

Galois Groups

May 4, 2022

Contents

| | | |
|----------|-----------------------------------|----------|
| 1 | Question 1 | 3 |
| 2 | Question 3 | 3 |
| 3 | Question 4 | 4 |
| 3.1 | $x^2 + x + 41$ | 4 |
| 3.2 | $x^3 + 2x + 1$ | 4 |
| 3.3 | $x^3 + x^2 - 2x - 1$ | 4 |
| 3.4 | $x^4 - 2x^2 + 4$ | 4 |
| 3.5 | $x^4 - x^3 - 4x + 16$ | 4 |
| 3.6 | $x^4 - 2x^3 + 5x + 5$ | 4 |
| 3.7 | $x^4 + 7x^2 + 6x + 7$ | 5 |
| 3.8 | $x^4 + 3x^3 - 6x^2 - 9x + 7$ | 5 |
| 3.9 | $x^5 + 36$ | 5 |
| 3.10 | $x^5 - 5x + 3$ | 5 |
| 3.11 | $x^5 + x^3 - 3x^2 + 3$ | 5 |
| 3.12 | $x^5 - 11x^3 + 22x - 11$ | 5 |
| 3.13 | $x^6 + x + 1$ | 5 |
| 3.14 | $x^7 - 2x^6 + 2x + 2$ | 5 |
| 3.15 | $x^7 + x^4 - 2x^2 + 8x + 4$ | 5 |
| 3.16 | $x^7 + x^5 - 4x^4 - x^3 + 5x + 1$ | 5 |
| 3.17 | Frequency of Cycle Types | 6 |
| 4 | Code | 6 |
| 4.1 | Question 1 | 6 |
| 4.2 | Question 2 | 9 |

Preface

This is my CATAM project, 16.1 for part II. The code for each question can be found in section 4.

1 Question 1

See the table below for some outputs for my programs which calculates the quotient q remainder r when dividing polynomials f by g over \mathbb{F}_p .

| f | g | q | r | p |
|--------------------------|-----------------------|------------------------|------------------------|----|
| $x^5 - 11x^3 + 22x - 11$ | $x^4 - x^3 - 4x + 16$ | $x + 1$ | $4x^3 + 4x^2 + 3x + 1$ | 7 |
| $x^5 - 11x^3 + 22x - 12$ | $x^3 + x^2 - 2x - 1$ | $x^2 + 4x + 2$ | $2x^2 + 1$ | 5 |
| $x^4 - x^3 - 4x + 16$ | $x + 1$ | $x^3 + 11x^2 + 2x + 7$ | 9 | 13 |
| $x^3 + 3x + 1$ | $x^2 + 3$ | x | 1 | 2 |

Table 1: Various outputs of polynomial division program

See the table below for some outputs for my programs which calculates the GCD of two polynomials f, g over \mathbb{F}_p .

| f | g | GCD | p |
|-----------------------|------------------------|----------------|----|
| $x^2 + x + 1$ | $x + 1$ | 1 | 3 |
| $x^2 + 2x + 1$ | $x^3 + 3x^2 + 3x + 1$ | $x^2 + 2x + 1$ | 17 |
| $x^6 + 5x^3 + 6x + 1$ | $x^5 + 3x^3 + 12x + 7$ | $3x + 4$ | 11 |
| $x^6 + 1$ | $x^3 + 1$ | 2 | 23 |

Table 2: Various outputs of polynomial GCD program

A way to efficiently calculate the power, say n , of a polynomial f modulo a polynomial g using my programs written in this question is as follows:

1. Firstly write n in base 2, say $n = b_n \dots b_0 = \sum_{i=0}^n b_i 2^i$.
2. Iteratively calculate $f^{2^i} \pmod{g}$ via $f^{2^i} = (f^{2^{i-1}})^2$. Then use the polynomial division algorithm to reduce it modulo g .
3. Iteratively calculate $\prod_{i=0}^k f^{b_i 2^i} \pmod{g}$ by consecutively multiplying it by $f^{2^{k+1}}$ if $b_{k+1} = 1$ and 1 otherwise. At each stage use the polynomial division program to reduce the expression modulo g .

2 Question 3

Find below the tables of the decomposition group of each polynomial for each prime between 2 and 97.

