公立はこだて未来大学 2016 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2016 System Information Science Practice

Group Report

プロジェクト名

FUN-ECM プロジェクト

Project Name

FUN-ECM Project

グループ名

A グループ

Group Name

A Group

プロジェクト番号/Project No.

15-A

プロジェクトリーダ/Project Leader

1014129　池野竜將　Ryusuke Ikeno

グループリーダ/Group Leader

1014129　池野竜將　Ryusuke Ikeno

グループメンバ/Group Member

1014068　駒ヶ嶺壮　Sou Komagamine
1014109　伊藤有輝　Yuki Ito
1014129　池野竜將　Ryusuke Ikeno
1014137　千葉大樹　Daiju Chiba
1014164　橋本和典　Kazunori Hashimoto
1014168　山下哲平　Teppei Yamashita
1014209　源啓多　Keita Minamoto
1013150　亀谷浩也　Hiroya Kametani

指導教員

白勢政明　由良文孝

Advisor

Masaaki Shirase　Fumitaka Yura

提出日

2017 年 1 月 18 日

Date of Submission

January 18, 2017

概要

私たちのプロジェクトの目的は，より大きな桁数の素因数を見つけることである．素因数分解は，約 40 年前から重要になってきている．その理由は，RSA 暗号にある．RSA 暗号は，安全性を 2 つの大きな素因数からなる合成数の素因数分解が難しいことに依存している．しかし，技術の発展とともに素因数分解が従来よりも容易になってきてしまっているため，RSA 暗号が破られる可能性が高くなっている．そこで今注目されているのが楕円曲線暗号である．楕円曲線暗号は，RSA 暗号と同じ鍵長で高い安全性を保障することができる．そこで私たちは素因数分解をより簡単なものとすることで，RSA 暗号から楕円曲線暗号を主流とさせたい．

私たちは，大きな素因数の発見のために，色々な文献を読んでその中から素因数分解を行うプログラムの改良法を発見する理論班と，それらの理論を利用して実際にプログラムの実装・改良を行い，プログラムを高速化させるプログラミング班に分かれて活動を行った． 理論班は，素因数分解がより高速に行われるようなアルゴリズムの発見を目標とした．楕円曲線法 (ECM) のプログラムは点の加算の繰り返しで行われるため，加算の計算コストを減らすことで高速な計算を可能とするための活動を行った．Atkin Morain ECPP を利用することで，従来の ECM よりも計算コストを削減できることを発見した． プログラミング班は，前年度に作成された素因数分解プログラムをさらに高速化することを目標とした．前年度と同様に大きな数を扱うために，任意精度演算ライブラリの GMP を使用した．また，プログラムの並列実行を行うために，並列プログラムの為の API である OpenMP を導入した．前年度に実装されたエドワーズ曲線よりも効率よく計算を行うため，extended twisted Edwards coordinates を採用した．同じ合成数に対してプログラムを実行する際の因数の発見確率をあげるために，パラメータ Y の値をランダムに設定した． また，理論班とプログラミング班で情報の交換を行ったり，協力を行ったりなど，2 つの班の活動により，素因数分解を高速に行うことができるプログラムが完成した． 更に，今年度からの活動としてより多くの人に ECM について知ってもらうため，私たち FUN-ECM の活動内容を広報することにした．広報の方法として新たに広報班を結成し，簡単に閲覧できるように Web ページを作成することにした．閲覧するターゲットは主に情報系の大学生とした． 広報班の活動により，半期で設定したターゲットに向けた Web ページを作成することができた．

キーワード 素因数分解, 楕円曲線法, ECMNET, エドワーズ曲線, 射影座標, RSA 暗号

（※文責: 山下哲平）

Abstract

The goal of our project team is to find prime factor as large as possible. Prime factorizations have become more important since about forty years ago because the difficulty of prime factorization is related to security of RSA cryptosystems which used for the Internet. However, prime factorization is getting to easier by development in technology. Therefore, security of RSA is less compared to previously. That's why Elliptic Curve Cryptography (ECC) is paid more attention than RSA now. ECC ensure security better than RSA cryptosystem with same key length. Accordingly, we make prime factorization simplify, we would like to change main cryptosystem from RSA cryptosystem to ECC.

We divided into two groups, one is "theory group" that reads various literature and find algorithms for prime factorizations to calculate faster, the other is "programming group" that write a program based on the algorithms.

"Theory group" aims to find algorithm of prime factorizations to calculate faster. The ECM program repeats process of addition law many times over, therefore we reduced calculation. To access Atkin Morain construction, we were successful in calculation faster compared to previously.

"Programming group" aims to improve program of last year project team faster than before. To treat large number likewise last year, we used arbitrary-precision arithmetic library called GMP. Also, we parallelize the program, we introduce Open MP which is API for parallel program. We implement extended twisted Edwards coordinates efficient than Edwards curve implemented last year. Also, we set random Y' s value to raise finding assembly towards same composite numbers.

We exchange information and cooperate "theory group" and "programming group", we made a program that is to perform factorization in prime numbers fast.

Furthermore, as activity from this fiscal year, we decided public relations activities of FUN-ECM. As method of public relations we formed public relation (P.R) group, and we decided to make webpage can browse easily. We established targets of browsing are university students of information system. According to P.R group, we could make webpage directed at targets half year.

Keyword Elliptic Curve Method, prime factorization, ECMNET, Twisted Edwards Curve, Extended Twisted Edwards Coordinates, RSA cryptosystem

（※文責: 山下哲平）

目次

第 1 章　背景

ECM(楕円曲線法) を利用した素因数分解は，実際のインターネットで使われる暗号技術の安全性の確認に必須であるため近年重要になってきており，それを利用し ECM-NET にランクインすることが私たちの目的である．

（※文責: 駒ヶ嶺壮）

1.1　本プロジェクトの背景

現在インターネットを含む通信での鍵暗号技術 (特に鍵共有と認証) においての主流は RSA 暗号である．RSA 暗号とは公開鍵暗号の一つで，大きな合成数を素因数分解することの難しさを安全性の根拠にした暗号である．つまり，RSA 暗号は公開鍵の整数を素因数分解できると解読可能となる．楕円曲線法は数体篩法とともに最良な素因数分解法の 1 つである．

コンピューターの並列処理能力と計算能力の向上等で現在使用されている鍵での RSA 暗号方式は近い将来解読される危険性が指摘されるようになった．ここで，今後の暗号技術には RSA に代わるものとして楕円曲線暗号 (ECC) が注目され始めている．ECC は現在の暗号技術において最も重要とされている手法である．これは，暗号化・復号においてある有限体上の楕円曲線の点の加算を用いることにより，RSA 暗号と同じ鍵長でより解読が難しくなるからである．

以上のように，楕円曲線は RSA 暗号への攻撃 (つまり安全性解析) 及び ECC の構成に使用され，両方とも社会的に重要性が高い．更に興味深いことに，両者の主要演算は楕円曲線のスカラー倍算出るため，プログラムに類似点が多い．私たちは ECM に焦点をあて，FUN-ECM の ECM-NET にランクインを目指すことで函館から楕円曲線を利用する技術の重要性について発信することを目標として掲げた．

（※文責: 山下哲平）

1.2　ECM-NET とは

ECM-NET とは，楕円曲線法を用いて大きい桁数の素因数分解をみつけることを目的とした競争サイトである．ECM-NET には現在登録されている素因数分解よりも大きな素因数を見つけることで誰でもランクインすることが可能である．過去にランクインした日本人は K.Aoki 氏と T.Izu 氏の 2 名である．

（※文責: 駒ヶ嶺壮）

1.3　課題の概要

FUN-ECM が ECM-NET へのランクインを目指すには大きい桁数の素因数を見つけなければいけないことから楕円曲線を用いた素因数分解のプログラムの並列処理と高速化を目指す．また，

FUN-ECM Project

本プロジェクトの活動を Web サイト等を用いて外部に発信する.

（※文責: 駒ヶ嶺壮）

第 2 章　到達目標

2.1　本プロジェクトにおける目的

FUN-ECM が ECM-NET にランクインするためには去年のプログラムをより改善する必要がある．この目標を達成するにあたって，2 つの目標を立てることにした．

（※文責: 伊藤有輝）

2.1.1　プログラムの高速化

ECM-NET にランクインするためには，巨大な素因数を発見しなければならない．巨大な素因数を発見するためには桁数の大きい合成数を素因数分解する必要があるが，それには多大な時間がかかってしまう．また，ECM は一度の試行で素因数を必ず発見できるとは限らず，数千回程度の試行が必要となる．そのためプログラムの処理を効率の良いアルゴリズムに変更し，処理を高速化させる必要がある．この目標を達成するにあたって，2 つの目標を立てることにした．

- 昨年度のプログラムのアルゴリズムの理解
- 昨年度のプログラムの書き換えたものの実装

まず，昨年度のプログラムを高速化するにはアルゴリズムの理解が必要である．また，楕円曲線法では大学までの学習で使用していない数学の概念を使用するため，基礎学習を行う．

（※文責: 伊藤有輝）

2.1.2　FUN-ECM の活動発信

今年度では，ただランクインを目指すだけでなく，函館から楕円曲線，素因数分解の重要性について発信することに決め，ホームページを設立することとした．

（※文責: 伊藤有輝）

2.2　課題達成の為の班分け

前年度のプロジェクトでは前期で楕円曲線法についての学習を行い，後期でアルゴリズムの提案・実装を行っていた．しかし，このような日程でプロジェクトを進行していくと以下のような問題が発生した．

