{t} d_i)$ arithmetic operations. slow!

Naive Multiplication - Problem 1

Problem 1: more variables in polynomial = more "complicated" `recden` data structure

Example

```
> f:= a + b + c + d + e;
                 f := a + b + c + d + e

> rpoly(f,[a,b,c,d,e],7);
                 (a + b + c + d + e) mod 7

> lprint(%);
POLYNOMIAL([7, [a, b, c, d, e], []],
[[[[[0, 1], [1]], [[1]]], [[1]]]], [[[[1]]]]])
```

⇒ longer time to access all the elements in the list.

Solution to Problem 1

Solution: multiple extensions $\longrightarrow$ single extension. How?

Theorem

Let K be a subfield of $\mathbb{C}$ and $\alpha, \beta \in \mathbb{C}$ be algebraic over K. Then there exists $\gamma \in \mathbb{C}$ that is algebraic over K such that $K(\alpha, \beta) = K(\gamma)$.

How to find γ:
Let $\alpha_2, \ldots, \alpha_m$ be the conjugates of $\alpha(= \alpha_1)$ and let $\beta_2, \ldots, \beta_n$ be the conjugates of $\beta(= \beta_1)$. Define the set

$$S = \left\{ \frac{\alpha_r - \alpha_s}{\beta_t - \beta_u} : r, s \in \{1, \ldots, m\}, t, u \in \{1, \ldots, n\}, t \neq u \right\}.$$

Now let $c \in K \backslash S$.
Proof of above theorem tells us that $K(\alpha, \beta) = K(\gamma := \alpha + c\beta)$.
This is a bit of work... We will randomly choose c instead (more on this later).

Solution to Problem 1 (ctd.)

Corollary

Let K be a subfield of $\mathbb{C}$ and let $\alpha_1, \ldots, \alpha_n$ be algebraic over K. Then there exists $\alpha \in \mathbb{C}$, algebraic over K, such that $K(\alpha_1, \ldots, \alpha_n) = K(\alpha)$.

Proof.

If $n = 1$ then let $\alpha = \alpha_1$. So suppose that $n \geq 2$. We repeatedly apply previous theorem:

$$\begin{aligned}
K(\alpha_1, \alpha_2, \ldots, \alpha_n) &= K(\alpha_1, \alpha_2)(\alpha_3, \ldots, \alpha_n) \\
&= K(\beta_2, \alpha_3, \ldots, \alpha_n) \text{ where } K(\beta_2) = K(\alpha_1, \alpha_2), \\
&= K(\beta_2, \alpha_3)(\alpha_4, \ldots, \alpha_n) \\
&= K(\beta_3, \alpha_4, \ldots, \alpha_n) \text{ where } K(\beta_3) = K(\beta_2, \alpha_3) \\
&= \cdots \\
&= K(\beta_n) = K(\alpha).
\end{aligned}$$

$\square$

Solution to Problem 1 (ctd.)

So the previous theorem and corollary tells us that we can always find γ such that

$$\mathbb{Q}(\alpha_1, \dots, \alpha_t)[x] = \mathbb{Q}(\gamma)[x].$$

i.e.,

$$\mathbb{Q}[u_1, \dots, u_t][x]/\langle m_{\alpha_1}(u_1), \dots, m_{\alpha_t}(u_t)\rangle = \mathbb{Q}[z][x]/\langle m_\gamma(z)\rangle.$$

Once γ is found, we can express f as a bivariate polynomial in $\mathbb{Q}[x, z]/m_\gamma(z)$.
$\Rightarrow$ simpler `recden` data structure!
In fact, it is a list of $n := deg_x(f)$ lists of length at most $d := \deg(\gamma)$:

$$\Big[\overbrace{\underbrace{[\,\square, \square, \cdots, \square\,]}_{\leq d \text{ elements}}, \underbrace{[\,\square, \square, \cdots, \square\,]}_{\leq d \text{ elements}}, \cdots, \underbrace{[\,\square, \square, \cdots, \square\,]}_{\leq d \text{ elements}}}^{n \text{ lists}}\Big]$$

Naive Multiplication Problem 2

Problem 2: "Naive" multiplication is slow : $\mathcal{O}(n^2 d^2)$
Solution: Use the fast Fourier Transform (FFT): $\mathcal{O}(nd^2 + dn \log_2 n)$

Solution to Problem 2: Multiplication using FFT

Suppose we wish to multiply $f(x,z), g(x,z) \in \mathbb{Z}_p[z][x]/\langle m_\gamma(z)\rangle$.

