公立はこだて未来大学 2019 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2019 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
グループ 1
Group Name
Group 1

プロジェクト番号/Project No.
17-A

プロジェクトリーダ/Project Leader
下地健介　Kensuke Shimozi

グループリーダ/Group Leader
下地健介　Kensuke Shimozi

グループメンバ/Group Member
下地健介　Kensuke Shimozi
富樫北斗　Hokuto Togashi
谷山真子　Mako Taniyama
松崎伸彦　Nobuhiko Matsuzaki

指導教員
白勢政明　由良文孝
Advisor
Masaaki Shirase　Fumitaka Yura

提出日
2020 年 1 月 22 日
Date of Submission
January 22, 2020

概要

私達のプロジェクトの目的は, 楕円曲線法 (ECM) によってより大きな桁数の素因数を見つけることである. 大きな桁数の素因数を見つけることにおける重要性として, RSA 暗号の適切な鍵長の分析や安全性の評価などが挙げられる. RSA 暗号とは, 約 40 年前に考案された公開鍵暗号であり, 現在でも広く利用されている. RSA 暗号は, 大きい桁数である合成数の素因数分解が難しいことから安全性が保証されている. また, 近年ではより高い安全性を持つ楕円曲線暗号が普及しつつある. 楕円曲線暗号とは, 約 35 年前に考案された公開鍵暗号であり, 楕円曲線上の離散対数問題 (ECDLP) の困難性が安全性に依拠している暗号である. 今回, FUN-ECM プロジェクトでは, 楕円曲線法によってより大きな桁数の素因数分解を行うプログラムを作成, 改良し RSA 暗号の安全性を検証する. 検証のため, 大きい数の素因数分解をランキングしたサイトである, STUDIO KAMADA[1] や ECMNET[2] へのランクインを目標とした. また今年度の目標として, 素因数分解を行うプログラムの改良, 新たなプログラムの作成, FUN-ECM プロジェクトにおける Web ページの更新という 3 つの目標を掲げ活動を行った.

私たちの主な活動は昨年度までの FUN-ECM プロジェクトで作成してきた ECM プログラムである funecm の改良, PARI/GP によるプログラムの実装と改良, Web サイトの作成と改善である. その他にも昨年度のプログラムを運用し, STUDIO KAMADA に載っている合成数の分解を試みている.

まずは楕円曲線法の理論の学習をした. 具体的には楕円曲線の学習を行い, 続いて楕円曲線法についての学習を行った. その後プログラムの改良, および新たなプログラムを作成するプログラム班と,Web ページの更新を行うための Web 班に分かれて活動を行った.

プログラム班では, ECM の部分である Stage 1, Stage 2 の学習を行った. そして PARI/GP での実装をするために, PARI/GP の使い方やプログラムの書き方を学んだ. その後, PARI/GP にて Stage 1 と Stage 2 を実装した. 実装後, Stage 1,Stage 2 の効率の良い方法を学習した. 次に, Stage 1, Stage 2 を改良した. Stage 2 に関しては, *Baby-step Giant-step* を用いた. また, 昨年度のプログラムの動かし方を学び, 実際に動作させて性能を確認した. 最後に昨年度のプログラムと同環境下で, 平均実行時間による速度比較を行った.

Web 班では, はじめに去年度の Web サイトの問題点を探した. 次に HTML と CSS の基礎学習と新たな Web サイトのレイアウトの考案を行った. その後,Web サイトの作成とロゴの作成を行った.

これらの活動を, チームで協力しながら活動を行った. その結果, PARI/GP でのプログラムの実装, 改良に成功し, 昨年度のプログラムを用いることで STUDIO KAMADA に名前を載せることが出来た.

キーワード　素因数分解, 楕円曲線法, *Baby-step Giant-step*, STUDIO KAMADA, ECMNET, RSA 暗号

（文責: 下地健介）

Abstract

The goal of our project team is to find prime factor as large as possible by using ECM (Elliptic Curve Method). RSA cryptosystem is related to the importance to find large prime factor. RSA cryptosystem was invented 40 years ago. It has been used for digital signatures. Security of RSA is based on the hardness of prime factorization of large composite number. However, in recent years, elliptic curve cryptography is used because it is more safe than RSA cryptosystem. Elliptic curve cryptography was invented about 35 years ago. Elliptic curve cryptography is guaranteed safety by security of RSA is based on the hardness of In this project, we are making and improving the program which does prime factorization. Then, we are verifying the safety of RSA cryptosystem by using the program. For verifying, we determined our goals which is to rank in the STUDIO KAMADA and ECMNET. Also, in this year, we acted for our goals which is to improve the program doing prime factorization, develop a new program and renew our webpage.

We mostly implement ECM by PARI/GP, and improve the "funecm", which is the program of last year. In addition, we have been tried to do prime factorization of composite numbers of STUDIO KAMADA by using the program of last year.

Firstly, we learned about theory of ECM. Afterwards, We divided the group responsible for the program and the group responsible for the web site. The group responsible for the program learned Stage 1, Stage 2 of ECM. Afterwards, this group learned how to use PARI/GP and how to write program, in order to implement ECM. Then, this group implemented Stage 1 and Stage2 of ECM. After implement, this group learned method of Stage 1, Stage 2 that become faster than before. Also, this group learned how to do the program of last year, and confirmed its performance. Lastly, this group compared speed of the program of last year and the program of PARI/GP about Stage 1.
The group responsible for the web site first looked for problems with the web page last year. Next, this group learned about HTML and CSS and considered the layout a new web page. After that, this group created a web page and a logo.

In result, we implemented the program of PARI/GP. In addition, our project ranked in STUDIO KAMADA by using the program of last year.