- 実際にプログラムを実装する期間が少ない
- 完成したプログラムを試行する期間が少ない
- 巨大な合成数を分解するには性能が不足している

そのため，本プロジェクトでは 5 月中旬まで全員で最低限の基礎学習を行い，そこから理論班とプログラミング班の 2 つに分けて作業を行うこととした．また，後期には広報班を作成し，3 つの作業を並行で行うこととした．以下にそれぞれの班の課題について述べる．

理論班

ECM について理解を深め，高速化の新たなアルゴリズムを提案する．

プログラミング班

基礎学習や理論班がまとめたアルゴリズムを実装し高速化を行う．

広報班

ECM について理解してもらえるような Web ページの作成をする．

（※文責: 伊藤有輝）

第 3 章　前期活動内容

プロジェクトが始まった当初，ほぼ全員楕円曲線についての前提知識がなかったため，昨年も前提知識を身に着けるために使われた全員楕円曲線についての資料を全員で輪読し，理解した．その際，理解できなかったところを由良先生，白勢先生に解説してもらった．それにより，楕円曲線法のアルゴリズムを理解するためにあたっての基礎知識を学んだ．その後，プロジェクト全体をプログラムの高速化につながる理論を探し，学習してアルゴリズムをノートにまとめる理論班，理論班がノートにまとめたアルゴリズムをプログラムに実装するプログラミング班に分けてプロジェクトを進めた．

（※文責: 伊藤有輝）

3.1　基礎学習

去年のプログラムを理解するために 5 月の中旬まではメンバ全員が教授の指導のもとで楕円曲線法のアルゴリズムや基礎知識ついての基礎学習を行った．具体的な内容は以下の通りである．

有限体

素数 p に対し，0 から $p-1$ までの整数の集合 $\mathbb{F}_p = \{0, 1, \cdots, p-1\}$ を有限体と言う．$\mathbb{F}_p$ では四則演算が可能であり，ECM ではこの範囲で考える．

Euclid の互除法

自然数 a, $b(a \geqq b)$ に対して以下の操作を繰り返し余りが 0 になるまで行うことによって a , b の最小公倍数を求めるものである．

Algorithm 1 Euclidean Algorithm

Require: $a, b \in \mathbb{N}, \quad a, b \neq 0, \quad a \geq b$
Ensure: $\gcd(a, b)$
  while $b \neq 0$ **do**
    $q \leftarrow a/b$
    $r \leftarrow a \mod b$
    $a \leftarrow b$
    $b \leftarrow r$
  end while
  return a, b

a, b の最大公約数を $\gcd(a, b)$ と表記する．

拡張 Euclid の互除法

与えられた整数 a, b, c に対し，未知数 x, y に関する一次方程式 $ax + by = c$ の整数解を求める問題を一次不定方程式という．ここで，自然数 a, b に関する一次不定方程式 $ax + by = gcd(a, b)$ を満たす無数の整数 x, y は拡張Euclid の互除法を用いることで効率よく求めることができる．これは Euclid の互除法で行った操作を逆に行うことで解を得

る．$gcd(174,\ 69) = 3$ を例にとって考える．

$$
\begin{aligned}
174/69 &= 2 * 69 + 36 \\
69/36 &= 1 * 36 + 33 \\
36/33 &= 1 * 33 + 3 \\
33/3 &= 11
\end{aligned}
$$

となるので

$$
\begin{aligned}
3 &= 36 - 33 * 1 \\
&= 36 - (69 - 36 * 1) * 1 \\
&= 69 * (-1) + 36 * 2 \\
&= 69 * (-1) + (174 - 69 * 2) * 2 \\
&= 174 * 2 + 69 * (-5)
\end{aligned}
$$

以上より,$174x + 69y = 3$ の解 $(x, y) = (2, -5)$ を得ることができる．有限体 $\mathbb{F}_p$ において除算 a/b を計算する場合, p と b は互いに素なので, 拡張 Euclid の互除法により不定方程式 $px + by = 1$ の解 (x, y) を求めることができる．このとき $px + by = 1$ となるので, 有限体 $\mathbb{F}_p$ 上では $by = 1$ となり,両辺を b で割ることで,$\mathrm{b}^{-1} = y$ が成立する．したがって $a \div b = a \times b - 1 = a \times y$ と変形することで, 除算を乗算に置き換えて計算できる．プログラムにおいて，除算を乗算に置き換えることは計算量の削減につながるが，今回のプロジェクトでは GMP ライブラリを用いたことでこれを実装することはなかった．

楕円曲線の定義方程式

$a, b \in \mathbb{F}_p$に対して $y^2 = x^3 + ax + b$ で定義される曲線を素体 Fp 上の楕円曲線という．

楕円曲線の加算・2 倍算

(加算) 楕円曲線上のある 2 点 P, Q を通る直線を l とすると,楕円曲線と直線 l の 3 つ目の交点 R’ $(= P \times Q)$ の x 軸に関する対称点を R とする．このとき 2 点 P, Q の和を $P + Q := R$ と定義し，楕円曲線の加算という．

(2 倍算) 楕円曲線上の 1 点 P で加算を考えるときは 2 点 P, P の通る直線 $(= P$ の接線) を l として考える．この時，楕円曲線と直線 l の P 以外の交点の x 軸に関する対称点を R としたとき，$2P = P + P := R$ とできる．これが楕円曲線の 2 倍算である．

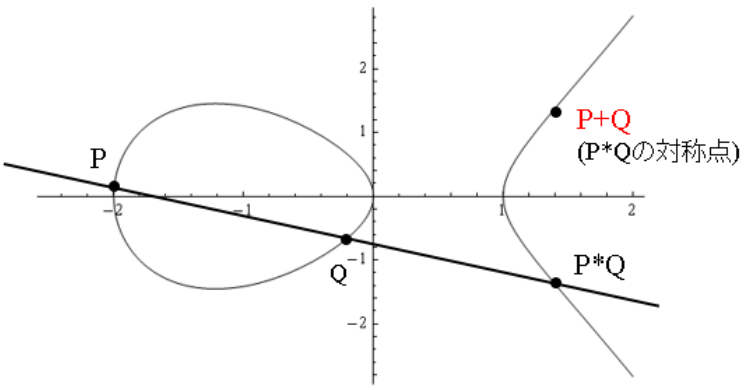

図 3.1　楕円曲線の加算

楕円曲線のスカラー倍

点 P と整数 m を使用して，$mP = P + P + P + P + \cdots + P(m$ 個の和) と表すことができる．これを楕円曲線のスカラー倍という．

加算公式

楕円曲線法のアルゴリズム

N を素因数分解したい合成数とする．$\mathbb{Z}/N\mathbb{Z}$ 上で，楕円曲線 E を構成して，点

$$P \in E(\mathbb{Z}/N\mathbb{Z}) \tag{3.1}$$

をとる．初めに P の座標を決めてから E を構成しても良い．

次に適切な B_1，$L = 2, 3, \cdots B_1$の最小公倍数とする．LP の計算の過程で生じる点の座標の分母 d が $\gcd(N, d) \neq 1$ となると N の約数を発見できる．

最後まで $\gcd(N, d) = 1$ ならば，E と P を選びなおしてやり直す．適切な B_1を選ぶことで，ECM は高速な素因数分解法になることが知られている．以下に具体的なアルゴリズムを示す．

Algorithm 2 Basic ECM Algorithm

Require: N is composite number, E is elliptic curve, $P = (x_0, y_0, Z_0) \in E(Z_n)$ is initial point, B_1 is smoothness bound

Ensure: q is factor of N, $1 \leq q \leq N$, or FAIL.

Phase 1.

$k \leftarrow \prod_{p \leq B_1} p^{\log_p B_1}$

$Q_0 \leftarrow kP_0$

$q \leftarrow \gcd(z_{Q_0}, N)$

if $q \geq 1$ **then**

  return q

else

  return FAIL

end if

以上のことを基礎学習として学んだ．以下の章ではに 2 班に分かれた後の理論班の活動内容を記述する．

（※文責: 橋本和典）

3.2 理論班

理論班では新たなアルゴリズムを探し，プログラミング班に新たな高速化手法の提案を行った．以下に具体的な内容を述べる．

3.2.1 Twisted Edwards Curve の理解

ECM の高速化アルゴリズムを実装するにあたって，先人の知恵を得ようと思いインターネットで類似研究の論文を検索し，その論文を解読することによって高速化アルゴリズムをプログラムに

導入しようと考えた．その際，Twisted Edwards Curves Revisited というエドワーズ曲線についての英語の論文が見つかったため，私たちはこの論文を読解することにした．

この論文は，最初に一般的な楕円曲線アルゴリズムより，エドワーズ曲線の方が計算コストは低く，速いスピードで素因数を求めることができるということが説明されており，そのエドワーズ曲線の数学的な理論とプログラム実装のためのアルゴリズムが書かれていた．

エドワーズ曲線については基礎学習で学んでいなかったため，私たちはエドワーズ座標を学習した．その中では射影座標が使用されていた．射影座標とは一般的な座標 (x,y) に対して $x = \frac{X}{Z}, y = \frac{Y}{Z}$ を満たす X, Y, Z を用いて（X, Y, Z）と表す座標であり，射影座標を用いるとECMアルゴリズムを高速化することができる．具体的な定義は以下の通りである．

射影座標

$$(X, Y, Z) = (\lambda X, \lambda Y, \lambda Z) = (\frac{X}{Z}, \frac{Y}{Z}, 1)\ (Z \neq 0)$$

理論班では，この拡張エドワーズ座標の理論を学ぼうとしたが，知識が乏しく，わからない変数が出てきたため，アルゴリズムだけを理解し，定義，証明などの理論を理解することはあきらめた．