| Polynomial | Prime | | | | | | | | | | | | |
|-----------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 | 37 | 41 |
| $x^2 + x + 41$ | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_1 |
| $x^3 + 2x + 1$ | C_2 | C_3 | C_3 | C_3 | C_2 | C_2 | C_1 | C_3 | C_2 | C_3 | C_2 | C_2 | C_3 |
| $x^3 + x^2 - 2x - 1$ | C_3 | C_3 | C_3 | - | C_3 | C_1 | C_3 | C_3 | C_3 | C_1 | C_3 | C_3 | C_1 |
| $x^4 - 2x^2 + 4$ | - | - | C_2 | C_2 | C_2 | C_2 | C_2 | C_1 | C_2 | C_2 | C_2 | C_2 | C_2 |
| $x^4 - x^3 - 4x + 16$ | - | - | C_4 | C_4 | - | C_2 | C_2 | C_4 | C_2 | C_2 | C_2 | C_2 | C_2 |
| $x^4 - 2x^3 + 5x + 5$ | C_4 | - | - | C_4 | C_4 | C_3 | C_3 | C_3 | C_4 | C_3 | C_3 | C_4 | C_3 |
| $x^4 + 7x^2 + 6x + 7$ | - | - | C_2 | C_1 | C_2 | - | C_2 | C_1 | C_2 | C_2 | C_1 | C_1 | C_2 |
| $x^4 + 3x^3 - 6x^2 - 9x + 7$ | - | C_2 | - | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_1 | C_2 | - |
| $x^5 + 36$ | - | - | - | C_4 | C_5 | C_4 | C_4 | C_2 | C_4 | C_2 | C_1 | C_4 | C_5 |
| $x^5 - 5x + 3$ | C_6 | C_2 | - | - | C_3 | C_2 | C_2 | C_2 | C_6 | C_3 | C_2 | C_6 | C_2 |
| $x^5 + x^3 - 3x^2 + 3$ | - | - | C_5 | C_3 | C_3 | C_5 | C_5 | C_5 | C_5 | C_2 | C_2 | C_5 | - |
| $x^5 - 11x^3 + 22x - 11$ | C_5 | C_5 | C_5 | C_5 | - | C_5 | C_5 | C_5 | C_1 | C_5 | C_5 | C_5 | C_5 |
| $x^6 + x + 1$ | C_6 | C_6 | C_3 | C_5 | C_6 | C_6 | C_6 | C_4 | C_3 | C_6 | C_6 | C_6 | C_4 |
| $x^7 - 2x^6 + 2x + 2$ | - | - | C_7 | C_7 | - | C_6 | C_4 | C_4 | C_7 | C_7 | C_7 | C_7 | C_5 |
| $x^7 + x^4 - 2x^2 + 8x + 4$ | - | - | C_6 | C_6 | C_2 | C_2 | C_2 | C_3 | C_2 | C_6 | C_2 | C_2 | C_6 |
| $x^7 + x^5 - 4x^4 - x^3 + 5x + 1$ | C_7 | - | C_2 | C_7 | C_2 | C_2 | C_7 | C_7 | C_2 | C_7 | C_7 | C_7 | C_7 |

Table 3: Decomposition groups of various polynomials for primes between 2 and 41

| Polynomial | Prime | | | | | | | | | | | |
|-----------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 43 | 47 | 53 | 59 | 61 | 67 | 71 | 73 | 79 | 83 | 89 | 97 |
| $x^2 + x + 41$ | C_1 | C_1 | C_1 | C_2 | C_1 | C_2 | C_1 | C_2 | C_2 | C_1 | C_2 | C_1 |
| $x^3 + 2x + 1$ | C_2 | C_2 | C_3 | - | C_2 | C_2 | C_1 | C_2 | C_3 | C_2 | C_2 | C_2 |
| $x^3 + x^2 - 2x - 1$ | C_1 | C_3 | C_3 | C_3 | C_3 | C_3 | C_1 | C_3 | C_3 | C_1 | C_3 | C_1 |
| $x^4 - 2x^2 + 4$ | C_1 | C_2 | C_2 | C_2 | C_2 | C_1 | C_2 | C_1 | C_2 | C_2 | C_2 | C_1 |
| $x^4 - x^3 - 4x + 16$ | C_4 | C_2 | C_4 | C_2 | C_2 | C_2 | C_2 | C_2 | C_4 | C_2 | C_4 | C_1 |
| $x^4 - 2x^3 + 5x + 5$ | C_2 | C_3 | C_2 | C_3 | C_4 | C_3 | C_3 | C_3 | - | C_4 | C_4 | C_2 |
| $x^4 + 7x^2 + 6x + 7$ | C_1 | C_2 | C_2 | C_2 | C_1 | C_1 | C_2 | C_1 | C_1 | C_2 | C_2 | C_1 |
| $x^4 + 3x^3 - 6x^2 - 9x + 7$ | C_2 | C_2 | C_2 | C_2 | C_2 | C_2 | C_1 | C_2 | C_1 | C_2 | C_1 | C_2 |
| $x^5 + 36$ | C_4 | C_4 | C_4 | C_2 | C_5 | C_4 | C_5 | C_4 | C_2 | C_4 | C_2 | C_4 |
| $x^5 - 5x + 3$ | C_2 | C_2 | C_2 | C_2 | C_6 | C_3 | C_2 | C_3 | C_2 | C_2 | C_4 | C_5 |
| $x^5 + x^3 - 3x^2 + 3$ | C_3 | C_3 | C_5 | C_3 | C_5 | C_3 | C_2 | C_2 | C_3 | C_3 | C_3 | C_3 |
| $x^5 - 11x^3 + 22x - 11$ | - | C_5 | C_5 | C_5 | C_5 | C_1 | C_5 | C_5 | C_5 | C_5 | C_1 | C_5 |
| $x^6 + x + 1$ | C_6 | C_6 | C_4 | C_4 | C_6 | C_5 | C_4 | C_4 | C_6 | C_5 | C_4 | C_3 |
| $x^7 - 2x^6 + 2x + 2$ | C_5 | C_7 | C_4 | C_7 | C_4 | C_5 | C_7 | C_3 | C_4 | C_7 | C_5 | C_4 |
| $x^7 + x^4 - 2x^2 + 8x + 4$ | C_2 | C_2 | C_6 | - | C_2 | C_2 | C_2 | C_2 | C_6 | C_2 | C_2 | C_2 |
| $x^7 + x^5 - 4x^4 - x^3 + 5x + 1$ | C_7 | C_7 | C_7 | C_2 | C_2 | C_2 | C_2 | C_7 | C_2 | C_7 | C_7 | C_2 |