- $N \leftarrow$ smallest power of 2 greater than $\deg_x(f) + \deg_x(g)$.
- $\omega \leftarrow$ primitive N^{th} root of unity

Multiplication using the Fast Fourier Transform (FFT) works as follows.

$$
\begin{aligned}
A &\leftarrow \left[f(1,z), f(\omega,z), \ldots, f(\omega^{N-1},z)\right] \\
B &\leftarrow \left[g(1,z), g(\omega,z), \ldots, g(\omega^{N-1},z)\right] \\
C &\leftarrow [A[1]\cdot B[1],\ A[2]\cdot B[2], \ldots, A[N-1]\cdot B[N-1]] \\
h &\leftarrow C[1] + C[2]x + \cdots + C[N-1]x^{N-1} \in \mathbb{Z}_p[x,z]/\langle m_\gamma(z)\rangle \\
H &\leftarrow N^{-1}\cdot\left[h(1,z), h(\omega^{-1},z), \ldots, h(\omega^{-(N-1)},z)\right]
\end{aligned}
$$

return $C[1] + C[2]x + \cdots C[N-1]x^{N-1}\,(= f(x,z)\cdot g(x,z))$

Naive Multiplication Problem 3

Problem 3: coefficients of the polynomials belong to $\mathbb{Q}$
$\Rightarrow$ rapid growth of numerators and denominators

Example

Let

$$f(x) = \frac{83375}{3698}\sqrt{2} + \frac{58523}{37544}x \in \mathbb{Q}(\sqrt{2})[x]$$

and

$$g(x) = \frac{9085}{702} + \frac{75149}{20728}\sqrt{2}x \in \mathbb{Q}(\sqrt{2})[x].$$

Then

$$f(x) \cdot g(x) = \frac{757461875}{2595996}\sqrt{2} + \frac{6265547875}{38326072}x + \frac{531681455}{26355888}x + \frac{4397944927}{778212032}\sqrt{2}x^2$$

$$= \frac{757461875}{2595996}\sqrt{2} + \frac{23188917472191595}{126264707638992}x + \frac{4397944927}{778212032}\sqrt{2}x^2.$$

Solution: map $\mathbb{Q}$ to $\mathbb{Z}_p$ where p is a "suitable" prime (and map back to $\mathbb{Q}$ after multiplying).

A Better Multiplication Strategy

Hence our strategy for finding the product of f and g will be:

1. Find $f_p, g_p \in \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x]$ from $f, g \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$.
2. Convert $f_p, g_p \in \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x]$ to $f_\gamma, g_\gamma \in \mathbb{Z}_p(\gamma)[x] = \mathbb{Z}_p[x, z]/\langle m_\gamma(z) \rangle$.
3. Find $h_\gamma := f_\gamma \cdot g_\gamma \in \mathbb{Z}_p[z, x]/\langle m_\gamma(z) \rangle$ using FFT.
4. Convert h_γ to a polynomial in $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x]$.
5. Use rational number reconstruction on h_γ to find $h := fg \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x]$.

Multiplication Step 2: $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \longrightarrow \mathbb{Z}_p(\gamma)[x]$

Recall: we need to find $c_2, \ldots, c_t \in \mathbb{Z}_p$ such that $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] = \mathbb{Z}_p(\gamma = \alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t)[x]$.