（※文責: 伊藤有輝）

3.2.2 Atkin-Morain ECPP

次に Atkin-Morain ECPP という ECM の初期座標を決定するアルゴリズムの理解に励んだ. 昨年度までは ECM の初期座標として (2,2) を用いて，素因数分解が完了できなければ (2,3)，(2,4)…といったように Y 座標を 1 ずつ動かすようにしていたが，今年度では少しでも因数を見つける確率を上げることが見込める Atkin-Morain ECPP を理解することにした．Atkin-Morain ECPP では新たな楕円曲線 $T^2 = S^3 - 8S - 32$ の点を用意し，$(S, T) = (12, 40)$ に対して $n(S, T)$ の座標 (s, t) を用いて以下を定める．

$$\alpha = \frac{(s-9)+1}{t+25}, \beta = \frac{2\alpha(4\alpha+1)}{8\alpha^2-1} \tag{3.2}$$

これらを用いることによって，素因数分解に用いる楕円曲線の初期座標を求めることができる．具体的には以下の通りである．

Algorithm 3 Atkin-Morain ECPP Algorithm

Require: $\alpha, \beta, s, t, \in \mathbb{N}$

Ensure: (X, Y)

$(s, t) \leftarrow (12, 40)$

while Prime factor is not found **do**

$\alpha \leftarrow \dfrac{(s-9)+1}{t+25}$

$\beta \leftarrow \dfrac{2\alpha(4\alpha+1)}{8\alpha^2-1}$

$d \leftarrow \dfrac{2(2\beta-1)^2-1}{(2\beta-1)^4}$

$E : x^2 + y^2 = 1 + dx^2y^2$

$X \leftarrow \dfrac{(2\beta-1)(4\beta-3)}{6\beta-4}$

$Y \leftarrow \dfrac{(2\beta-1)(t^2+50t-2s^3+27s^2-104)}{(t+3s-2)(t+s+16)}$

Run ECM with $E : x^2 + y^2 = 1 + dx^2y^2$ and (X, Y)

$(s, t) \leftarrow 2(s, t)$

end while

このアルゴリズムを用いると具体的には従来の 1.5 倍ほど高速化できる見込みであるが，これは論文上のデータである．したがって，後期はプログラミング班が実装し，どのくらい速くなるかどうかを検証したいと考えている．

（※文責: 伊藤有輝）

3.2.3 入門書の理解

私たちがプロジェクト活動をしていくにあたって，論文などを読み進めていく際に基礎学習では学ばなかった数学用語が頻出していた．そのため, 論文の読解が難しく，たびたび中止するということが多かった．そこで, 楕円曲線についての入門書を読み，数学的知識を増やすことにより，楕円曲線についての理解を深めることにした．入門書は，J.H. シルヴァーマン，J. テイト著の「楕円曲線論入門」を使用した．その中でも，基礎学習の部分も多く含まれている P，Q，P ∗ Q と P ＋Q の関係性についての章と加算公式の導出についての章を読み進めることにした．しかし, 前期中では時間が少なく, 基礎学習で学んだ部分の内容と重複していたため, 入門書の理解が十分に論文の読解力への大きな改善が行われたとは言えなかった．

（※文責: 駒ヶ嶺壮）

3.3 プログラミング班

プログラミング班では，昨年度の FUN-ECM プロジェクトで作成した ECM プログラムをさらに高速化するために，4 月から 5 月にかけて行った全体での基礎学習や，理論班がまとめた理論・アルゴリズムを元にプログラムを変更した．主に，射影座標や extended twisted Edwards coordinates を用いて乗算・除算を減らすことによって高速化を図った．また，前年度のプログラ

ムの不具合等も改善した．具体的には以下の通りである．

（※文責: 源啓多）

3.3.1 座標変換の際の冗長なコストの削減

前年度のプロジェクトで作成された ECM プログラムでは，スカラー倍をする際の座標をアフィン座標から射影座標に変換することで計算効率を上昇させていた．このアフィン座標から射影座標への変換は複数回呼び出される為，ECM プログラムの計算コストに影響する．Algorithm 4 にアルゴリズムを記す．

Algorithm 4 Affine Coordinates to Projective Coordinates (Past ver.)

Require: (AX, AY) is Affine, (PX, PY, PZ) is Projective, $N \geq 2$

Ensure: (PX, PY, PZ)

$Z \leftarrow Random(0 \leq Z < N)$

if $Z = 0$ **then**

$Z \leftarrow 1$

end if

$AX \leftarrow AX \times Z$

$AY \leftarrow AY \times Z$

$AX \leftarrow AX \mod N$

$AY \leftarrow AY \mod N$

$(PX, PY, PZ) \Leftarrow (AX, AY, Z)$

前述の冗長部分として乗算が 2 回と *mod* の計算が 2 回発生している．プログラミング班では，Z の値を 1 に設定することで乗算と *mod* の計算を省略できると考えた．プログラムを一通り読み直し，問題が発生しないことを確認したのち，新たなアルゴリズムを実装した．Algorithm 5 に新しいアルゴリズムを示す．

Algorithm 5 Affine Coordinates to Projective Coordinates (New ver.)

Require: (AX, AY) is Affine, (PX, PY, PZ) is Projective, $N \leq 2$

Ensure: (PX, PY, PZ)

$Z \leftarrow 1$

$(PX, PY, PZ) \Leftarrow (AX, AY, Z)$

（※文責: 源啓多）

3.3.2 Extended twisted Edwards coordinates の実装

前年度のプロジェクトで作成された ECM プログラムでは，twisted Edwards curve を利用している．今回のプログラミング班ではさらに extended twisted Edwards coordinates を用いた．extended twisted Edwards coordiantes はエドワーズ曲線のスカラー倍を高速化するための座標系であり，以下で定義される補助座標 T を加えた 4 つの座標でスカラー倍を行う．

Extended twisted Edwards coordinates

射影座標（X,Y,Z）をに対し，T=$\frac{XY}{Z}$ という補助座標を加える．これを Extended twisted Edwards coordinates と呼ぶ.

$$(X, Y, Z) \to (X, Y, T, Z)$$

（※文責: 源啓多）

3.3.3 楕円曲線の生成法の変更

楕円曲線法を利用した ECM プログラムは，楕円曲線を生成しその座標を利用し素因数分解を行うプログラムである．また，本プロジェクトで素因数分解しようと試みている合成数は 200 桁前後のため，1 度の試行では素因数分解できないことが多くある．よって，同じ合成数に対して複数回の試行をすることを想定してプログラムを作成する必要がある．前年度のプログラムでは，楕円曲線を生成する際に，Y 値を for 文のカウンタを利用して 1 から順に決めるアルゴリズムを採用していた．そのため，複数回試行した際に同じ曲線を使用してしまうことが多くあり，効率が落ちていたと仮定した．そこで曲線を生成する際に使用している Y 値に乱数を使用することとした.

（※文責: 源啓多）

3.4 中間発表

3.4.1 準備

ポスター

初めに，前年度のプロジェクトで作成されたポスターを参考に構成を決定した．次に，概要，基礎学習，理論班，プログラミング班の 4 つの項目に分け，作成を分担した．ポスターの作成には「Microsoft PowerPoint」というソフトウェアを使用した．ポスターが完成次第，理論班・プログラミング班でレビューを行い，誤字脱字等を修正した．しかしポスターレビューが不十分だったため，最終的に完成したポスターで誤植が見つかってしまった.

（※文責: 亀谷浩也）

プレゼンテーション資料

本プロジェクトの内容を説明するには，ポスターだけでは足りないと判断しプレゼンテーション資料を作成することに決定した．作成にあたって，まず 1 名がプレゼンテーションの大まかな流れを作成し，各自作成する章を分担した．プレゼンテーション資料の作成には「Microsoft PowerPoint」というソフトウェアを使用した．また，一度完成したプレゼンテーション資料を先生にレビューしていただき，資料中のグラフの不備や内容についての助言を受けた．それを受け，文章や図の修正を行った．これにより，より見やすいプレゼンテーション資料が完成した.

（※文責: 亀谷浩也）

原稿

前述のプレゼンテーション資料の作成と並行して，発表用の原稿の作成を行った．こちらも 1 名が大まかな流れを作成し，各自作成する章を分担した．特に楕円曲線法については，何も知らない聴衆でもわかりやすく説明できるように，専門的な用語を最小限にするように注意して作成した．何度か原稿とプレゼンテーション資料を使用しプレゼン練習を行い，伝わりにくい表現や冗長な表現を修正した．

（※文責: 亀谷浩也）

3.4.2 発表

発表は前後半で 4 人ずつに分かれ，発表を行った．それぞれが自分の担当する部分を読み上げ，その間他の 3 人は評価アンケート配布や，ポスターに関しての質問に対応した．発表途中にプロジェクターの電源が落ちてしまうというアクシデントがあったが，落ちている間は PC の画面を直接見せることでプレゼンを行い，他の 3 人で復旧作業を行った．発表後に評価アンケートの集計を行った結果，発表技術は 10 点中平均 7.1 点，発表内容は 10 点中 7.5 点だった．コメントでは内容を理解していた人と全く理解できない人が分かれていたため，さらに前提知識のない聴衆にも伝わるような内容にしていきたい．

（※文責: 亀谷浩也）

第 4 章　後期活動内容

後期の活動は，前期の活動に加えて広報活動を行った．また，理論班には作成したプログラムの検証を行った．

（※文責: 橋本和典）

4.1　理論班

理論班は，新たに取り入れた理論とプログラミング班によって実装されたプログラムがどれほど高速化されたのかを検証した．

（※文責: 橋本和典）

4.1.1　活動目的

ECM プログラムの効率化は本プロジェクトの目的である ECMNET へのランクインに必要不可欠な要素である．効率化のために取り入れたプログラムが処理時間にどのような影響がでるのかを調べることは今後プログラムを改善していくうえで重要な項目である．今回は昨年度に使用された ECM プログラムと今年度のプログラムの比較評価を行った．昨年度と今年度のプログラムの主な違いは，スカラー倍算の高速化（kP）アルゴリズムの実装，Atkin-Morain ECPP の実装，そして Stage2 の実装である．この実装により ECM プログラムの素因数発見の効率を上げ，処理速度の向上に貢献したと予想をふまえて，今年度のプログラムが昨年度よりどのように改善されたのかを検証する．またこの変更がプログラムにどのような効果をもたらすかを知ることで，今後のプログラムの改善に役立つことができる．