Keyword Prime factorization, Elliptic Curve Method, Baby-step Giant-step, STUDIO KAMADA, ECMNET, RSA cryptosystem

（文責: 下地健介）

目次

第 1 章　背景

RSA 暗号の安全性は, 大きい桁数の素因数分解が難しいことに依拠している. 私たちの目的は, 楕円曲線法 (ECM) を利用し高速に素因数分解を行うことにより RSA 暗号の安全性を検証することである. また, プログラムを利用して大きい桁数の素因数分解を行い, STUDIO KAMADA や ECMNET へ名前を載せることも目標としている.

（文責: 下地健介）

1.1　前年度の成果

暗号技術は, プライバシーの保護やコンピュータセキュリティにおいて必須となる技術である. 暗号技術は TLS 通信や, 無線 LAN での通信など多くの場面で利用されている. しかし, 暗号技術は常に進化する攻撃方法により解読の危険性がある. 様々な攻撃方法に対しても解読されない暗号アルゴリズムを作成するためには, 作成者が暗号解読の手段を知る必要がある. 暗号の安全性評価には暗号解読の技術が利用され, 暗号の強度は解読に必要な情報量と計算量により評価される. 今回のプロジェクトでは, RSA 暗号の解読に関係する素因数分解アルゴリズムの 1 つである楕円曲線法 (ECM) について学ぶ.

現在, 有名な公開鍵暗号には RSA 暗号や楕円曲線暗号がある. RSA 暗号は, 桁数が大きい合成数の素因数分解の困難性に依っている暗号である. 楕円曲線暗号は, 楕円曲線上の離散対数問題 (ECDLP) の困難性に依っている暗号である. どちらも有名な暗号であるが, 当プロジェクトでは RSA 暗号を対象とする. RSA 暗号を解読する際は, 合成数の元である 2 つの素因数を見つける必要がある. ECM では, 与えられた曲線の点のスカラー倍が無限遠点になった場合に素因数が発見される. この性質を利用し, RSA 暗号を解読する. 尚, ECM と楕円曲線暗号はプログラムに類似している部分があるが, 目的は全く別であることに注意されたい.

ECM には Stage 1 と Stage 2 があり, Stage 1 で素因数を発見できなかった時, Stage 2 で因数の発見を試みる. 前年度のプロジェクトでは ECM のプログラムである funecm がほぼ完成していたため, 今年度のプロジェクトでは funecm の改良, 新しいプログラムの実現を目指した. また, STUDIO KAMADA, ECMNET という分解した素因数の大きさを競うサイトがあり, 昨年度のプログラムや今年度のプログラムを動かすことで STUDIO KAMADA や ECMNET に名前を載せることを目標として掲げた.

（文責: 下地健介）

1.2　ECMNET とは

ECMNET[1] とは, ECM を用いてカニンガム数を素因数分解し, 発見した素因数の大きさをランキング形式で競う Web サイトである. ECMNET は, 特定の範囲のカニンガム数の素因数分解を行うことで, Cunningham project に貢献することを目標として掲げている. このサイトに載るためには, 現在登録されている素因数よりも大きな素因数を見つける必要がある.

カニンガム数と Cunningham project の詳細を次項に記す.

カニンガム数

以下の条件を満たす数をカニンガム数という. なお ECMNET では, 変数 b, n の範囲を限定している.

$$\text{カニンガム数: } b^n \pm 1 \ s.t. \ b, n \in \mathbb{N},\ b \text{ は累乗数ではない}$$

Cunningham project

Cunningham project とは, b, n を表 1.1 の範囲に限定したカニンガム数を素因数分解するプロジェクトである.

表 1.1　Cunningham project における b, n の範囲

b	2	3	5	6	7	10	11	12
n の上限	1300	850	550	500	450	400	350	350

（文責: 松崎伸彦）

1.3　STUDIO KAMADA とは

STUDIO KAMADA[2] とは, 鎌田誠氏により開設されたレプディジットやニアレプディジットなどの合成数の素因数分解を行い, 結果を報告する Web サイトである. 素因数分解するための方法も多様であり, その中の 1 つに ECM がある. ECMNET と同様に, 分解した素因数の大きさをランキング形式で競う. STUDIO KAMADA で対象とされている合成数を以下で説明する.

レピュニット

repeated unit（反復単位数）の略で, 1, 11, 111, 1111…のような 1 だけからなる自然数.

レプディジット

repeated digit の略で, レピュニットを含む, すべての桁の数字が同じ（ぞろ目）自然数.

ニアレプディジット

near-repeated unit の略で, レプディジットから 1 桁だけ他の数字に置き換えた自然数.

プラトウアンドデプレッション

レプディジットから, 両端の数字を共通の異なる数字に置き換えた自然数.

クワージレプディジット

レプディジットから, 2 つの桁の数字を共通の異なる数字に置き換えた自然数.

ニアレプディジット回文数

ニアレプディジットのうち, 桁数が奇数で中央の数字だけが異なる自然数.

（文責: 松崎伸彦）

1.4　今年度の課題の概要

本プロジェクトでは, 去年の時点で ECM のプログラムはほぼ完成していたため, 昨年度のプログラムの改良や新たなプログラムにより ECM を実現することを目指す. そして, STUDIO

KAMADA や ECMNET へのランクインも目指す.

（文責: 谷山真子）

第 2 章　到達目標

2.1　本プロジェクトにおける目的

RSA 暗号は桁数の大きい合成数の因数分解が難しいことが安全性に繋がっている．しかし, 巨大な合成数の因数分解が容易にできてしまうと RSA 暗号は簡単に解読されてしまう．そうなると RSA 暗号の安全性が保障されない．当プロジェクトでは, RSA 暗号の安全性を検証するために素因数分解を行うプログラムを作成した．また,FUNECM の活動や ECM について知ってもらう必要があると考えた. 理由として,ECM や FUNECM の活動を知ってもらうことで,ECM をより世の中に広めることが出来ると考えたためである. この目的を達成するため, 新しく Web サイトを作成した.