Table 4: Decomposition groups of various polynomials for primes between 43 and 97

3 Question 4

3.1 $x^2 + x + 41$

Modulo 2 this polynomial has Galois Group C_2 . Hence over \mathbb{Q} it also has Galois group C_2 since this is the biggest group a degree 2 polynomial can have.

3.2 $x^3 + 2x + 1$

We can see over \mathbb{Q} this polynomial has a Galois group which contains a 2 and 3. Hence it contains a 2, $\deg f - 1 = 2$ and $\deg f$ cycle and so has S_3 as it's Galois group.

3.3 $x^3 + x^2 - 2x - 1$

We can see over \mathbb{Q} that this polynomial must contain a 3 cycle It maximally can have size 6 so either the group is D_6 or C_6 or C_3 .

3.4 $x^4 - 2x^2 + 4$

We know that it's Galois group over \mathbb{Q} contains a double transposition Hence it's Galois group over \mathbb{Q} must be a subgroup of S_4 which contains C_2 as a subgroup. Further we know this inclusion is strict since if it's $\text{Gal}(f)$ over \mathbb{Q} was C_2 then it would be a reducible polynomial as all degree 2 extensions are quadratic. Then since $x^4 - 2x^2 + 4$ is irreducible, the degree of the extension can't be two.

3.5 $x^4 - x^3 - 4x + 16$

We know that it's Galois group over \mathbb{Q} must be a subgroup of S_4 which contains a 4-cycle, a double transposition and single transposition.

3.6 $x^4 - 2x^3 + 5x + 5$

We can see that $\text{Gal}(f)$ over \mathbb{Q} must contain C_3, C_4 as a subgroup, further we know it must contain a 2-cycle since it's Galois group over \mathbb{F}_{43} is generated by such a cycle. Hence $\text{Gal}(f)$ over \mathbb{Q} contains a $\deg f, \deg f - 1$ cycle and a transposition and so must S_4 .

3.7 $x^4 + 7x^2 + 6x + 7$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^4 + 7x^2 + 6x + 7 = (x^2 - x + 7)(x^2 + x + 1).$$

Further note over \mathbb{F}_{17} it's Galois group is generated by a double transposition and so it's Galois group over \mathbb{Q} must contain a double transposition. But it's the orbits of the action of $\text{Gal}(f)$ on it's roots correspond to it's irreducible factors so we must in fact have that it's Galois group is $C_2 \times C_2$.

3.8 $x^4 + 3x^3 - 6x^2 - 9x + 7$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^4 + 3x^3 - 6x^2 - 9x + 7 = (x^2 + x - 1)(x^2 + 2x - 7).$$

Similarly to the previous polynomial we have the Galois group of this polynomial over \mathbb{Q} must be $C_2 \times C_2$.

3.9 $x^5 + 36$

We can see that over \mathbb{Q} , the Galois group of this polynomial contains a double transposition, a 4 cycle and a 5 cycle. Hence the Galois group of the polynomial must be a subgroup of S_5 which contains C_5, C_4 as subgroup.

3.10 $x^5 - 5x + 3$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^5 - 5x + 3 = (x^2 + x - 1)(x^3 - x^2 + 2x - 3).$$

Then it's Galois group over \mathbb{Q} contains C_6 as a subgroup. Hence $\text{Gal}(f)$ over \mathbb{Q} is either $C_6, C_2 \times D_{12}$.

3.11 $x^5 + x^3 - 3x^2 + 3$

We can see that over \mathbb{Q} , the Galois group of this polynomial contains C_2, C_3, C_5 as subgroups. Hence the Galois group of this polynomial over \mathbb{Q} is a subgroup of S_5 containing a 3,5 cycle and a double transposition. Since the finite fields over which this polynomial has C_2 as it's Galois group is generated by a double transposition we can't be sure if $\text{Gal}(f)$ is S_5 , which would be the case if $\text{Gal}(f)$ over \mathbb{Q} contains a single transposition.

3.12 $x^5 - 11x^3 + 22x - 11$

We can see that over \mathbb{Q} $\text{Gal}(f)$ contains a C_5 a subgroup. Hence $\text{Gal}(f)$ over \mathbb{Q} is a subgroup of S_5 containing C_5

3.13 $x^6 + x + 1$

We can see that $\text{Gal}(f)$ over \mathbb{Q} is a subgroup of S_6 containing C_4, C_5, C_6 as subgroups.