Fact

Let $\deg(\alpha_i) = d_i$ *for* $i = 1, \ldots t$ *and* $\deg(\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)) = \prod_{\mathrm{I}=1}^{t} d_i := d$.
Suppose that we randomly choose a set of numbers $\chi := \{c_2, \ldots, c_t\}$, *where each* $c_i \in \mathbb{Z}_p$.
The probability of choosing the "wrong" χ *such that* $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \neq \mathbb{Z}_p(\gamma = \alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t)[x]$ *is approx.* $\frac{d^2}{p}$.

Our prime p is large (more on this later), so $\frac{d^2}{p}$ will be small.
In light of the above fact, we will pick the the numbers $c_2, \ldots, c_t$ at random.

Multiplication Step 2: $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \longrightarrow \mathbb{Z}_p(\gamma)[x]$

Lemma

Let $\deg(\mathbb{Z}_p(\alpha_i)) = d_i$ *and* $\deg(\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)) = \prod_{i=1}^{t} d_i = d$*. Then* $B_1 := \{1, \gamma, \gamma^2, \ldots, \gamma^{d-1}\}$ *and* $B_2 := \{\alpha_1^{j_1} \alpha_2^{j_2} \cdots \alpha_t^{j_t}, j_i = 0, 1, \ldots, d_i - 1\}$ *are bases for* $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t) = \mathbb{Z}_p(\gamma)$*.*

We are given f_p and g_p whose coefficients are expressed in terms of the elements in B_2.
How to change these to be expressed in terms of the elements in B_1?
Answer: use a change of basis matrix.

Multiplication Step 2: $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \longrightarrow \mathbb{Z}_p(\gamma)[x]$

We would like to build a $d \times d$ change-of-basis matrix
$C = \left[\gamma^0,\ \gamma^1,\ \gamma^2,\ \ldots,\ \gamma^{d-1}\right]$, where each column
$\gamma^i = (\alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t)^i$ is expressed as a linear combination of elements in $B_2 = \{\alpha_1^{j_1}\,\alpha_2^{j_2}\cdots\alpha_t^{j_t}, j_i = 0, 1, \ldots, d_i - 1\}$
[So that C^{-1} is a change-of-basis matrix from B_2 to B_1].
But in `recden` data structure, the γ^i's may not all be of length d...

Example

Let our field be $\mathbb{Z}_7(\alpha_1, \alpha_2)$ with $\alpha_1 = \sqrt{2}$ and $\alpha_2 = \sqrt{5}$. Then
$\gamma = \sqrt{2} + \sqrt{5}$. In `recden` with $\alpha_1 > \alpha_2$,
$\gamma^0 = [[1]]$
$\gamma^1 = \alpha_1 + \alpha_2 = [[0, 1], [1]]$
$\gamma^2 = (\alpha_1 + \alpha_2)^2 = [0, [0, 2]]$
$\gamma^3 = (\alpha_1 + \alpha_2)^3 = [[0, 4], [3]]$

So we need a new data structure.

Multiplication Step 2 - : $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \longrightarrow \mathbb{Z}_p(\gamma)[x]$

Definition

A **completely dense representation (cdr)** of f in $K[x_1, \ldots, x_n]$ is a list of coefficients of f written in increasing order of lexicographical ordering on $x_1, \cdots, x_n$. This data structure stores *every* coefficient of f up to $x_1^{d_1} \cdots x_n^{d_n}$, where d_i is the largest degree of x_i in f.

Example

Let $f(x, y) = 8x^2y - 4y + 13 \in \mathbb{Z}_7[x, y]$. The **cdr** stores the coefficients of f in the following order (with $y \prec_{lex} x$):

$$1,\ y,\ x,\ xy,\ x^2,\ x^2y.$$

So the **cdr** of f is: `[ 6, 3, 0, 0, 0, 1 ]`.

Let us extend the idea of **cdr** to fields with extensions.

Definition

A **completely dense representation (cdr)** of f in $K[u_1, \ldots, u_t]/\langle m_1, \ldots, m_t \rangle$ is a list of coefficients of f written in increasing order of lexicographical ordering on $u_1, \cdots, u_t$ that stores *every* coefficient of f up to $u_1^{d_1-1} \cdots u_t^{d_t-1}$, where $d_i = \deg(m_i)$.