（文責: 谷山真子）

2.2　PARI/GP での実装

まず, 昨年度のプログラムの性能及び動かし方を確認するために, STUDIO KAMADA に記載されている合成数などを用いて計算を行った．STUDIO KAMADA に記載されている合成数の素因数分解ができた場合, 当プロジェクトの名前が STUDIO KAMADA に載る．その後, 昨年度のプログラムの内容を確認した．昨年度のプログラムのコードが膨大になっており, 容易にその全体を把握することは困難だったため, 新しく PARI/GP でプログラムを実装することを目標とした. この目標を達成するために以下のことを実施した.

- ECM の基礎的な学習
- ECM の Stage 1 の理解
- mod の理解
- PARI/GP の使い方についての学習
- PARI/GP のプログラムの書き方についての学習
- Stage 1 のより効率の良い方法についての学習
- Stage 1 のためのプログラムの作成
- Stage 1 の改良
- ECM の Stage 2 の理解
- *Baby-step Giant-step* についての学習
- Stage 2 の改良

（文責: 谷山真子）

2.3 課題設定

本プロジェクトでは, 5 月上旬まで全員で ECM についての基礎学習を行った. 5 月中旬以降, 昨年度のプログラムを動かしつつ, 新しく PARI/GP で実装を行った. その後, Stage 1 について改良を行った. 後期ではプログラム班と WEb 班の 2 つに分かれて活動を行った.

プログラム班では Stage 2 の実装を行い, *Baby-step Giant-step* による Stage 2 の改良を行った. その後, PARI/GP で楕円曲線法の実装に取り組んだ.

Web 班では昨年度に作成された現状の Web サイトの問題点を探し出し, その問題点を解決した新しい web サイトを作成することとした. そのため,HTML と CSS の基礎学習と新しい Web サイトのレイアウトの考案や新しいロゴの考案を行った. その後, レイアウトを元に新しい Web サイトの作成を行った.

（文責: 谷山真子）

第 3 章　活動内容

プロジェクトが始まった当初では, 楕円曲線についての基礎知識がなかったため, 昨年度のプロジェクトで基礎知識を身につけるために使われた資料 [3] 及び楕円曲線論入門 [4] を用いて, 理解した. その後, PARI/GP についての学習を行い, プロジェクトを進行した. 後期は ECM の改良方法について調査し, Stage 2 や *Baby-step Giant-step*, 素数テーブルを実装した. その後は速度比較を行い, PARI/GP のプログラムの性能を調査した.

（文責: 松崎伸彦）

3.1　基礎学習

楕円曲線法を理解するために 5 月の中頃まで全員が楕円曲線法のアルゴリズムや基礎知識についての学習を行った. 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 では合成数剰余で計算を行っていく.

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で a, b の最大公約数を求めるものである. 以降, a,b の最大公約数を $\gcd(a,b)$ と表記する.

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

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 $ax + by = c$ の整数解は, 1 組存在するならば無数に存在する. この方程式を一次不定方程式と言う. 一次不定方程式を解くためには, 拡張 Euclid の互除法が有効である. 拡張 Euclid の互除法は, 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 を効率よく求めることができる.

有限体 $\mathbb{F}_p$ において除算 $a \div b$ 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_0, y_0) を求められる. このとき, $px_0 + by_0 = 1$ となるので, 有限体 $\mathbb{F}_p$ 上では $by = 1$ より $b^{-1} = y_0$ が成立する. これより

$$a \div b = a \times b^{-1} = a \times y_0$$

と変形することで, 除算を逆元計算と乗算で計算できる.

楕円曲線の定義

$a, b \in \mathbb{F}_p$ に対して $y^2 = x^3 + ax + b(4a^3 + 27b^2 \neq 0)$ で表される曲線を有限体 $\mathbb{F}_p$ 上の楕円曲線という.

楕円曲線の加算と逆元

楕円曲線上のある 2 点 P,Q を通る直線 l を考える. この時, 楕円曲線と直線 l の交点を $R'(= P * Q)$ とし, x 軸に関する対称点を R とする. 得られた点 R を R' の逆元と呼び , $R = -R'$ と表記する. また, R を P と Q を加算した点と定義し, $R = P + Q$ と表記する.

無限遠点

楕円曲線上の点 P とその逆元 $-P$ を取り, $P + (-P)$ を考える. その時, 2 点を通る直線と楕円曲線には P と $-P$ 以外に交点が存在しない. このような場合に, 存在しない点を仮想的に考え, その点を無限遠点 O と呼ぶ. 無限遠点を考えると, $P + (-P) = O$ が成り立つ.

楕円曲線の 2 倍算

楕円曲線上の点 P の接線を l とし, 楕円曲線と直線 l の P 以外の交点を R' とし, R' の逆元を R とする. この R は $R = P + P = 2P$ であり, 楕円曲線の 2 倍算となる. 上記で説明した $+$ に関して, E は O を単位元とする群になる. なお, 点の個数を位数という.

楕円曲線のスカラー倍算

点 P と自然数 d に対して, 点 P を d 倍 ($dP = P + P + P + \cdots + P$($d$ 個の和)) することを, 楕円曲線のスカラー倍算という.

（文責: 下地健介）

3.2 楕円曲線法

ECM は, 1985 年 Lenstra によって提案された最大素因数が数十桁のような中規模合成数にとって最良と言われている素因数分解アルゴリズムである. 楕円曲線は曲線のパラメータが変わることにより, 位数がランダムに変わる性質がある. そのため群の位数と合成数の相性が悪ければ, パラメータを変更し曲線を変更すればよいので, 効率的に因数を見つけられる. ECM は大きく Stage 1, Stage 2 に分けることができる.

ECM の Stage 1

本節では, 合成数 N の因数 p を見つけることについて考える. 合成数 N の桁数から, 因数の大きさを予想して, パラメータ B_1, B_2 を決める. 本プロジェクトでは B_1, B_2 の値を ECMNET から引用した. ECMNET における合成数 N と B_1, B_2 の対応表を表 3.1 に示す.