3.14 $x^7 - 2x^6 + 2x + 2$

We can see that $\text{Gal}(f)$ over \mathbb{Q} is a subgroup of S_6 containing C_4, C_5, C_6, C_7 as subgroups.

3.15 $x^7 + x^4 - 2x^2 + 8x + 4$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^7 + x^4 - 2x^2 + 8x + 4 = (x^3 + 2x + 1)(x^4 - 2x^2 + 4).$$

Then it's Galois group over \mathbb{Q} contains C_6 as a subgroup. Note as previously discussed $\text{Gal}_{\mathbb{Q}}(x^3 + 2x + 1) = D_6$ and $C_2 < \text{Gal}_{\mathbb{Q}}(x^4 - 2x^2 + 4)$. Then certainly $\text{Gal}(f)$ over \mathbb{Q} must contain a copy of D_6 and C_6 .

3.16 $x^7 + x^5 - 4x^4 - x^3 + 5x + 1$

We can see that $\text{Gal}(f)$ over \mathbb{Q} must contain C_2 and C_7 as a subgroup.

3.17 Frequency of Cycle Types

I conjecture that the frequency of cycle types that occurs in the Galois group of a polynomial over \mathbb{F}_p for p primes between 1 and N is roughly fixed as N gets large. Below I've included two tables, both giving the frequency of cycle types for polynomials over F_p between 1 and N . The first table gives this for $N = 100$ and the second for $N = 31$. The data given supports my hypothesis with exception of $f(x) = x^2 + x + 41$. The ratio of a 1 cycle to a 2 cycle occurring for $N = 1000$ is $0.43 : 0.57$ which does in fact support my conjecture.

| Polynomial | Cycle Type Frequency | | | | | | |
|-----------------------------------|----------------------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $x^2 + x + 41$ | 0.32 | 0.68 | 0 | 0 | 0 | 0 | 0 |
| $x^3 + 2x + 1$ | 0.08 | 0.58 | 0.33 | 0 | 0 | 0 | 0 |
| $x^3 + x^2 - 2x - 1$ | 0.29 | 0 | 0.71 | 0 | 0 | 0 | 0 |
| $x^4 - 2x^2 + 4$ | 0.22 | 0.78 | 0 | 0 | 0 | 0 | 0 |
| $x^4 - x^3 - 4x + 16$ | 0.05 | 0.64 | 0 | 0.32 | 0 | 0 | 0 |
| $x^4 - 2x^3 + 5x + 5$ | 0 | 0.14 | 0.5 | 0.36 | 0 | 0 | 0 |
| $x^4 + 7x^2 + 6x + 7$ | 0.45 | 0.55 | 0 | 0 | 0 | 0 | 0 |
| $x^4 + 3x^3 - 6x^2 - 9x + 7$ | 0.18 | 0.82 | 0 | 0 | 0 | 0 | 0 |
| $x^5 + 36$ | 0.05 | 0.23 | 0 | 0.55 | 0.18 | 0 | 0 |
| $x^5 - 5x + 3$ | 0 | 0.08 | 0.08 | 0.29 | 0.25 | 0.29 | 0 |
| $x^5 + x^3 - 3x^2 + 3$ | 0 | 0.18 | 0.45 | 0 | 0.36 | 0 | 0 |
| $x^5 - 11x^3 + 22x - 11$ | 0.13 | 0 | 0 | 0 | 0.87 | 0 | 0 |
| $x^6 + x + 1$ | 0 | 0 | 0.12 | 0.28 | 0.12 | 0.48 | 0 |
| $x^7 - 2x^6 + 2x + 2$ | 0 | 0 | 0.05 | 0.27 | 0.18 | 0.05 | 0.45 |
| $x^7 + x^4 - 2x^2 + 8x + 4$ | 0 | 0.68 | 0.05 | 0 | 0 | 0.27 | 0 |
| $x^7 + x^5 - 4x^4 - x^3 + 5x + 1$ | 0 | 0.42 | 0 | 0 | 0 | 0 | 0.58 |