（※文責: 橋本和典）

4.1.2　活動内容

検証班は今年度にプログラミング班が作ったプログラムと昨年度のプログラムを実行しどれだけ改善したかを検証することにした. 今回は素因数分解アルゴリズムの処理速度を評価したいと考えた．今年度の ECM プログラムでは素因数の発見確率を上げたことから素数ではなく合成数を入力することにした．検証結果を統計的に信憑性を持たせるためにいくつかの統計方法を調べた．しかし，どの検証方法も今回の検証に適切ではないと判断した．そこで今年度は 20 桁から 50 桁の合成数を 5 桁刻みで各 5 回ずつ入力し，各桁の処理時間のトータルタイムの平均時間を比較した. しかし一回プログラムを検証するのに 3 時間かかるものもある上に学内でしか検証ができないことから統計的には検証回数が少なく信憑性の低い結果になった．検証環境については学内のサーバーを利用し，以下の通りである．

検証環境

コンパイラ: icc 14.0.1
OpenMP : 3.1
GMP : 6.0.0
CPU : Intel Xeon Phi 5110P(60 コア)
RAM : 64GB（DDR3L-1600 8GB DIMM × 8）

（※文責: 橋本和典）

4.1.3 検証結果

検証結果のまとめは以下の表のようになった.

桁数	今年度 (秒)	昨年度（秒）	平均改善率
20	3.400〜3.696	1.689〜2.257	-82%
25	10.906〜11.323	8.528〜9.513	-25%
30	47.165〜51.537	44.223〜47.599	-4%
35	177.932〜190.253	191.348〜200.441	+6%
40	686.793〜693.397	711.594〜713.633	+4%
45	2682.470〜2745.112	2653.112〜2667.896	+0%
50	10173.577〜10622.231	11763.204〜11849.030	+12%

20 桁から 30 桁では今年度のプログラムの平均改善率は昨年度より下回る結果になったが 35 桁以降は徐々に上がっていき最大 15 %も改善し処理速度が速くなった．上述の通り，本プロジェクトの目的である巨大な桁数の合成数を素因数分解するのに昨年度のプログラムより改善した. したがって素因数発見確率を上げたことによって処理速度を向上させたことを証明した.

（※文責: 橋本和典）

4.2 プログラミング班

プログラミング班は前期の活動に引き続き，ECM プログラムの改善を行った.

（※文責: 源啓多）

4.2.1 Atkin-Morain ECPP の実装

後期にまず行ったのは Atkin-Morain ECPP の実装だ．このアルゴリズムは前期中に理論班によって提示されたものだ．詳細なアルゴリズムは 3.2.2 に記述した.

（※文責: 源啓多）

4.2.2 スカラー倍算の高速化

Atkin-Morain ECPP を実装後，私たちは楕円曲線上の点のスカラー倍算の高速化に取り組んだ．スカラー倍算は ECM のアルゴリズムの中で最も計算量の多い箇所の為，高速化が期待できると考えたからだ．そのために，移動窓法（sliding/moving window method）を実装した．移動窓法はバイナリ法（binary method）や m 進展開法（window method），符号付き m 進展開窓法などと比較される楕円曲線演算の高速化手法だ．移動窓法の理解に際し，バイナリ法及び m 進展開法について学んだので，まずはこれらについて記述する．

（※文責: 源啓多）

バイナリ法

バイナリ法は楕円曲線上の点 P を k 倍した点 kP を求める際に，k を 2 進展開することで，高速なスカラー倍算を実現する手法である．昨年度から引き継いだプログラムでは，このアルゴリズムが採用されていた．バイナリ法の詳細なアルゴリズムを以下に示す．

Algorithm 6 binary method

Require: P, nbit integer $k = \Sigma_{i=0\to n-1} k_j 2^j, k_j \in \{0, 1\}$
Ensure: $Q = kP$
  $P_1 \leftarrow P$
  $Q \leftarrow 0$
  for $j = d - 1$ to 0 by -1 **do**
    $Q \leftarrow 2Q$
    $Q \leftarrow Q + P_{k_j}$
  end for

バイナリ法を用いなかった場合，kP を計算する手順は $P + P + P + \cdots + P$ であり，これには加算が $k - 1$ 回必要である．対して，バイナリ法を用いた場合は加算と 2 倍算がそれぞれ回で済む．したがって，バイナリ法を採用することで大きい k に対して高速にスカラー倍算を行うことができる．

（※文責: 源啓多）

m 進展開法

m 進展開法では，バイナリ法を応用して，2 進展開ではなく m 進展開を行っている．m 進展開の容易さから，m は $2^r (r \in Z, r > Z)$ のような値であることが多い．事前計算として $2P, 3P, \cdots, (m-1)P$ を計算する必要があるが，であるとき，r ビット単位で計算を行うことができるので，高速化につながる．以下は，m 進展開法のアルゴリズムである．

Algorithm 7 window method

Require: $P, k = \Sigma_{i=0\to d-1} k_j m^j, k_j\{0, 1, ..., m-1\}$

Ensure: $Q = kP$

$P_1 \leftarrow P$
for $i = 2$ to $m-1$ **do**
$P_i \leftarrow P_{i-1} + P$
end for
$Q \leftarrow 0$
for $j = d-1$ to 0 by -1 **do**
$Q \leftarrow mQ$
$Q \leftarrow Q + P_{k_j}$
end for

（※文責: 池野竜將）

移動窓法

m 進展開法をさらに発展させたのが移動窓法である．移動窓法では，計算を行う単位を r ビットに固定しておらず，末尾のビットが 1 かつ r ビット以下で最長になるような単位で計算を行う．末尾のビットが 1 であるようにすることで，事前計算の量が m 進展開法に比べて半分で済むことが移動窓法の特徴である．移動窓法のアルゴリズムを以下に示す．

Algorithm 8 moving/sliding window method

Require: $P, k = \Sigma_{i=0\to n-1} k_j 2^J, k_j \in \{0, 1\}$

Ensure: $Q = kP$

$P_1 \leftarrow P$
$P_2 \leftarrow 2P$
for $i = 1$ to $2^{r-1} - 1$ **do**
$P_{2i+1} \leftarrow P_{2i-1} + P_2$
end for
$j \leftarrow n-1$
$Q \leftarrow 0$
while $j \geq 0$ **do**
if $k_j = 0$ **then**
$Q \leftarrow 2Q$
$j \leftarrow j-1$
else
$t = \min\{j - t + 1 \leq r AND k_t = 1\}$
$h_j \leftarrow (k_j, k_{j-1}, \cdots, k_t)_2$
$Q \leftarrow [2^{j-t+1}]Q + P_{h_j}$
$j \leftarrow t-1$
end if
end while

他に採用するアルゴリズムとして，移動窓法の実装後に Montgomery ladder が挙がったが，実装・比較する期間を設けられなかったので来年度への課題とする．

（※文責: 池野竜將）

4.2.3 Stage2

前述したようなアルゴリズムを調査した結果，現状のアルゴリズムを改善するより新たなアルゴリズムを導入することが良いと考え，後期では Stage2 というアルゴリズムを実装した．まず，Stage2 の前提として，Stage1 を説明する．Stage1 は，P の k 倍，すなわち kP を計算する過程である．具体的には，以下のような手順で k を決定し，計算を行っている．

Stage1

$$k = \prod_{2 \leq p \leq B_1, p \in \mathbb{P}} p^{\lfloor \log_p B_1 \rfloor}$$

前述の式で求めた k を利用して kP を計算し，kP の x 座標と合成数の最大公約数をとる．その結果最大公約数が 1 でなければ素因数分解が成功したことになる．しかし，Stage1 だけでは，B_1 より大きい素数を 1 つだけ kP にかけていれば，素因数が求まった，ということが起こりうる．Stage2 は，これの頻度を減らすためのアルゴリズムである．Stage2 のための新たなパラメータ $B_2 (> B_1)$ を設定し，B_1 より大きく，B_2 以下の素数 p' それぞれを Stage1 の計算結果 kP にかけ，それぞれの $p'(kP)$ の x 座標と合成数 N の最大公約数を計算する．もし計算結果が 1 でなければ，素因数が求められたことになる．

（※文責: 千葉大樹）

基本的な **Stage2** の実装

Algorithm 9 Basic ECM Algorithm

Require: N is composite number, E is elliptic curve, $P = (x_0, y_0, Z_0) \in E(Z_n)$ is initial point, B_1 is smoothness bound for Phase 1, B_2 is smoothness bound for Phase 2, $B_2 \geq B_1$.

Ensure: q is factor of N, $1 \leq q \leq N$, or FAIL.

Phase 1.

$k \leftarrow \prod_{p \leq B_1} p^{\log_p B_1}$

$Q_0 \leftarrow kP_0$

$q \leftarrow \gcd(z_{Q_0}, N)$

if $q \geq 1$ **then**

  return q

else

  go to Phase 2

end if

Phase 2.