表 3.1　素因数の桁数に対応する適切な B_1, B_2 の値

digits	20	25	30	35	40	45	50
optimal B_1	11,000	50,000	250,000	1,000,000	3,000,000	11,000,000	43,000,000
optimal B_2	1,100,000	5,000,000	25,000,000	100,000,000	300,000,000	1,100,000,000	4,300,000,000

ある楕円曲線上の射影座標で与えられた点 $P = (X_P : Y_P : Z_P)$ に対し, 以下の手順によって, ある確率で因数を見つけることができる.

Step 1 楕円曲線を決定し, 点 P を選択する.

Step 2 B_1 以下の自然数全てにおける最小公倍数 l =lcm(2,3,⋯,B_1) を求める.

Step 3 点 $Q = lP = (X_Q : Y_Q : Z_Q)$ を計算する.

Step 4 Z_Q と N の最大公約数 d =gcd(Z_Q, N) を計算する.

(1) d が 1 よりも大きいならば, d を出力してプログラムを終了する.

(2) d が 1 ならば, Stage 2 に移行する.

ECM の Stage 2

Stage 1 のみを利用していると, 素因数を見つけやすくするには B_1 を大きくし, 非常に大きい数 l に対して, スカラー倍 lP を計算する必要がある. スカラー倍の計算は, 非常に計算量が大きくなるため, Stage 2 を利用する.

ECM の Stage 1 で因数を発見することができなかったとき, mod p での E の位数は, 少なくとも B_1 よりも大きい. よって, ECM の Stage 2 では, B_1 から B_2 までの素数 s の中に, 位数があるかを調べる.

$B_1 < s < B_2$ を満たす全ての素数 $s_1, s_2, s_3, \cdots, s_k$ について Stage 1 の結果で得られた点 Q に対するスカラー倍 $s_iQ = Q_i$ をそれぞれ求める.

$$
\begin{aligned}
s_1 \times Q = Q_1 = (x_1, y_1) \\
s_2 \times Q = Q_2 = (x_2, y_2) \\
s_3 \times Q = Q_3 = (x_3, y_3) \\
\cdots \\
s_k \times Q = Q_k = (x_k, y_k)
\end{aligned}
$$

$x_1, x_2, x_3, \cdots, x_k$ に対して, 合成数 N との最大公約数を求める. $Q_1, Q_2, Q_3, \cdots, Q_k$ のうちの点 Q_i で $Q_i = O \pmod p$ を満たせば素因数が求まり, 成功する. 以上のことを基礎学習として学んだ. 以下の章では, 活動内容を記述する.

（文責: 下地健介）

3.3 プログラム班の活動

3.3.1 高速化の方法の検討

今年度では, PARI/GP を使って, ECM での素因数分解を行うプログラムを作成することで高速化, ECM が何を行っているのかを把握しやすくすることを図った. 高速化に関しては, 通常の ECM よりも効率の良い方法を実装することで更なる高速化を図った.

まず, Stage 1 では P に対するスカラー倍は B_1 の階乗よりも 1 から B_1 までの数すべての最小公倍数を入れた方が効率がよくなる. そのため, 1 から B_1 までの数すべての最小公倍数を求めるプログラムを作成した.

また, 本来の Stage 2 では B_1 から B_2 までの素数を Stage 1 が終わった後の点のスカラー倍に入れていくが, それよりも効率の良い *Baby-step Giant-step* がある. そして *Baby-step Giant-step* の高速化として素数テーブルがある. これらを用いることで高速化を目指した.

（文責: 松崎伸彦）

3.3.2 Baby-step Giant-step

Baby-step Giant-step とは, $B_1 < s < B_2$ を満たす全ての素数 $s_1, s_2, s_3, \cdots, s_k$ に対して, s を変形して計算を行うことで高速化を図る手法である． Stage 2 は $sP = O$ の際に成功する. この式を変形する. まず s を

$$s = av + u(\tfrac{-a}{2} < u < \tfrac{a}{2})$$

と変形すると

$$sQ \equiv O$$
$$(av + u)Q \equiv P$$
$$(av)Q = -uQ$$

に変形することができる．この式を満たせば Stage 2 は成功する．(uQ は$\frac{-a}{2} < u < \frac{a}{2}$であるため事前に計算できる) ここで逆元の関係にある点の x 座標は等しい．よって $(av)Q$ の x 座標を $G_x, -uQ$ の x 座標を H_xとすると

$$G_x \equiv H_x \pmod p \; G_x - H_x \equiv 0 \pmod p$$

が成り立つ. $G_x - H_x \equiv 0 \pmod p$ より $G_x - H_x$ は p の倍数であることがわかる.

つまり, $G_x - H_x \equiv 0 (\mathrm{mod}\ p)$ となる s が存在するならば, $(av)P = -uP$ となる s が存在する. すなわち, $sP = O$ となる s が存在するため, Stage 2 が成功し, 素因数分解がなされる.

ここで, 全ての s について $G_x - H_x$ を計算し, すべてを掛け合わせた数を d とする. $G_x - H_x \equiv 0 \pmod p$ となる s が存在すれば, $d \equiv 0 \pmod p$ となる. よって d と合成数 N の最大公約数を求めることで素位数 p が求められる. *Baby-step Giant-step* の手順は, 以下に示す. ここで $a = \sqrt{B_2}$とする.

Step 1 $B_1 < s < B_2$ を満たすすべての素数 s について計算する

(1) 素数 s について v,u を求める
(2) $(av)Q = -uQ$ の左辺の x 座標 G_x, 右辺の x 座標 H_x を求める.