Table 5: Frequency of Cycle Types of Galois Groups of Polynomials Over Primes

| Polynomial | Cycle Type Frequency | | | | | | |
|-----------------------------------|----------------------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $x^2 + x + 41$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $x^3 + 2x + 1$ | 0.09 | 0.45 | 0.45 | 0 | 0 | 0 | 0 |
| $x^3 + x^2 - 2x - 1$ | 0.2 | 0 | 0.8 | 0 | 0 | 0 | 0 |
| $x^4 - 2x^2 + 4$ | 0.11 | 0.89 | 0 | 0 | 0 | 0 | 0 |
| $x^4 - x^3 - 4x + 16$ | 0 | 0.63 | 0 | 0.38 | 0 | 0 | 0 |
| $x^4 - 2x^3 + 5x + 5$ | 0 | 0 | 0.56 | 0.44 | 0 | 0 | 0 |
| $x^4 + 7x^2 + 6x + 7$ | 0.38 | 0.63 | 0 | 0 | 0 | 0 | 0 |
| $x^4 + 3x^3 - 6x^2 - 9x + 7$ | 0.11 | 0.89 | 0 | 0 | 0 | 0 | 0 |
| $x^5 + 36$ | 0.13 | 0.25 | 0 | 0.5 | 0.13 | 0 | 0 |
| $x^5 - 5x + 3$ | 0 | 0.1 | 0.1 | 0.3 | 0.1 | 0.4 | 0 |
| $x^5 + x^3 - 3x^2 + 3$ | 0 | 0.22 | 0.22 | 0 | 0.56 | 0 | 0 |
| $x^5 - 11x^3 + 22x - 11$ | 0.1 | 0 | 0 | 0 | 0.9 | 0 | 0 |
| $x^6 + x + 1$ | 0 | 0 | 0.18 | 0.09 | 0.09 | 0.64 | 0 |
| $x^7 - 2x^6 + 2x + 2$ | 0 | 0 | 0 | 0.25 | 0 | 0.13 | 0.63 |
| $x^7 + x^4 - 2x^2 + 8x + 4$ | 0 | 0.56 | 0.11 | 0 | 0 | 0.33 | 0 |
| $x^7 + x^5 - 4x^4 - x^3 + 5x + 1$ | 0 | 0.4 | 0 | 0 | 0 | 0 | 0.6 |

Table 6: Frequency of Cycle Types of Galois Groups of Polynomials Over Primes between 2 and 31

4 Code

4.1 Question 1

```

1 function [Division] = poldiv(n,d,p)
2 %Divides polynomial n by polynomial d modulo p
3 %Note this program makes use of the DocPolynom package and polynomials

```

```

4  %should be given in the DocPolynom form
5
6  n=DocPolynom(mod(double(n),p));
7  d=DocPolynom(mod(double(d),p));
8
9
10
11 zero=DocPolynom([0]);
12 q=zero;
13 r=n;
14 vd=double(d);
15 vq=double(q);
16 vr= double(r);
17
18 dd=size(vd,2)-1;
19 dr=size(vr,2)-1;
20 j=1;
21
22
23 while norm(vr) ~= 0 && dd<=dr
24     lr=vr(1);
25     ld=vd(1);
26     a=ld;
27     C=0;
28     i=1;
29     while i <= p-1 && C==0
30         if mod(a*i,p)==1
31             ldinv=i;
32             C=1;
33         else
34             end
35             i=i+1;
36         end
37         dr;
38         dd;
39         t=zeros(1,dr-dd+1);
40         t(1)=(mod(lr*ldinv,p));
41         t=DocPolynom(t);
42         t;
43         q=plus(q,t);
44         r=minus(r,mtimes(t,d));
45         q=DocPolynom(mod(double(q),p));
46         r=DocPolynom(mod(double(r),p));
47
48         vq=double(q);
49         vr= double(r);
50         r;
51         norm(vr);
52         dd;
53         dr;
54         vr;
55
56         dq=size(vq,2)-1;
57         dr=size(vr,2)-1;
58
59
60
61 end
62
63 q

```

```

64     d
65     r
66     DocPolynom(mod(double(plus(r,mtimes(q,d))),p))
67
68
69
70
71
72
73
74
75 end

```

```

1  function [outputArg1,outputArg2] = polgcd(a,b,p)
2  %finds the gcd of polynomials a,b (given in DocPolynom format) over the
3  %field F_p
4
5  a=DocPolynom(mod(double(a),p));
6  b=DocPolynom(mod(double(b),p));
7  va=double(a);
8  vb=double(b);
9  da=size(va,2);
10 db=size(vb,2);
11 %if da>db
12     % holding=b;
13     %b=a;
14     %a=holding;
15 %end
16 a=DocPolynom(mod(double(a),p));
17 b=DocPolynom(mod(double(b),p));
18 va=double(a);
19 vb=double(b);
20 da=size(va,2);
21 db=size(vb,2);
22
23
24 index=0;
25 divalg=0;
26 while norm(vb) ~= 0
27     T=b;
28     a;
29     b;
30     % Run divisor algorithm
31         n=DocPolynom(mod(double(a),p));
32         d=DocPolynom(mod(double(b),p));
33         n;
34         d;
35
36
37         zero=DocPolynom([0]);
38         q=zero;
39         r=n;
40         vd=double(d);
41         vq=double(q);
42         vr= double(r);
43
44         dd=size(vd,2)-1;
45         dr=size(vr,2)-1;
46         j=1;
47

```



```

48
49     while norm(vr) ~= 0 && dd<=dr
50         lr=vr(1);
51         ld=vd(1);
52         a=ld;
53         C=0;
54         i=1;
55         while i <= p-1 && C==0
56             if mod(a*i,p)==1
57                 ldinv=i;
58                 C=1;
59             else
60                 end
61                 i=i+1;
62             end
63             dr;
64             dd;
65             t=zeros(1,dr-dd+1);
66             t(1)=(mod(lr*ldinv,p));
67             t=DocPolynom(t);
68             t;
69             q=plus(q,t);
70             r=minus(r,mtimes(t,d));
71             q=DocPolynom(mod(double(q),p));
72             r=DocPolynom(mod(double(r),p));
73
74             vq=double(q);
75             vr= double(r);
76
77
78             dq=size(vq,2)-1;
79             dr=size(vr,2)-1;
80
81
82
83
84             q;
85             r;
86         end
87
88     % Finished running divisor algorithm
89     b=DocPolynom(mod(double(r),p));
90     a=DocPolynom(mod(double(T),p));
91     va=double(a);
92     vb=double(b);
93     da=size(va,2);
94     db=size(vb,2);
95 end
96 a
97
98
99
100
101 end