$d \leftarrow 1$

for each prime $p = B_1$ **to** B_2 **do**

  $(x_{pQ_0}, y_{pQ_0}, z_{pQ_0}) \leftarrow pQ_0$

  $d \leftarrow d * Z_{pQ_0} (mod N)$

end for

$q \leftarrow \gcd(d, N)$

if $q \geq 1$ **then**

  return q

else

  return FAIL

end if

（※文責: 源啓多）

4.2.4 開発環境と実行環境

プログラムを実行する環境について記述する．前期までは，Xeon Phi 5110P を用いてプログラムを実行していたが，Xeon E5-2640 で実行した方がプログラムの実行速度が速かったため，Xeon E5-2640 を主に使用することにし，Xeon Phi 5110P は検証用に使用することにした．メインメモリについては，64GB（DDR3L-1600 8GB DIMM × 8）を搭載している．コンパイラは icc 14.0.1，OpenMP のバージョンは 3.1，GMP は 6.0.0 を使用している．コンパイル時には，最適化オプション（-O2）を指定している．

（※文責: 橋本和典）

4.2.5 GMP について

ECM を C 言語で実装するにあたり，符号なし 64bit の数値型である unsigned long long を用いてもオーバーフローすることは避けられない．そのため，多倍長計算を高速に行うライブラリである GMP を使用した．GMP で使用した関数について説明する．

void mpz_init(mpz_t x)
mpz_init() は mpz_t 型の変数 x を初期化するための関数である．

void mpz_inits(mpz_t x, …)
mpz_inits() は複数の mpz_t 型の変数を初期化するための関数である．引数として与えられた変数のリストは，NULL で終端しなければならない．

void mpz_clear(mpz_t x)
mpz_clear() は mpz_t 型の変数 x が確保していたメモリを開放するための関数である．

void mpz_clears(mpz_t x, …)
mpz_clears() は複数の mpz_t 型の変数が確保していたメモリを開放するための関数である．引数として与えられた変数のリストは，NULL で終端しなければならない．

void mpz_set(mpz_t rop, const mpz_t op)
mpz_set() は mpz_t 型の変数の代入を行う関数である．mpz_set_ui()，mpz_set_si()，などのバージョンがあり，それぞれ第 2 引数が unsigned long int，signed long int になっている．mpz_set_str() という関数もあり，第 2 引数が const char*になっており，第 3 引数には基数を指定する．そして，第 2 引数に指定した文字列を第 3 引数に指定した基数に基づき，整数だと解釈して第 1 引数に指定した mpz_t 型の変数に代入する．

void mpz_add(mpz_t rop, const mpz_t op1, const mpz_t op2)
op1+op2 を rop に代入する．第 3 引数の型が unsigned long int である mpz_add_ui も存在する．

void mpz_sub(mpz_t rop, const mpz_t op1, const mpz_t op2)
op1-op2 を rop に代入する．第 3 引数の型が signed long int である mpz_sub_si(), unsigned long int である mpz_sub_ui() も存在する．

void mpz_mul(mpz_t rop, const mpz_t op1, const mpz_t op2)
op1*op2 を rop に代入する．第 3 引数の型が signed long int である mpz_mul_si()，unsigned long int である mpz_mul_ui() も存在する．

void mpz_pow_ui(mpz_t rop, const mpz_t base, unsigned long int exp)
base を exp 乗し，rop に代入する．

void mpz_mod(mpz_t r, const mpz_t n, const mpz_t d)
n mod d を r に代入する．

（※文責: 源啓多）

4.3 広報班

広報班では，プロジェクト活動や楕円曲線法の解説を外部に発信することにした．発信をする方法として，FUN-ECM の WEB ページを制作することにした．WEB サイトを制作するにあたっ

て後期の活動から広報班を新たに結成し，活動を行った．発信する対象は主に情報系の大学生とし，学部一年生の知識でも理解できるように楕円曲線法とその周辺の理論的な基礎知識を表記した．また，来年度以降の活動や，専門知識をもった人にむけて今年得た知識やプログラムで変更した点など，今年度の専門的な活動内容も表記した．

（※文責: 駒ヶ嶺壮）

4.3.1 動機

FUN-ECM は今年で 3 年目であるが，毎年 ECM のプログラムの改良を重ねる活動であるため毎年外部への露出が少なく，継続性の強いプロジェクトであることから私たちは今年度ならではの対外的な新規活動をしたいと考えた．新規的な活動をするにあたり，未来大生でも ECM について知らない方が多いことからより多くの人に ECM を伝えられるように広報的活動を行うことにした．また，中間発表での意見でプレゼンでは理解しにくかったとの意見を頂いたことからよりわかりやすく伝えられ，手軽にみることができる媒体として web ページを採用した．

（※文責: 駒ヶ嶺壮）

4.3.2 Web ページの構成と内容

Top

本サイトのトップページ．SNS の共有ボタン，活動写真のスライダーや FUN-ECM とは何かに関する簡単な説明がある．また，ここでは来年度の活動のためのアンケートページも設置した．一番下にあるボタンから各ページに飛ぶことができる．

（※文責: 駒ヶ嶺壮）

About

広報班が活動目的としていた内容のコンテンツである．ECM の基礎理論の説明ページ，FUN-ECM の活動目的，今年度の ECM プログラムの解説ページの 3 つに分かれている．章ごとに分けて詳細的な説明を行い，段階的に読み進めることができるようにした．また，重要な箇所での色の変更や gif 画像を挿入するなどしてより理解しやすいように工夫した．ECM プログラムの解説ページでは今年度のプログラムのソースコードの一部を掲載し，すべてのソースコードについては URL から github のページに飛ぶことができるようにした．

（※文責: 駒ヶ嶺壮）

History

2016 年度の活動月表を掲載した．前期活動と後期活動の 2 つに分けて，後期活動の中には 2016 年度の全体成果についても記載した．どの班がどのような活動をしたかについて簡単にわかるようにした．

（※文責: 駒ヶ嶺壮）

Link
リンク集．本プロジェクトで使用した ECM-NET や本学のホームページ等を掲載した．

（※文責: 駒ヶ嶺壮）

4.3.3 Web ページ内のファイルの説明

bootstrap.css
Web サイトや web アプリケーションを作成するための web アプリケーションフレームワークである．Class 属性を指定するだけで簡単に豊富なスタイルを指定することができるファイル．

bootstrap.min.css
bootstrap.css の圧縮版であるファイル．読み込みを早くしたい時などは bootstrap.css ではなくこちらを使用する．

jquery.bxslider.css
スライドショー形式で表示された画像のスタイルを指定しているファイル．

jsxgraph.css
javascript を用いた楕円曲線のグラフのスタイルをしているファイル．

original.css
上記以外のスタイルを指定している．レイアウトのために私たちが設定した

bootstrap.js
bootstrap の javascript の部分を動かすためのファイル．後述する jQuery のファイルを先に読み込まなければ機能しない．

bootstrap.min.js
bootstrap.js の圧縮版であるファイル．読み込みを早くしたい時などは bootstrap.js ではなくこちらを使用する．

jquery.bxslider.min.js
画像をスライドショー形式で表示するための関数を組み込んでいるファイルである．

jquery.js
jQuery を WEB ページ内に組み込むためのファイル．jQuery とは java script をより扱いやすくしたファイルであり，本来であれば複雑なプログラムの記述も jQuery を用いることで簡易的に記述することができる．

jquery.min.js

jquery.js の圧縮版であるファイル．読み込みを早くしたい時などは jquery.js ではなくこちらを使用する．

jsxgraphcore.js，GeonextReader.js

web ページ内のグラフを動かすための javascript が記述されたファイル．

ecm.html

FUN-ECM ウェブサイトの”ECM とは”について書かれている html ファイルである．「ECM とは何か」について，基礎知識として $modN$ の説明や点同士に置ける加算と 2 倍算についての説明が記述されている．

図 4.1　ecm.html

ecm1.html

FUN-ECM ウェブサイトの”活動目的”について書かれている html ファイルである．FUN-ECM がなぜ素因数分解をするのか，また ECM-NET とは何かについて記述されている．

図 4.2　ecm1.html

index.html

FUN-ECM ウェブサイトの”トップページ”について書かれている html ファイルである．FUN-ECM の活動風景の写真や名前の由来，各ページへのリンクが記述されている．

図 4.3　index.html

link.html

FUN-ECM ウェブサイトの”リンク”について書かれている html ファイルである．ECM-NET や STUDIO KAMADA など，ECM に関係するサイトや ECM についての説明がされているサイト，また FUN-ECM の教授へのリンクも記述されている．

図 4.4　link.html

log.html

FUN-ECM ウェブサイトの”前期活動”について書かれている html ファイルである．4 月から 8 月までの FUN-ECM の活動記録が記述されている．

図 4.5　log.html

log2.html

FUN-ECM ウェブサイトの”後期活動”について書かれている html ファイルである．9 月から 1 月までの FUN-ECM の活動記録が記述されている．

図 4.6　log2.html

member.html

FUN-ECM ウェブサイトの”メンバー紹介”について書かれている html ファイルである．メンバー全員の名前と班，一言が記述されている

図 4.7　member.html

program.html

FUN-ECM ウェブサイトの”プログラムについて”について書かれている html ファイルである．FUNECM プログラムの使い方や ECM の基本的な実装，さらにその他の ECM に関する専門的な知識やそのプログラムが記述されている．

図 4.8　program.html

（※文責: 山下哲平）

4.3.4　展望

私たちは当初，情報大学生の学部 1 年生でも私たちの活動を理解することのできるような解説ページを設けるという一つの目標を立て，実際に ECM についての大まかな流れを難しいと思われる部分を噛み砕きつつ解説したページを作成した．そして最終発表会において私たちの発表を見てもらった方にウェブページにあるアンケートで感想を答えてもらった．しかし，アンケートに答えてもらった方の母数が少なかったため，来年度はアンケートに答えてもらう人数を今年度より増やすことによって正確な理解度の調査を行い，その結果を使いウェブページの改善を行いたい．