Step 2 $d = (G_{x_1} - H_{x_1})(G_{x_1} - H_{x_2}) \cdots (G_{x_1} - H_{x_j})(G_{x_2} - H_{x_1})(G_{x_2} - H_{x_2}) \cdots (G_{x_2} - H_{x_j})$
$\cdots (G_{x_i} - H_{x_1})(G_{x_i} - H_{x_2}) \cdots (G_{x_i} - H_{x_j})$ を計算する.
$(i = \sqrt{B_2}, j = \frac{\sqrt{B_2}}{4})$

Step 3 d と合成数 N の最大公約数をとる.

Step 4 最大公約数が 1, N 以外なら成功.

（文責: 松崎伸彦）

Algorithm 2 Baby-step Giant-step を用いた ECM の Stage 2

Require: E: 楕円曲線, Q_0: 楕円曲線法の Stage 1 で得られた点, B_1, B_2

Ensure: p: 因数

```
for each i=1 to a-1 do
  H[i] ← iQ_0
end for
Q ⇐ aQ_0
d ⇐ 1
for each prime s = B_1 to B_2 do
  u ⇐ s/a
  v ⇐ s%a
  G ⇐ vQ
  d ⇐ d(G_x − H[i]_x)
end for
q ⇐gcd(d,N)
if q > 1 then
  return q
else
  return FAIL
end if
```

3.3.3 素数テーブルの実装

Baby-step Giant-step の改良として素数テーブルを用いた．*Baby-step Giant-step* では x 座標の差を掛け合わせるが, この差が素数でなければかける必要はない．そのため, 素数かどうかを判断することが高速化につながる．PARI/GP の関数としては isprime 関数があるが時間がかかる．そのため, 素数テーブルを実装した．素数テーブルとはあらかじめメモリ上に小さい順に素数が用意されたものである．メモリ上には連番で配置されており, 配列としてアクセスすることで n 番目の素数が取得できる．これにより, 一瞬で素数判定ができるため, 高速化につながる．素数テーブルは予め別のプログラムで作成し用いる．この手法は多くのメモリを消費し, 素数テーブルを読み込む時間が必要となるが, その後の素数判定の部分が大幅に高速化される．

（文責: 松崎伸彦）

3.3.4 速度比較

作成したプログラムを用いて昨年度のプログラムと実行時間を比較した．PARI/GP の実行環境は Core i3-7100U 及び Xeon E5-2640 を用いた．昨年度の実行環境は Xeon E5-2640 を用いた．Core i3-7100U のスペックとしては, CPU2 コアである．Xeon E5-2640 のスペックは, CPU12 コアである．この時, $B_1 = 3000000$, $B_2 = 300000000$, 分解する合成数は STUDIO KAMADA に載っている桁数 159 桁の合成数とした．そして, 200 回の合計試行時間から平均を取り, 昨年度のプログラムは一番速い実行時間を平均として比較した．表 3.2 に 1 回にかかる Stage 1, Stage 2 の平

均時間を示す. Stage 1 での実行時間は funecm と比べて約 1.14 倍速くなった.

表 3.2　Stage 1 の速度比較

	PARI/GP	(Xeon)funecm
平均時間 (s)	39.24	44.57

また, Stage 1, Stage 2 合わせた速度比較も行った. PARI/GP の実行環境は Core i3-7100U 及び Xeon E5-2640 を用いた. 昨年度の実行環境は Xeon E5-2640 を用いた. この時, $B_1 = 4572523$, $B_2 = 457252300$, 分解する合成数は STUDIO KAMADA に載っている合成数とした. そして, 100 回の合計試行時間から平均を取った. 表 3.3 に 1 回にかかる Stage 1, Stage 2 の平均時間を示す. 括弧内には用いた CPU を示す. ここで, Core とは Core i3-7100U を指し, Xeon とは Xeon E5-2640 を指すこととする.

表 3.3　PARI/GP のプログラムと昨年度までのプログラムの速度比較

	(Core)PARI/GP	(Xeon)PARI/GP	(Xeon)funecm
Stage 1 にかかる平均時間 (s)	51.44	84.15	67.78
Stage 2 にかかる平均時間 (s)	31.80	24.23	27.51

Core i3 を用いた PARI/GP のプログラムと, Xeon を用いた funecm に着目する. Stage 1 では Core を用いた PARI/GP のプログラムの方が約 1.32 倍速くなった. Stage 2 では Core を用いた PARI/GP のプログラムの方が遅くなったものの, 全体としては Core の PARI/GP のプログラムが約 1.15 倍速くなった. 次に, Xeon を用いた PARI/GP のプログラムと, 同じく Xeon を用いた funecm に着目する. Stage 2 では PARI/GP のプログラムの方が約 1.14 倍速くなった. しかし, 全体として比較すると funecm の方が約 1.4 倍速いという結果となった.

（文責: 松崎伸彦）

3.3.5　昨年度のプログラムの運用

今年度では, 昨年度のプログラムには一切変更を加えず運用し, STUDIO KAMADA に載っている合成数を分解することを試みた. その結果, 1 つの合成数を分解することに成功し, STUDIO KAMADA に funecm の名前が載った.

（文責: 松崎伸彦）

3.3.6　PARI/GP の運用

昨年度のプログラムではプログラムのコードが膨大になり, 全体の把握が難しくなったこともあり, 今年度では, 新しく PARI/GP での一からの実装を試みた. そして, Stage 1 と Stage2 を合わせ素因数分解を行った. その結果, ランダムな 10 桁同士の素数の積である合成数を分解することに成功した.

（文責: 谷山真子）

3.4 Web 班の活動

3.4.1 Web サイトの構造化

後期から, 楕円曲線法をより一般に知ってもらうために新しく Web サイトを作成することとなった. はじめに, 去年度の Web サイトを参考にし, どのような Web サイトを作成するかをメンバーと話し合った. 去年度の Web サイトは, 老若男女幅広い年代の方々が親しみやすく, 操作しやすい Web サイトを目指していた. また, 去年度の Web サイトの問題点は 2 つあった. 1 つ目は, デザインの問題である. Web サイトのページが中心からずれていたり, スマートフォンに対応していなかった. 2 つ目は, 掲載されている楕円曲線法の説明文が長くなり, 分かりづらくなっていた. それらの問題点を踏まえて今年の Web サイトは対象者である高校生が見やすく, 理解しやすい Web サイトを目指した.