```

4.2 Question 2

```

1 function [outputArg1,outputArg2] = decompcalc4(f)
2 %Calcualtes decomposition group of f from primes 2 to 97

```

```

3
4
5 %Preparation
6 primes=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
7       71, 73, 79, 83, 89, 97];
8
9
10 % primes=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61,
11       67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139,
12       149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227,
13       229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311,
14       313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401,
15       409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491,
16       499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599,
17       601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683,
18       691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797,
19       809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887,
20       907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997];
21
22
23 %primes=ones(1,25);
24 %primes=5*primes;
25 pindex=1;
26     holdf=f;
27     vholdf = double(holdf);
28     dholdf = size(vholdf,2)-1;
29     G_pf=zeros(size(primes,2), dholdf );
30     Group=zeros(size(primes,2),1);
31
32 while pindex <= size(primes,2)
33     cou=1;
34     %Setup
35     p=primes(pindex);
36     vf=mod(double(holdf),p);
37     f=DocPolynom(vf);
38     df=size(vf,2)-1;
39
40     %Check if (f,Df)=1
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

52
53         %Run gcd program on (f,Df)
54
55         a=f;
56         b=Df;
57
58
59         %finds the gcd of polynomials a,b (
                    given in DocPolynom format) over
                    the
60         %field F_p
61
62         a=DocPolynom(mod(double(a),p));
63         b=DocPolynom(mod(double(b),p));
64         va=double(a);
65         vb=double(b);
66         da=size(va,2);
67         db=size(vb,2);
68         if da<db
69             holding=a;
70             a=b;
71             b=holding;
72         end
73         a=DocPolynom(mod(double(a),p));
74         b=DocPolynom(mod(double(b),p));
75         va=double(a);
76         vb=double(b);
77         da=size(va,2);
78         db=size(vb,2);
79
80
81         index=0;
82         divalg=0;
83         while norm(vb) ~= 0
84             T=b;
85             % Run divisor algorithm
86             n=DocPolynom(mod(double(a),p));
87             d=DocPolynom(mod(double(b),p));
88
89
90
91             zero=DocPolynom([0]);
92             q=zero;
93             r=n;
94             vd=double(d);
95             vq=double(q);
96             vr= double(r);
97
98             dd=size(vd,2)-1;
99             dr=size(vr,2)-1;
100             j=1;
101
102
103             while norm(vr) ~= 0 && dd<=dr
104                 lr=vr(1);
105                 ld=vd(1);
106                 ae=ld;
107                 C=0;
108                 i=1;
109                 while i <= p-1 && C==0

```

```

110         if mod(ae*i,p)==1
111             ldinv=i;
112             C=1;
113         else
114             end
115             i=i+1;
116         end
117         dr;
118         dd;
119         t=zeros(1,dr-dd+1);
120         t(1)=(mod(lr*ldinv,p));
121         t=DocPolynom(t);
122         t;
123         q=plus(q,t);
124         r=minus(r,mtimes(t,d));
125         q=DocPolynom(mod(double(q),p));
126         r=DocPolynom(mod(double(r),p));
127
128         vq=double(q);
129         vr= double(r);
130
131
132         dq=size(vq,2)-1;
133         dr=size(vr,2)-1;
134
135
136         divalg=divalg+1;
137         q;
138         r;
139     end
140
141     % Finished running divisor algorithm
142     b=DocPolynom(mod(double(r),p));
143     a=DocPolynom(mod(double(T),p));
144     va=double(a);
145     vb=double(b);
146     da=size(va,2);
147     db=size(vb,2);
148 end
149
150
151
152     gcd=a;
153
154     %Ran gcd program
155
156
157     %Check if gcd(f,Df)=1
158     vgcd=double(gcd);
159
160
161 if size(vgcd,2)==1
162
163     %For each rr compute n_rr
164
165     %Calculating f_rr
166
167     v=f;
168     w=DocPolynom([1 0]);
169     rr=1;