（※文責: 駒ヶ嶺壮）

4.4 成果発表

成果発表会では，前期に行った中間発表のレビューを元に改善をした．レビューでは，内容が理解できた人とできていない人が分かれていたため，さらに前提知識のない聴衆にも伝わるような内容を目指した．

（※文責: 千葉大樹）

4.4.1 準備

ポスター

後期の活動では，3 つの活動を並行して行っていたため，ポスターを前期に比べて 1 枚増やし 3 枚で構成を考えた．次にメインポスターとサブポスターを分け，メインは全体の活動を大まかに伝える，サブポスターはそれぞれの活動を具体的に伝えるという目標を設定し，各自作成をした．ポスターの作成には前年度に引き続き「Microsoft PowerPoint」というソフトウェアを利用した．ポスターが完成次第，グループごとにレビューを行い，誤字脱字を修正した．

（※文責: 千葉大樹）

プレゼンテーション資料

前期の中間発表のレビューでは，伝わった人と伝わらなかった人が分かれていたため，前提知識が殆どない聴衆でもわかりやすくなるように，専門的な用語を最小限にするように注意し，最も大事でなところを枠で囲み，その中身を見るだけで大まかな内容を理解できるようにした．また，楕円曲線法についての説明では，例示を多く含めることで数学に抵抗のある人でも触れやすいようにした．加えて，長い数式に関しては説明を省きスライドに表示するだけにとどめ，数学が苦手だという方の抵抗を減らすようにした．作成には前年度に引き続き「Microsoft PowerPoint」というソフトウェアを利用した．

（※文責: 千葉大樹）

4.4.2 発表

発表は，前後半で 4 人ずつに分かれて行った．前半は 3 人がそれぞれの担当について発表を行い，1 人がポスターの前で質問を受けたり，評価シートを配るという配役で行った．後半は 4 人がそれぞれの担当について発表を行い，発表を行っていない 1 人が他の作業を行った．前期の発表中にプロジェクターの電源が落ちてしまうアクシデントがあった為，そのようなアクシデントに対応するために，PC でプレゼンテーションを行いながら復旧作業を行うことを決めた．発表後に評価アンケートの集計を行った結果，発表技術は 10 点中平均 7.1 点，発表内容は 10 点中 7.9 点だった．共に前期の評価よりも点数が上昇しており，発表の工夫の効果が表れていることが確認できた．しかし，いくつかのコメントに内容を省きすぎている，数式の説明をしないのはよくない，などの意見もあった．

（※文責: 千葉大樹）

第 5 章　プロジェクト内のインターワーキング

- 池野竜將（プロジェクトリーダー・プログラミング班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 大まかな作業スケジュールを作成し，進捗管理を行った.
  (3) 源と協力して前年度の ECM プログラムを理解した.
  (4) 源のコーディング作業にアドバイスをした.
  (5) 理論班からのプログラミング班に関しての質問に回答し，必要があれば聞かれた内容を源に伝えた.
  (6) 中間発表会に向けて，プレゼンテーション資料・原稿の原案を作成した.
  (7) 中間発表会に向けて，プログラミング班の部分のプレゼンテーション資料を作成した.
  (8) 成果発表会に向けて，プログラミング班の部分のプレゼンテーション資料を作成した.
  (9) プロジェクト報告書の作成を行った.
- 源啓多（プログラミング班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 池野と協力して前年度の ECM プログラムを理解した.
  (3) ECM プログラムのバージョン管理の為，Git を学んだ.
  (4) 前年度の ECM プログラムの実装上のミス（3.3.1）を改善した.
  (5) ECM プログラム改善のために，新たなアルゴリズム（3.3.2, 3.3.3）の実装を行った.
  (6) 中間発表会に向けて，プログラミング班のプレゼンテーション資料・原稿を作成した.
  (7) Stage2 の解読・実装をいち早く進めた.
  (8) 解析班の作業を助けるためのマクロを作成した.
  (9) 広報班に協力し，ウェブページの作成の手助けをした.
  (10) 成果発表会に向けて，プログラミング班の部分のプレゼンテーション資料の修正を行った.
  (11) 最終報告書作成の際，各グループの作成物のレビューを行った.
- 山下哲平（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 伊藤・駒ヶ嶺と協力して Edwards Curve を利用した ECM アルゴリズムの読解を行い，プログラミング班に提案を行った.
  (3) 伊藤・駒ヶ嶺と協力して Atkin-Morain ECPP アルゴリズムの理解に取り組んだ.
  (4) 中間発表会に向けて，「背景」の部分についてポスターをを作成した.
  (5) プログラミング班の要請でプログラムの速度について簡易的な検証を行った.
  (6) 伊藤と協力してウェブページの基本的な要素を作成した.
  (7) 最終報告書の広報班ページを作成する際に，中心なって活動し進捗を管理した.
- 伊藤有輝（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 駒ヶ嶺と協力して，エドワーズ曲線の式が導き出される過程を学んだ.
  (3) 駒ヶ嶺･山下と協力し，Edwards Curve を利用した ECM アルゴリズムの読解を行った.

(4) 駒ヶ嶺・山下と協力し，Atkin-Morain ECPP アルゴリズムの理解に取り組み，プログラミング班に提案を行った．
(5) 中間発表会に向けて，「理論班」の部分のプレゼンテーション資料を作成した．
(6) 源・池野と協力し，Github の使い方を理解して広報班に伝えた．
(7) 成果発表会に向けて，理論班の部分のプレゼンテーション資料の修正を行った．
(8) 山下と協力してウェブページの基本的な要素を作成した．

- 駒ヶ嶺壮（理論班）
  (1) 楕円曲線法の基礎を学んだ．
  (2) 伊藤と協力して，エドワーズ曲線の式が導き出される過程を学んだ．
  (3) 山下・伊藤と協力して Edwards Curve を利用した ECM アルゴリズムの読解を行った．
  (4) 山下・伊藤と協力して Atkin-Morain ECPP アルゴリズムの理解に取り組んだ．
  (5) 中間発表会に向けて，「理論班」の部分のポスターを作成した．
  (6) 広報班のウェブページ作成のため，過去の作業ログを見直し，まとめた．
  (7) 中間報告書の間違いを修正し，新たにセクションを追加した．
- 橋本和典（理論班）
  (1) 楕円曲線法の基礎を学んだ．
  (2) 千葉・亀谷と協力して入門書を読み，基礎学習を行った．
  (3) 亀谷と協力して基礎学習を簡潔にまとめた解説ノートを作成した．
  (4) 中間発表会に向けて，千葉・亀谷と協力して来るであろう質問を予測して対策を行った．
  (5) ECM の改善に直結するような文献を探した．
  (6) 中間発表会に向けて，ポスターの「理論班」の章を英訳した．
  (7) 理論班で検証を行う際に，管理者として中心となって作業した．
  (8) 行った検証の結果をまとめ，グラフ化して見やすくした．
- 千葉大樹（理論班）
  (1) 楕円曲線法の基礎を学んだ．
  (2) 亀谷・橋本と協力して入門書を読み，基礎学習を行った．
  (3) 中間発表会に向けて，ECM についての英論文から重要な単語を抜粋し解説した．
  (4) 中間発表会に向けて，亀谷・橋本と協力して来るであろう質問を予測して対策を行った．
  (5) 中間発表会に向けて，ポスターの「プログラミング班」の章を英訳した．
  (6) 理論班で検証を行う際に，実際にプログラムを動かし，データを全体に共有した．
- 亀谷浩也（理論班）
  (1) 楕円曲線法の基礎を学んだ．
  (2) 橋本・千葉と協力して入門書を読み，基礎学習を行った．
  (3) 橋本と協力して，基礎学習を簡潔にまとめた解説ノートを作成した．
  (4) 中間発表会に向けて，橋本・千葉と協力して来るであろう質問を予測して対策を行った．
  (5) 中間発表会に向けて，ポスターの「概要・基礎学習」の章を英訳した．
  (6) 理論班で検証を行う際に，データの管理やまとめを手伝い，橋本の補佐として活動した．
  (7) 最終報告書の広報班ページを作成する際に，中心なって活動し進捗を管理した．

（※文責: 池野竜將）

第 6 章　前期活動成果

本プロジェクトでは，理論班で理解することに成功した高速化アルゴリズムをプログラミング班に伝え，プログラミング班がそのアルゴリズムを実装することにより ECM プログラムを作成した．

（※文責: 千葉大樹）

6.1　理論班

理論班は，活動内容で示した射影座標を用いたスカラー倍算楕円曲線プログラムにおける変数の点の与え方のアルゴリズムを発見した．これにより乗算の回数，除算の回数が減少したことにより素因数を発見する効率を理論上 1.5 倍に上昇させることができる．しかし，実装前との計算コストの実数値の比較をすることはできなかった．また，Atkin-Moraine ECPP アルゴリズムの理解に成功した．このアルゴリズムを実装することにより，位数があらかじめ小さな因数 d を持つ曲線のみを使用し，素因数を p とした場合，ランダムに動くサイズが p から p/d に減少するため因数分解に成功する確率を高めることができる [6]．しかし，まだ実装には至っていないため，前期の活動はアルゴリズムの読解についてまとめたレポートを作成し，論文の読解を終了した．

（※文責: 駒ヶ嶺壮）

6.2　プログラミング班

プログラミング班では，新たなアルゴリズムを実装し，理論上は 6.1 のように計算量が減少することが分かった．詳細な実験は行っておらず有意な差があるかどうかは確認できていない．だが，実際に素因数分解を行った結果，処理が早くなっていることが確認できた．