次に, 高校生が理解しやすいような Web サイトにはどのようなものが必要であるかを話し合った. 挙がった意見では, 高校生は楕円曲線法を調べることがない, というものであった. そのため, 公開鍵暗号方式や共通鍵暗号方式などの暗号の説明を丁寧にしてから, 楕円曲線法について説明することとなった. また, 数式をできるだけ使わず, 図を多く入れることを重視すべきだと考えた.

（文責: 谷山真子）

3.4.2 HTML と CSS の学習

Web 班のメンバーは HTML と CSS をほとんど触ったことがなかった. そのため, HTML と CSS の参考書を用いて, 2 週間ほど学習した. 参考書は, HTML などのタグの説明とソースコード, 実行結果が並んで表示されており, その工程をすべて行うと一つの Web サイトが作成できる仕組みとなっていた. はじめに大まかな Web サイトを HTML で作成した. その後, CSS を実際に使ってコーディングした. 実際に Web サイトが作成することができる点が魅力的であった. このことによって, 理解しやすく, 楽しみながら学習を進めることができた. また, よく分からないところやうまくコンパイルできなかった時には, メンバー同士で作成したソースコードを見ながら学習を行った. 学習はメンバーが揃って行っていたので助け合い, 情報共有しながらスムーズに進めることができた.

（文責: 谷山真子）

3.4.3 全体のレイアウトの試案

Web サイトのレイアウトを決定する前に, 広報班のメンバーで様々な Web サイトを閲覧し, どのようなサイトが見やすいかを挙げた. そのようなサイトはシンプルなデザインで, ページごとに色が統一されていた. また, 画面上部にヘッダーや画面下部にフッターがあった. その後, 閲覧した Web サイトを参考にしながら, Web サイトのレイアウトを作成した. 採用したレイアウトは, 画面上部にヘッダーがあり, シンプルなデザインなものである. このとき, ワイヤーフレームを作成し, 実際にページの移動や見出しの位置などを確認した. ワイヤーフレームを作成する際には, Adobe

の XD というソフトを用いた．Adobe XD は, ユーザ操作性をデザイン, プロトタイプ化し, ユーザの立場になってそのレイアウトを確認することができるものである．

（文責: 谷山真子）

3.4.4 ロゴの制作

FUN-ECM の Web サイトにはすでにロゴがあった．去年のロゴは十分良いものであったが, 今回の Web サイトはシンプルなものを目指していたので, 去年のロゴの良い点を取り入れながら新しく制作した．新しいロゴには, 去年と引き続き代表的な楕円曲線と, 新たに有限体の記号を追加した．また, 全体の色を黒にした．このロゴは Web サイトの画面最上部に使用した．ロゴを制作したのは Web 班の富樫である．Adobe Illustrator を用いて制作した．

（文責: 谷山真子）

3.4.5 カテゴリ分け

Web サイトで主に伝えたいことを軸とし, それに付随する情報のカテゴリを決定した．Web サイトの軸は楕円曲線法とはどのようなものかを知ってもらうことである．カテゴリは, HOME, 暗号, 楕円曲線法 (ECM), 活動内容, Link, その他である．

（文責: 谷山真子）

3.4.6 Web サイトの作成

今年の Web サイト作成は Ameba Ownd という Web サイト無料作成サービスを用いた．Ameba Ownd を用いた理由としては 3 つ挙げられる．1 つ目は, 複数人で 1 つのものを編集することができることである．後でファイルをまとめる必要がなく, 他のメンバーが制作した部分を見ることができるため, 間違っているところをお互いに直しながら作成することができる．2 つ目は, 操作が簡単なことである．Ameba Ownd では用意されているデザインの中から 1 つ選び, 編集していく．画像, 地図やテキストなどを簡単に入れることができる．さらに, 細かいコーディングやメンバーがそれぞれコーディングするわけではないので, ページのずれが起きない．また, スマートフォンやタブレットにも対応しているため, 去年の問題点が解決する．3 つ目は, HTML と CSS でのコーディングが可能であることである．Ameba Ownd ではテキストをそのまま打ち込めるが, 色や大きさなどを細かく指定するとき, 線を引いたり囲んだりしたいときに, HTML と CSS を用いることができる．Ameba Ownd 内での HTML と CSS は通常のタグを使用することができる．以上の点を踏まえて, Ameba Ownd を利用することにした．

次に, 主に担当するカテゴリを決めた．下地が HOME と暗号を, 谷山が楕円曲線法 (ECM) とその他である．活動内容と Link は 2 人で作成した．担当するページを作り, 時々プレビューを確認しながら進めていった．こうしたことによって, Ameba Ownd 内ではテキストとして入力したものと, HTML として文章を入力したもので大きさが異なることに早めの段階で気づくことができた．メンバーと話し合った結果, HOME 以外のすべてのページで文字の大きさを調整できる HTML として書くことに決めた．また, 画像の大きさがページによって異なることが分かった．また, 楕

円曲線法と暗号の仕組みの図はすべて Adobe Illustrator を用いて作った．楕円曲線のグラフは Python で作成した．

最終成果発表会の一週間前に, 全体の微調整と最終確認を行ったところで終わった．

図 3.1　トップページ

FUN-ΣCM

図 3.2　ロゴ

（文責: 谷山真子）

3.5　中間発表

3.5.1　発表準備

3.5.1.1　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, ECM についての説明, 活動内容, 今後の課題の 4 つの項目に分けて作成した．ポスターの作成には「Adobe Illustrator」というソフトウェアを使用した．ポスターが完成次第, 班メンバー全員でレビューを行い, 誤字脱字やフォントサイズ等の修正をした．その後, 最終確認として担当教員にレビューを行なってもらい, 必要な情報を追加した．