```

```

170         dv=df;
171
172
173
174     while 2*rr <= dv
175
176
177
178         %efficient program to calculate x^p^rr
179         bin=dec2bin(p^rr);
180         sizebin=size(bin,2);
181         powerindex=2;
182         if str2num(bin(sizebin)) == 1
183             nn=DocPolynom([1 0]);
184         else
185             nn=DocPolynom([ 1]);
186         end
187
188         nn_i=DocPolynom([1 0]);
189         while powerindex <= sizebin;
190             nn_i=nn_i^2;
191             n=nn_i;
192             d=v;
193             %run divisor algorithm with n=nn_i, d=v
194
195             n=DocPolynom(mod(double(n),p));
196             d=DocPolynom(mod(double(d),p));
197
198
199
200             zero=DocPolynom([0]);
201             q=zero;
202             r=n;
203             vd=double(d);
204             vq=double(q);
205             vr= double(r);
206
207             dd=size(vd,2)-1;
208             dr=size(vr,2)-1;
209             j=1;
210
211
212             while norm(vr) ~= 0 && dd<=dr
213                 lr=vr(1);
214                 ld=vd(1);
215                 a=ld;
216                 C=0;
217                 i=1;
218                 while i <= p-1 && C==0
219                     if mod(a*i,p)==1
220                         ldinv=i;
221                         C=1;
222                     else
223                         end
224                         i=i+1;
225                 end
226                 dr;
227                 dd;
228                 t=zeros(1,dr-dd+1);
229                 t(1)=(mod(lr*ldinv,p));

```

```

230         t=DocPolynom(t);
231         t;
232         q=plus(q,t);
233         r=minus(r,mtimes(t,d));
234         q=DocPolynom(mod(double(q),p));
235         r=DocPolynom(mod(double(r),p));
236
237         vq=double(q);
238         vr= double(r);
239         r;
240         norm(vr);
241         dd;
242         dr;
243         vr;
244
245         dq=size(vq,2)-1;
246         dr=size(vr,2)-1;
247
248
249
250     end
251
252
253
254
255     %ran divisor algorithm and let nn_i=r
256     nn_i=r;
257     if str2num(bin(sizebin-powerindex+1)) ==1
258         nn=mtimes(nn,nn_i);
259         n=nn;
260         d=v;
261
262         %run divisor algorithm with n=nn and
263         %d=v
264
265         n=DocPolynom(mod(double(n),p));
266         d=DocPolynom(mod(double(d),p));
267
268
269
270         zero=DocPolynom([0]);
271         q=zero;
272         r=n;
273         vd=double(d);
274         vq=double(q);
275         vr= double(r);
276
277         dd=size(vd,2)-1;
278         dr=size(vr,2)-1;
279         j=1;
280
281
282         while norm(vr) ~= 0 && dd<=dr
283             lr=vr(1);
284             ld=vd(1);
285             a=ld;
286             C=0;
287             i=1;
288             while i <= p-1 && C==0
289                 if mod(a*i,p)==1

```

```

290                                     ldinv=i;
291                                     C=1;
292                                     else
293                                     end
294                                     i=i+1;
295                                     end
296                                     dr;
297                                     dd;
298                                     t=zeros(1,dr-dd+1);
299                                     t(1)=(mod(lr*ldinv,p));
300                                     t=DocPolynom(t);
301                                     t;
302                                     q=plus(q,t);
303                                     r=minus(r,mtimes(t,d));
304                                     q=DocPolynom(mod(double(q),p));
305                                     r=DocPolynom(mod(double(r),p));
306
307                                     vq=double(q);
308                                     vr= double(r);
309                                     r;
310                                     norm(vr);
311                                     dd;
312                                     dr;
313                                     vr;
314
315                                     dq=size(vq,2)-1;
316                                     dr=size(vr,2)-1;
317
318
319
320                                     end
321
322
323                                     %rand divisor algorithm and let nn=r
324                                     nn=r;
325                                     end
326                                     powerindex=powerindex+1;
327                                     end
328                                     nn=DocPolynom(mod(double(nn),p));
329                                     w=nn;
330
331                                     %end of efficient program to calculate w^p
332                                     n=nn;
333                                     d=v;
334
335                                     %Run divisor algorithm
336
337                                     n=DocPolynom(mod(double(n),p));
338                                     d=DocPolynom(mod(double(d),p));
339
340
341
342                                     zero=DocPolynom([0]);
343                                     q=zero;
344                                     r=n;
345                                     vd=double(d);
346                                     vq=double(q);
347                                     vr= double(r);
348
349                                     dd=size(vd,2)-1;

```

```

350         dr=size(vr,2)-1;
351         j=1;
352
353
354         while norm(vr) ~= 0 && dd<=dr
355             lr=vr(1);
356             ld=vd(1);
357             a=ld;
358             C=0;
359             i=1;
360             while i <= p-1 && C==0
361                 if mod(a*i,p)==1
362                     ldinv=i;
363                     C=1;
364                 else
365                     end
366                     i=i+1;
367             end
368             dr;
369             dd;
370             t=zeros(1,dr-dd+1);
371             t(1)=(mod(lr*ldinv,p));
372             t=DocPolynom(t);
373             t;
374             q=plus(q,t);
375             r=minus(r,mtimes(t,d));
376             q=DocPolynom(mod(double(q),p));
377             r=DocPolynom(mod(double(r),p));
378
379             vq=double(q);
380             vr= double(r);
381
382
383             dq=size(vq,2)-1;
384             dr=size(vr,2)-1;
385
386
387
388         end
389
390
391         %Once ran divisor algorithm let w=r
392
393         w=r;
394
395
396
397
398
399
400         vw=double(w);
401         dw=size(vw,2)-1;
402
403
404         %Run GCD program on w-x and v
405         a=minus(w,DocPolynom([1 0]));
406         b=v;
407
408         a=DocPolynom(mod(double(a),p));
409         b=DocPolynom(mod(double(b),p));