Example

Let $f(x,y) = x^2y^2 + x + y + 3 \in \mathbb{Z}_7[x,y]/\langle x^2 - 3, y^3 - 2 \rangle$. Then $f(x,y) \equiv x + 3y^2 + y + 3$. *Every* polynomial in this polynomial ring can be written as $c_0 + c_1 y + c_2 y^2 + c_3 x + c_4 xy + c_5 xy^2$ where each $c_i \in \mathbb{Z}_7$. That is, every polynomial in **cdr** (with $y \prec_{lex} x$) in this polynomial ring is:

$$[c_0, c_1, c_2, c_3, c_4, c_5].$$

So $f(x,y) = [3, 1, 3, 1, 0, 0]$.

Multiplication Step 2: $\mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \longrightarrow \mathbb{Z}_p(\gamma)[x]$

Using the **cdr** data structure, each $\gamma^i = (\alpha_1 + c_2\alpha_2 + \cdots + c_t\alpha_t)^i$ will be of length d ($= \deg(\gamma)$).
So we can build a $d \times d$ change-of-basis matrix C
(note: this matrix C will be used for going from B_1 to B_1 in Step 4, namely $h_\gamma = f_\gamma g_\gamma \in \mathbb{Z}_p(\gamma)[x] \longrightarrow h_p \in \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x]$).

We require $C^{-1} \mod p$ to go from B_2 to B_1.

Problem: What if C is not invertible in $\mathbb{Z}_p$?

Choosing the "Right" Prime

There are two restrictions on the prime p:

1. p must be a Fourier prime (i.e. a prime of form $k \cdot 2^r + 1$, k odd and $r \geq R$, where 2^R is the smallest power of two greater than $\deg_x(f) + \deg_x(g)$).
2. C must be invertible in $\mathbb{Z}_p$.

We will further restrict p to be between 2^{30} and $2^{31.5}$, so that all numbers arising from our algorithm can be stored in a 64-bit machine without overflow.

Choosing the “Right” Prime

We will choose a Fourier prime as follows.

Step 1. randomly choose a prime $p \in (2^{30}, 2^{31.5})$.

Step 2. check if remainder upon dividing $p-1$ by N is zero. If so, this p is a Fourier prime. If not, go back to Step 1.

Fact

Out of all Fourier primes between 2^{30} and $2^{31.5}$ for a given $N = 2^R$ and $d = \deg(\gamma)$, the probability that a random Fourier prime divides $\det(C)$ is at most

$$\max\left\{\frac{d/2 + Rd}{8.7458 \times 10^7}, \frac{(d/2+Rd)\cdot 2^R}{9.8163 \times 10^8}\right\}.$$

Choosing the "Right" Prime

Example

Suppose we wish to multiply two polynomials $f(x)$ and $g(x) \in \mathbb{Z}_p(\alpha_1, \ldots, \alpha_4)[x]$ with $\deg_x(f) + \deg_x(g) = 500$. If $\deg(\alpha_i) = 5$ for each $i = 1, \ldots, 4$ then $d = 4^5 = 1024$. Also, $N = 2^9$. So we expect, on average, to find an "unfortunate" prime p with probability of at most

$$\max\left\{\frac{1024/2 + 9 \cdot 1024}{8.7458 \times 10^7}, \frac{(9 \cdot 1024 + 1024/2) \cdot 2^9}{9.8163 \times 10^8}\right\} = 0.0050739$$
$$\approx 5/1000.$$

i.e., if we pick a random Fourier prime between 2^{30} and $2^{31.5}$ with $N = 500$ and $d = 1024$, we expect to choose an "unfortunate" prime with the probability of approximately 5 in 1000.