（文責: 富樫北斗）

3.5.1.2　スライド

本プロジェクトを説明するのにポスターと共に, プレゼンテーション資料としてスライドを作成することにした．情報デザインコースに所属している富樫を中心に, 他メンバーと話し合いながら発表内容を考えた．

スライド作成には「Microsoft Power Point」を使用し, 「Slack」を使用してスライド内容を常に共有した．模擬発表を行い, 修正点をあぶり出し, 文章で伝わりづらい部分の図解表現をスライドに挿入し, わかりやすく見やすいスライドを目指した．

（文責: 富樫北斗）

3.5.1.3 発表原稿

前述のプレゼンテーション資料と並行して発表用原稿の作成を行った．ECM について理解してもらえるようするために，基礎学習に関する説明に重点を置いた. 作成した原稿はプレゼンテーション資料作成者とともに表現のすり合わせを行い，伝わりにくい表現，並びに資料との表記ゆれ等の修正を行った. 加えて，担当教員にも文章などの添削をしていただき，伝わりにくいと思われる部分の修正を行った.

（文責: 富樫北斗）

3.5.2 アンケート

中間発表用アンケートを作成した．アンケートの質問項目は

- 発表技術について（10 段階評価, コメント）
- 発表内容について（10 段階評価, コメント）

の 2 点であった.

中間発表終了後, アンケートの回答の分析を行った．アンケートは発表を聞いた教員生徒合計 38 人が回答を行った．分析結果として「発表技術について」は, 10 段階で平均 6.5 であった．コメントとしては, 「声が小さい」「原稿を見すぎ」「スライドが小さい」「スライドの図と式がわかりやすい」などがあった．また, 「発表内容について」では, 10 段階評価で平均 6.9 であった．コメントとしては, 「目的がよくわからない」「素因数分解を解く理由がわからない」「素因数分解を行うアルゴリズムの説明がよかった」「ECM がどのように使われているのかを説明した方がよい」などがあった.

アンケートの分析結果として, 発表技術では原稿の見過ぎなど発表者の準備不足が考えられる．また, 発表内容については, 低い評価が目立った．理由として, スライドの順序が悪かった, ECM への理解が難しかったなどが考えられる.

以上のことから反省点として, 発表の準備不足が考えられる．また, スライドの順序の再検討や ECM についてわかりやすい説明を行う必要があった.

（文責: 富樫北斗）

3.5.3 発表

中間発表は, 7 月 14 日 (4 時限 5 時限) に, 3 階 394 前で行った．発表回数は, 全 3 回で後半のみ行った．1 回あたり 15 分の発表で, 10 分間プレゼンテーションを行い, 残り 5 分間質問を受けた．人数の振り分けは 4 人のうち 3 人を毎回選び, 3 回を発表順番を変えながら行った.

発表はスライドの説明をメインとし, ポスターにはスライドの内容をより詳しく説明したものを用意した．中間発表会での発表を行い, アンケートを集計した結果, 高評価もあったが, 低い評価が目立った．反省点として, 準備不足やスライドの順序の再検討などが挙げられた.

（文責: 富樫北斗）

3.6 成果発表会

3.6.1 発表準備

3.6.1.1 ポスター

初めに, 中間発表で使用したポスターを見直した. その結果, 中間発表で使用した概要と楕円曲線法について説明しているメインポスターはそのまま利用し, 活動内容と今後の課題について説明していたサブポスターを変更することにした. 具体的には, サブポスターではプログラム班と Web 班のそれぞれの活動内容を記載した. プログラム班では, 後期の目標, プログラムの改良案, その実行結果, プログラムの速度比較について紹介した. Web 班では,Web サイトを作成する過程を構造化, カテゴリ化, 構築の 3 つに分けて紹介した. また, 実際に作成した Web ページとその内容についても紹介した. ポスターの完成後, 班メンバー全員でレビューを行い, 誤字脱字やフォントサイズ等の修正をした.

（文責: 富樫北斗）

3.6.1.2 スライド

本プロジェクトを説明するのにポスターと共に, プレゼンテーション資料としてスライドを作成することにした. 情報デザインコースに所属している富樫を中心に, 他メンバーと話し合いながら発表内容を考えた.

スライド作成には「Microsoft Power Point」を使用し, 「Slack」を使用してスライド内容を常に共有した. 模擬発表を行い, 修正点をあぶり出し, 文章で伝わりづらい部分の図解表現をスライドに挿入し, わかりやすく見やすいスライドを目指した. 後期ではプログラム班と Web 班に分かれて活動を行ったため, スライドの構成を班ごとに分けて説明することにした. スライドの説明で専門用語を多く使うので, 出来るだけ簡潔な説明を心掛けた.

（文責: 富樫北斗）

3.6.1.3 発表原稿

前述のプレゼンテーション資料と並行して発表用原稿の作成を行った. ECM について理解してもらえるようするために，基礎学習に関する説明に重点を置いた. 作成した原稿はプレゼンテーション資料作成者とともに表現のすり合わせを行い，伝わりにくい表現，並びに資料との表記ゆれ等の修正を行った. 加えて，担当教員にも文章などの添削をしていただき，伝わりにくいと思われる部分の修正を行った.

実際に発表を行うと, 原稿を見過ぎてしまい上手に伝えることが出来なかった. 本プロジェクトでは, 楕円曲線の説明やプログラムの説明が複雑になりがちである. そのような複雑な説明において, 原稿を見てしまうのは仕方ないが, ある程度の説明は覚えておく必要があるように感じた.

（文責: 富樫北斗）

3.6.2 アンケート

- 発表技術について（10 段階評価, コメント）
- 発表内容について（10 段階評価, コメント）

の 2 点であった.