```



```

410 va=double(a);
411 vb=double(b);
412 da=size(va,2);
413 db=size(vb,2);
414 %if da>db
415     % holding=b;
416     %b=a;
417     %a=holding;
418 %end
419 a=DocPolynom(mod(double(a),p));
420 b=DocPolynom(mod(double(b),p));
421 va=double(a);
422 vb=double(b);
423 da=size(va,2);
424 db=size(vb,2);
425
426
427 index=0;
428 divalg=0;
429 while norm(vb) ~= 0
430     T=b;
431     a;
432     b;
433     % Run divisor algorithm
434         n=DocPolynom(mod(double(a),p));
435         d=DocPolynom(mod(double(b),p));
436         n;
437         d;
438
439
440         zero=DocPolynom([0]);
441         q=zero;
442         r=n;
443         vd=double(d);
444         vq=double(q);
445         vr= double(r);
446
447         dd=size(vd,2)-1;
448         dr=size(vr,2)-1;
449         j=1;
450
451
452         while norm(vr) ~= 0 && dd<=dr
453             lr=vr(1);
454             ld=vd(1);
455             a=ld;
456             C=0;
457             i=1;
458             while i <= p-1 && C==0
459                 if mod(a*i,p)==1
460                     ldinv=i;
461                     C=1;
462                 else
463                     end
464                     i=i+1;
465             end
466             dr;
467             dd;
468             t=zeros(1,dr-dd+1);
469             t(1)=(mod(lr*ldinv,p));

```

```

470         t=DocPolynom(t);
471         t;
472         q=plus(q,t);
473         r=minus(r,mtimes(t,d));
474         q=DocPolynom(mod(double(q),p));
475         r=DocPolynom(mod(double(r),p));
476
477         vq=double(q);
478         vr= double(r);
479
480
481         dq=size(vq,2)-1;
482         dr=size(vr,2)-1;
483
484
485
486
487         q;
488         r;
489     end
490
491     % Finished running divisor algorithm
492     b=DocPolynom(mod(double(r),p));
493     a=DocPolynom(mod(double(T),p));
494     va=double(a);
495     vb=double(b);
496     da=size(va,2);
497     db=size(vb,2);
498 end
499
500 %Once ran GCD program let g=gcd
501 g=a;
502 w;
503 v;
504 g;
505 rr;
506
507 vg=double(g);
508 dg=size(vg,2)-1;
509 if dg ~= 0
510
511     g;
512     rr;
513     G_pf(pindex,rr)= dg/rr;
514
515     n=v;
516     d=g;
517     %Run divisor algorithm
518
519     n=DocPolynom(mod(double(n),p));
520     d=DocPolynom(mod(double(d),p));
521
522
523
524     zero=DocPolynom([0]);
525     q=zero;
526     r=n;
527     vd=double(d);
528     vq=double(q);
529     vr= double(r);

```

```

530
531         dd=size(vd,2)-1;
532         dr=size(vr,2)-1;
533         j=1;
534
535
536         while norm(vr) ~= 0 && dd<=dr
537             lr=vr(1);
538             ld=vd(1);
539             a=ld;
540             C=0;
541             i=1;
542             while i <= p-1 && C==0
543                 if mod(a*i,p)==1
544                     ldinv=i;
545                     C=1;
546                 else
547                     end
548                 i=i+1;
549             end
550             dr;
551             dd;
552             t=zeros(1,dr-dd+1);
553             t(1)=(mod(lr*ldinv,p));
554             t=DocPolynom(t);
555             t;
556             q=plus(q,t);
557             r=minus(r,mtimes(t,d));
558             q=DocPolynom(mod(double(q),p));
559             r=DocPolynom(mod(double(r),p));
560
561             vq=double(q);
562             vr= double(r);
563
564
565             dq=size(vq,2)-1;
566             dr=size(vr,2)-1;
567
568
569
570         end
571
572         %Once ran divisor algorithm let v=q
573         v=q;
574         vv=double(v);
575         dv=size(vv,2)-1;
576
577         %cut the rest out
578
579
580         %end of cutting out
581
582         end
583         rr=rr+1;
584     end
585
586     if dv ~= 0
587
588         v;
589         dv;

```

```

590         G_pf(pindex,dv)=1;
591
592     end
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611     else
612         G_pf(pindex,:)=(-1)*ones(1,dholdf);
613     end
614
615     %Calculate size of group
616
617
618     if G_pf(pindex,1) == -1
619         Group(pindex,1)=-1;
620     else
621         i=1;
622         N=zeros(1,df);
623         while i <= df
624
625             if G_pf(pindex,i)==0
626
627                 N(i)=1;
628
629             else
630                 N(i)=i;
631
632             end
633             i=i+1;
634         end
635         N;
636         x=lcm(sym(N));
637         Group(pindex,1)=x;
638
639     end
640
641     pindex=pindex+1;
642 end
643 G_pf
644 Group
645 end

```