Summary of the Multiplication Procedure

$$
\begin{tikzcd}
f, g \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x] \arrow[r, "\text{mod } p"] \arrow[ddr, bend right, "\textit{multiply}"'] & f_p, g_p \in \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \arrow[r, "\textit{change of}", "\textit{basis}"'] \arrow[d, "\textit{multiply}"] & f_\gamma, g_\gamma \in \mathbb{Z}_p(\gamma)[x] \arrow[d, "\textit{multiply}"] \\
& f_p g_p \in \mathbb{Z}_p(\alpha_1, \ldots, \alpha_t)[x] \arrow[d, "\textit{rational reconstruction}"] & f_\gamma g_\gamma \in \mathbb{Z}_p(\gamma)[x] \arrow[l, "\textit{change of}"', "\textit{basis}"] \\
& fg \in \mathbb{Q}(\alpha_1, \ldots, \alpha_t)[x] &
\end{tikzcd}
$$

Overall cost of the improved multiplication algorithm is $\mathcal{O}(d^3 + nd^2 + dn \log_2 n)$.

Benchmarks

All computations were performed on a Mac OS X with 2.4 GHz Intel Core 2 Duo and 2 GB of 1.07 GHz RAM. For all cases we used $p = 3037000453$, the largest prime for which arithmetic is done in the 64-bit machine.

	$\mathbb{Z}_p(\sqrt{111}, \sqrt{131})[x]$			$\mathbb{Z}_p(\sqrt{111} + \sqrt{131})[x]$		
d	`mulrpoly`	**FFTMult**	conversion 1	`mulrpoly`	**FFTMult**	conversion 2
12	0.024	0.025	0.005	0.001	0.009	0.002
24	0.076	0.078	0.005	0.004	0.021	0.003
48	0.420	0.115	0.009	0.014	0.049	0.005
96	1.380	0.289	0.015	0.060	0.108	0.009
192	5.022	0.651	0.032	0.230	0.244	0.017
384	20.022	1.420	0.077	0.904	0.554	0.035

	$\mathbb{Z}_p(\sqrt{111}, \sqrt{131}, \sqrt{171})[x]$			$\mathbb{Z}_p(\sqrt{111} + \sqrt{131} + \sqrt{171})[x]$		
d	`mulrpoly`	**FFTMult**	conversion 1	`mulrpoly`	**FFTMult**	conversion 2
12	0.148	0.078	0.007	0.002	0.011	0.002
24	0.542	0.256	0.009	0.004	0.024	0.005
48	1.912	0.372	0.018	0.017	0.052	0.010
96	7.832	0.766	0.037	0.065	0.117	0.020
192	30.632	1.707	0.091	0.286	0.266	0.040
384	124.354	3.748	0.247	1.132	0.607	0.079

Benchmarks

- α_1 is a root of $z^4 - 94 - 7z^3 + 22z^2 - 55z$,
- α_2 is a root of $z^4 - 62 + 87\alpha_1^3 - 56\alpha_1^2 + (-83 - +97\alpha_1^3 - 73\alpha_1^2 - 4\alpha_1)z + (80 - 10\alpha_1^3 + 62\alpha_1^2 - 81\alpha_1)z^2 + (-75 - 44\alpha_1^3 + 71\alpha_1^2 - 17\alpha_1)z^3$, and
- α_3 is a root of $z^4 + 42 - 10\alpha_2^3 - 7\alpha_2^2 - 40\alpha_2 + (-92 - 50\alpha_2^3 + 23\alpha_2^2 + 75\alpha_2)z + (37 + 6\alpha_2^3 + 74\alpha_2^2 + 72\alpha_2)z^2 + (29 - 23\alpha_2^3 + 87\alpha_2^2 + 44\alpha_2)z^3$.

	$\mathbb{Z}_p(\alpha_1, \alpha_2, \alpha_3)[x]$			$\mathbb{Z}_p(\gamma)[x]$		
d	mulrpoly	**FFTMult**	conversion 1	mulrpoly	**FFTMult**	conversion 2
12	1.329	0.522	0.210	0.016	0.023	0.051
24	4.177	1.100	0.183	0.055	0.052	0.151
48	15.567	2.280	0.535	0.189	0.118	0.249
96	59.968	4.792	1.209	0.736	0.271	0.448
192	237.216	10.289	3.363	2.857	0.613	0.825