また，実際に巨大な合成数を分解し，昨年度のプログラムとの性能を比較することにした．評価するにあたって，2015 年度に作成されたプログラムでテストに使用されていた合成数 $10^{306}+1$ を素因数分解することで，以前のプログラムとの比較をすることとした．2015 年度のプログラムでこの合成数を分解した結果，発見されたもっとも大きな素因数は 1575389803198166 07(21 桁) であった．同様に今年改善されたプログラムで分解した結果，発見されたもっとも大きな素因数は 112544281755782732673671367061(30 桁) であり，より大きな素因数を見つけることができるよ

表 6.1　昨年度と今年度のプログラムの計算コストの比較

	2 倍算	2 倍算→加算
昨年度	3**M**+4**S**+1**D**[*1]	13**M**+5**S**+3**D**
今年度	3**M**+4**S**+1**D**	12**M**+4**S**+1**D**

[*1] **M**:乗算，**S**:2 乗算，**D**:楕円曲線の係数 a,d を用いた乗算

うに改善された．素因数が見つかった際のログを下に示す．

30 桁発見の際のログ

Stage1: d = 126909574787277066813799168682279610402560329810862501906041

Stage1 time: 5.359253 seconds

Stage2 time: ——

total time: 5.359 seconds

Y=467302486668311950652368367858681144797779577435280581316 79

———————————————

@ probable prime factor found: 184736584265492707905284574931 digits: 30 cofactor: 738759178437819643189478148923

（※文責: 源啓多）

第 7 章　後期活動成果

7.1　理論班

検証するにあたり，まず今年度はどのような方法で昨年度のプログラムと比較するか考えた．昨年度の報告書を参考にし，昨年度では素数を入力しアルゴリズムが終了するまで時間を計測し比較していたが，今年度はプログラムの処理速度の改善ではなく素因数を発見する確率を上げたため今回はこの方法では検証しなかった．おそして，白勢先生にアドバイスをいただき，そのアドバイスに基づいて，検証を行った．統計に関しては検証に時間がかかり検証回数が少なかったので統計としてはデータが少なく信頼性が低いが，数値として結果を表すことができた．

（※文責: 駒ヶ嶺壮）

7.2　プログラミング班

後期の活動では，新たに 3 つのアルゴリズムを導入した．改善率などの具体的な数値に関しては 4.1 に示しているので省略する．また昨年度までは実装されていなかった Stage2 を新たに適用した．理論班の検証結果によると ECM プログラムを最大で 15% ほど高速化することに成功した．また，広報班と協力し ECM プログラムの使用法やアルゴリズムについて解説した．また，前期に引き続き，巨大な合成数の分解を行った．後期の間に発見されたもっとも大きな素因数は 95371895138956317843189468149739281（35 桁）であり，前期に比べてより大きな素因数を見つけることができるように改善された．素因数が見つかった際のログを以下に示す．

35 桁発見の際のログ

Stage1: d = 57530732891203017734290530349497316581741957102926595699495436568048 9
Stage1 time: 3.971019 seconds
Stage2 time: —
total time: 3.971 seconds
Y=208593389463818851094026758572553177387116277220987942007066385150361 0

@ probable prime factor found: 95371895138956317843189468149739281 digits: 35 cofactor: 219748319567365238928091473625832 87

（※文責: 駒ヶ嶺壮）

7.3　広報班

後期の半期間で我々の活動内容や ECM の基礎理論について紹介する web ページを作成した．ページ自体は期間内に完成させることが出来たが，2 か月半という短い時間での制作だったため最終発表までに，作成したページに関するアンケートをとることが出来なかった．そのため，ページ

のコンテンツ力が高いか否かについての統計をとることが出来ず残念ながら効果の測定を十分に行うことができなかった．十分な測定とは言えないが，最終発表内の時間を用いてページを紹介し，有志でアンケートを依頼した結果若干名ではあるが回答してくれた．回答してくれた方の約半数がECM の基礎理論について理解できたと回答してくれた．

（※文責: 駒ヶ嶺壮）

第 8 章　まとめ

8.1　前期活動結果

前期は参考資料，論文，担当教員の白勢先生の講義による楕円曲線法の理解から始め，楕円曲線が楕円曲線法においていつどのように使われるかを理解した．その後，理論班，プロジェクト班の 2 班に分かれ作業を行った．理論班は，論文，入門書の読解をし，プログラム高速化のための改善案を出すことに成功した．しかし，前期中にプログラミング班が実装することはできなかった．プログラミング班は前年度のプロジェクトで作成された ECM プログラムを理解した．その後，実装ミスの改善や，新たなアルゴリズムの実装を行い，計算コストの減少に成功した．

（※文責: 千葉大樹）

8.2　後期の展望

後期は，理論班が作成した Atkin-Morain ECPP アルゴリズムを実装し，さらに ECM プログラムの改善を図る．また，大きな合成数の分解を続け ECMNET へのランクインを目指す．加えて，前期中に活動できなかった広報について新たに班を設置し活動していく．

（※文責: 橋本和典）

8.3　後期活動結果

後期はプログラミング，理論，広報にわかれ作業を始めた．プログラミング班は初めに，前期中に理論班によって提案された Atkin-Morain ECPP の実装を行った．その後は，スカラー倍算の高速化アルゴリズムを実装し，プログラムの高速化に成功した．また新たな試みとして今まで実装されていなかった Stage2 の実装を行ったが，効率はほとんど変わらなかった．理論班は完成したプログラムを検証するために，検証方法の調査・提案を行った．その後，自分たちで提案した検証方法を元に検証を行い，結果をまとめた．広報班では，最初にウェブページのコンテンツやターゲットについての提案を行った．その結果，メインターゲットを未来大を中心とする情報系の大学生とし，ECM についての基礎理論についてのページを作成することにした．またサブターゲットとして，来年のプロジェクトメンバー向けに今年度作成したプログラムについてのページを作成することにした．完成したウェブページは gh-pages というサービスを利用して公開した．

（※文責: 池野竜將）

8.4　全体を通して

ECM を利用した素因数分解プログラムは，検証の結果分解する合成数の桁数が大きければ大きいほど改善しており，最大で 15% ほど高速化されている．しかし，発見できた合成数は

95371895138956317843189468149739281(35) が最大であり，ECMNET へのランクインには最低でも 64 桁以上の素因数を発見する必要があるため，現状では ECMNET へのランクインは難しい．また，今年度の新しい活動として行った FUN-ECM の広報活動は，アンケートによると理解出来た人とできない人に分かれたが，アンケートの解答数が少なく，評価はできなかった．

（※文責: 駒ヶ嶺壮）

8.5 今後の課題と展望

- 今までの試行で発見できた合成数は 30 桁が最大であり，ECMNET へのランクインは難しいと予想される．しかし，ECM は運要素の強いアルゴリズムの為，どのような原因で素因数が発見できていないかを理解していない．そのため，来年度は既に分解されている合成数の分解を並行して行い，その時点のプログラムでどれくらいの桁数の素因数を発見ができるかについても確認・検証が必要だと考えられる．
- 広報作業開始が後期であったため，作成後に評価をするための時間が十分に取れなかった．そのため，来年度も広報活動を行うのであれば，前期から活動を開始し，学生からのフィードバックでを受ける時間を確保することが必要だと考えられる．

（※文責: 池野竜將）

付録 A　新規習得技術

- PARI/GP の使用
- Microsoft PowerPoint の使用
- Git の使用
- GitHub の使用
- Xeno Phi の使用
- functionview の使用

（※文責: 駒ヶ嶺壮）

付録 B　相互評価

ここでは，最終発表後の相互フィードバックで述べられたコメントを列挙する．

- 池野竜將
  - リーダーとして，進捗管理，タスクの振り分け，発表資料の統合などを滞りなく行っていた．
  - プロジェクトリーダーとして一年間とても頼れる存在であった．メンバーのまとめ役や，スケジュールの管理，タスクの分担などリーダーとして様々な分野で活躍していた．
  - プロジェクトのリーダーとして，1 年間プロジェクトの進行をしてくれた．日程調整や，役割分担，そのほかの作業に大きく関わってくれ，本プロジェクトのキーパーソンとして非常に活躍してくれた．
  - 1 年間プロジェクト全体の統括をしてくれた．報告書やポスター等の全体で行う作業だけでなく，各班の進捗を確認するなど細かいところまで気配りができる頼れる存在だった．
  - 一年間プロジェクトリーダーとしての活動を責任持ってしてくれた．特にメンバー全員が効率よく活動できるよう日程調節と役割分担を毎回のプロジェクト時間までに考えてくれた．
  - プロジェクトリーダーとして責任をもって活動しており活動方針や予定を明確に表してくれて作業が滞ることなく進められるようにしてくれた．
  - プロジェクトリーダーとして，プロジェクトを先導していた．スケジュールの管理，作業状況の管理など活動をまとめるための重要な仕事を行っていた．
- 源啓多
  - 1 年間通してプログラムの改善に取り組んでいた．また，プロジェクトを進行するうえで必要なツールをいち早く取り入れ使いこなしていた．人に教えるのは苦手と言いながらも他のプロジェクトメンバーに積極的に GitHub の使い方を教える等，活動の中心になっていた．
  - プログラミング班として，自分たちが見つけた atkin-moraign ECPP の実装や，その他の実装など，ECM プログラムにおいて大きく貢献していた．彼の活躍のおかげで 2016 年度 FUN-ECM はより大きな素因数を発見することができた．
  - 1 年を通し，本プロジェクトの目的であるプログラムの改善を行っていた．その他にも，ウェブページ作成の際に使い慣れていないツールの使用方法を他メンバーに使い方を教えてあげていたりなど，他のメンバーの作業の手助けも積極的に行っていた．
  - プログミング班の中心として ECM プログラムの改善に大きく関わっていた．また，Github や bootstrap 等の導入に関して補助してくれたり，広報班の活動でも大きく貢献してくれた．
  - 一年間プログラムの製作に没頭してくれた．後期はプログラムだけではなく，検証班の検証方法についてもいろいろ教えてくれた．
  - 1 年間のプロジェクトを通してプログラムに積極的に関わっていた．またほかの班にもプログラム使い方などを教えてくれたりしていた．

  - プログラマとして 1 年間プログラムの改善に努めていた．また，検証を行う上で必要な情報について教えてもらった．
- 山下哲平
  - 広報の作業では，インターンで学んだウェブページ作成の技術を生かし，伊藤・駒ヶ嶺と協力してスムーズに広報作業を進行させていた．前期に引き続きムードメーカー的な要素も持ちつつ，みんなが嫌がるような英訳などの地道な仕事も引き受け多方面からプロジェクトを支えていた．
  - ムードメーカーの役割を担いつつ，伊藤・駒ヶ嶺と Web サイトのデザインの改善・コンテンツの制作を積極的に行っていた．
  - 後期はウェブページ班として，活躍していた．もともとウェブデザインに興味を持っていたらしく，専門的な知識も周りのメンバーよりも多く持っていたため，ウェブページ作成の際にその知識を活用していた．
  - 後期では広報班の活動で私と協力して作業をしてくれた．共に web ページ制作に関して勉強し，スキルを高め合った．
  - 後期では FUNECM で初めての活動内容である広報班のメンバーとして web ページ製作を進めてくれた．
  - 後期からはウェブページの作成に取り組んでいた．またプロジェクトの場を盛り上げるなどし活動の雰囲気をよくしてくれていた．
  - 後期では，広報班としてウェブページの構成，コンテンツの作成を行っていた．ウェブページデザインについては前々からよく学んでいたらしく，ページ作成において重要なポジションを取っていた．
- 伊藤有輝
  - 広報の作業では，インターンで学んだウェブページ作成の技術を生かし，山下・駒ヶ嶺と協力してスムーズに広報作業を進行させていた．一度ウェブページが完成した後も，よくないと感じたところをすぐに提案し，自ら手直しを加える等積極的にブラッシュアップを行い，よりよいものを作成していた．
  - インターンで学んだ技術を活かし，山下・駒ヶ嶺と Web サイトのデザインの改善・コンテンツの制作を積極的に行っていた．
  - web ページ作成において中心となって活動していた．自分とともに最初は試行錯誤していたが，最終的にスキルを高めあうことができたたまにトイレに行っていた．
  - 後期はウェブページ班として活動を行っていた．もともと数学が好き，ということで，論文の読解の際，積極的に取り組んでいた．後期の活動の際もウェブページ作成の要として活動に積極的に取り組んでいた．
  - 広報班で山下，駒ヶ嶺と共に FUNECM 活動発信してくれた．また発表前ではプレゼンの改善点をいろいろな方向から発言してくれた．
  - 広報班でウェブページの作成におり，ウェブページ以外にも発表資料の作成などにも活躍していた．
  - 広報班として活動を行い，ウェブページの作成を先導していた．プレゼン資料の作成も積極的に行っていた．
- 駒ヶ嶺壮
  - 後期から山下・伊藤とともに広報の作業を行っていた．あまり（ウェブページを作成する）技術はないと言っていたが，だからと言って受け身になるわけではなく，アイデア

出しから紹介ページの作成など幅広く活動し，サポート役のような形で積極的に作業に参加していた．

- 伊藤・山下と Web サイトのデザインの改善・コンテンツの制作を積極的に行っていた．
- 伊藤，山下とともに web ページ作成において貢献した．また，前期も理論班として論文の解読において大きな貢献をしていた．発表や報告書作成においても大きな貢献をしていた．
- 広報班の活動で共に作業した．web ページ作成だけでなく，早い段階からポスターや報告書の作成に回り，チーム全体としての活動も進めてくれた．
- 広報班で山下，伊藤と共に ECM の活動発信してくれた．また，手の空いてる時は他のメンバーの協力も積極的にしてくれた．
- 広報班として駒ヶ嶺と山下とでウェブページの作成に取り組んでいた．プレゼン発表の練習にもアドバイスなどしており活躍していた．
- 広報班としてウェブページの作成を行っていた．また，プレゼン資料やポスターの作成も積極的に行い，発表の一助となった．

- 橋本和典
  - 後期では，検証を中心に行ってもらった．検証方法の調査から検証データのまとめまですべてに関わり，リーダーとして最後までやり切っていた．プログラムが改善されるたびに増える検証項目やまとめるデータ量に心を折ることなく仕事を全うしていた．
  - 亀谷・千葉と協力し，プログラムの速度がどれだけ改善されたかを検証し，まとめてくれた．
  - 理論班，検証班として前期は亀谷，千葉ととも専門書の解読，後期はプログラムの検証で活躍してくれた．また，報告書やポスター，発表の際も活躍していた．
  - 後期は前期に引き続き理論班として活動を行っていた．プログラムの検証という難しい役だったが，他のメンバーに教えてもらったり，自力で調べたりなどし，積極的に活動に参加していた．
  - 理論班の活動として作成したプログラムを検証してくれた．検証では他のメンバーを先導し，検証の仕方から一生懸命取り組んでくれた．
  - 同じ班員として協力して成果を上げることができた，検証結果などを積極的にまとめてくれていた．
  - 検証班として，データの整理やまとめを積極的に行っていた．図表等で整理されたデータから分かったこと・考えられたことも多く，考察の時に役立った．良い検証班の先導役だった．
- 亀谷浩也
  - 前期に引き続き橋本のサポート役に徹していた．渡されたデータのまとめ・可視化や，検証活動を行っていた 3 人での意見交換の際に話を進める等，円滑な活動に必要不可欠だった．また，最終発表の際は，聴衆を意識した発表を行い企業の方からも評価されていた．
  - 橋本・千葉と協力し，プログラムの速度がどれだけ改善されたかを検証し，まとめてくれた．
  - 理論班，検証班として前期は橋本，千葉とともに専門書の解読，後期はプログラムの検証で活躍してくれた．また，報告書やポスター，発表の際も活躍していた．
  - 後期は，千葉，橋本とともにプログラムの検証を行っていた．後期から始まった活動

で，最終発表のスライドやポスターなどのグラフの作成なども最初からだったが，積極的に作成を行っていた．
  - 橋本と共にプログラムの検証を行っていた．検証以外の活動ではポスターの制作や最終スライドの準備に関して積極的に活動していた．
  - 検証班として私の意見を参考にしていろいろとアドバイスをくれた．また検証結果を可視化などの工夫をしてくれた．
  - 検証班として活動．報告書やポスター・スライドの作成に積極的に携わり，良い発表が行えるように努めていた．
- 千葉大樹
  - 検証作業に参加し，プログラムのデータをとることを中心に行っていた．プログラムを動かして，データを渡すという作業は多大な時間はかかるが，めんどくさがらずに行っていた．また，プレゼン資料のブラッシュアップの際には自分の意見をはっきりと伝え，よりよい方向に導いていた．
  - 橋本・亀谷と協力し，プログラムの速度がどれだけ改善されたかを検証し，まとめてくれた．
  - 理論班，検証班として前期は亀谷，橋本とともに専門書の解読，後期はプログラムの検証で活躍してくれた．また，報告書やポスター，発表の際も活躍していた．
  - プログラムの検証班として積極的に検証を行っていた．もともと発表が苦手だったらしいのだが，一人で最終発表の練習を行っているなど，苦手を克服しようと積極的に活動していた．
  - プログラムの検証班として積極的に検証を行っていた．もともと発表が苦手だったらしいのだが，一人で最終発表の練習を行っているなど，苦手を克服しようと積極的に活動していた．プログラムの検証ではわからない点が出ると先生に聞きに行ったりなど問題を解決するために精力的に活動していた．発表練習では前日に学校に残るなどグループ全体での成果に貢献していた．
  - 一年を通して，理論的な部分で困ってた時に理解した部分を教えてくれた．苦手なプレゼンもできるように練習をしていた．
  - プログラムの検証に主に関わってくれており，個人作業が多かったが検証などに取り組んでいた．

（※文責: 亀谷浩也）

参考文献

[1] ECMNET. https://members.loria.fr/PZimmermann/ecmnet/, (最終アクセス 2017 年 1 月 16 日)

[2] Bernstein, D.J. , Birkner, P. , Lange, T. , Peters, C. ECM USING EDWARDS CURVES. Mathematics of Computation, 2013.

[3] Hisil, H., Wong, K.K.-H., Carter, G., Dawson, E. Twisted Edwards curves revisited. Advances in Cryptology - ASIACRYPT 2008, 2008.

[4] STUDIO KAMADA. http://stdkmd.com/, （最終アクセス 2017 年 1 月 16 日）.

[5] Joseph H. S., John T. 楕円曲線論入門丸善出版, 2012.

[6] 國廣 昇 , 鶴岡行雄, 小山謙二. 適切な位数を持つ楕円曲線に基づく素因数分解. SCIS, 1997.

[7] Kris Gaj, Soonhak Kwon, Patrick Baier, Paul Kohalbrenner, Hoang Le, Mohammed Khaleeluddin, Ramakrishna Bachimanchi. Implementing th Elliptic Curve Methof of Factoring in Reconfogurable Hardware. CHES-2006, 2006.

[8] 森下拓也,Jibhui Chao. 疑似的 2 次拡大環上の楕円曲線法. FIT2015, 2015.

[9] Henriette Heer, Gary McGuire, Oisin Robinson. JKL-ECM: an implemention of ECM using Hessian curves. LMS Journal of Computation and Mathmatcis, 2016.