成果発表終了後, アンケートの回答の分析を行った. アンケートは発表を聞いた教員生徒合計 30 人が回答を行った. 分析結果として「発表技術について」は, 10 段階で平均 7.2 であった. コメントとしては, 「声が小さい」「丁寧で聞きやすかった」「原稿を見すぎ」「スライドが小さい」などがあった. また, 「発表内容について」では, 10 段階評価で平均 7.2 であった. コメントとしては, 「目的がよくわからない」「素因数分解を解く理由がわからない」「なぜ高速化するのかわからなかった」などがあった.

アンケートの分析結果として, 発表技術では中間発表と同様に原稿の見過ぎや声の小ささなど発表者の準備不足があったと考えられる. しかし, 中間発表と比べて声が大きく聞き取りやすかったなどの肯定的な意見もみられた. また, 発表内容についても中間発表と同様に, 否定的な意見が多かった. 理由として, 専門用語が理解しづらかった, ECM への理解が難しかったなどが考えられる.

以上のことから, 中間発表と比較して評価の平均点が上がったため, よりよい発表が行えたと考える. しかし, 専門用語や ECM についてわかりやすい説明を行う必要があると考えられる.

（文責: 富樫北斗）

3.6.3 発表

成果発表は, 12 月 6 日 (4 時限 5 時限) に, 3 階 E 工房前で行った. 発表回数は, 全 3 回で後半のみ行った. 1 回あたり 15 分の発表で, 10 分間プレゼンテーションを行い, 残り 5 分間質問を受けた. 人数の振り分けは 4 人全員がスライドを分担して発表を行った.

発表はスライドの説明をメインとし, ポスターにはスライドの内容をより詳しく説明したものを用意した. 成果発表での発表を行い, アンケートを集計した結果, 高評価と低評価が半々であった. 反省点として, 内容が分かりやすいと感じた人と分かりにくいと感じた人の両方の意見があったので, その差異をなくしていく必要があると考えられる.

（文責: 富樫北斗）

第 4 章　プロジェクト内のインターワーキング

- 下地健介（プロジェクトリーダー）

(1) ECM の基礎について学習した
(2) 中間発表に向けて評価アンケートの作成をした
(3) 中間発表に向けてスライドの作成を行った
(4) 中間発表のスライドの原稿を作成した
(5) 中間報告書の担当箇所について執筆した
(6) 既存の Web サイトの問題点について精査した
(7) HTML,CSS について学習した
(8) Web サイトの作成を行った
(9) 成果発表会のスライドの作成を行った
(10) 成果発表会のスライドの原稿を作成した
(11) 最終報告書の担当箇所について執筆した

（文責: 下地健介）

- 富樫北斗

(1) ECM の基礎について学習した
(2) 中間発表会に向けての準備を行った
(3) 中間発表に向けてポスターの作成を行った
(4) 中間発表に向けてスライドの作成を行った
(5) 中間発表のスライドの原稿を作成した
(6) 中間報告書の担当箇所について執筆した
(7) 既存の Web サイトの問題点について精査した
(8) Web サイトに載せるロゴを制作した
(9) 成果発表会のスライドの作成を行った
(10) 成果発表会のポスターの作成を行った
(11) 最終報告書の担当箇所について執筆した

（文責: 富樫北斗）

- 谷山真子

(1) ECM の基礎について学習した
(2) 中間発表に向けてポスターの作成を行った
(3) 中間発表のアンケートを回収, 分析を行った
(4) 中間報告書の担当箇所について執筆した
(5) 既存の Web サイトの問題点について精査した

(6) HTML,CSS について学習した
(7) Web サイトの作成を行った
(8) Web サイトに載せるイラストを Adobe Illustrator を用いて作成した
(9) 成果発表会のポスターの一部を作成した
(10) 成果発表会のスライドの担当部分を作成した
(11) 最終報告書の担当箇所について執筆した

（文責: 谷山真子）

- 松崎伸彦

(1) ECM の基礎について学習した
(2) 昨年度のプログラムを運用した
(3) STUDIO KAMADA に素因数分解の報告をした
(4) PARI/GP について学習した
(5) PARI/GP で指定した 2 つの数字の間に存在する素数を数えるプログラムを作成した
(6) PARI/GP で指定した数字までに存在する 2 つの素数の差を順に出力するプログラムを作成した
(7) PARI/GP で 1 から B_1 までの数字すべての最小公倍数を求めるプログラムを作成した
(8) PARI/GP で ECM のプログラムを作成した
(9) 昨年度のプログラムと PARI/GP のプログラムで速度比較を行った
(10) Baby-step Giant-step を実装した
(11) 素数テーブルを作成するためのプログラムを作成した
(12) 中間報告書の担当箇所について執筆した
(13) Web サイトの文章を一部作成した
(14) 成果発表会のスライドを一部作成した
(15) 最終報告書の担当箇所について執筆した

（文責: 松崎伸彦）

第 5 章　活動結果

本プロジェクトでは, 前期は全員で ECM の学習を初め, ECM の概要を理解した. その後は PARI/GP での実装や昨年度のプログラムの運用に取り組んだ. 後期は Stage 2 の実装, *Baby-step Giant-step*, 素数テーブル実装による Stage 2 の改良, Web ページの作成に取り組んだ.

（文責: 下地健介）

5.1　プロジェクトの成果

5.1.1　プログラム班

PARI/GP では Stage 1 や Stage2 の実装を行った. Stage 1 ではスカラー値として B_1 の階乗ではなく 1 から B_1 の最小公倍数を採用した. Stage2 では *Baby-step Giant-step* を実装した. また, *Baby-step Giant-step* の高速化のために素数テーブルを実装した.　また, B_1 を 3000000 とし, PARI/GP の試行回数を 200 として平均を取り, funecm を最速の場合の平均とした時に速度比較を行った結果, Stage 1 のみでの実行時間では funecm よりも速かった. この時 PARI/GP の実行環境を Core i3-7100U とし, 昨年度のプログラムの実行環境を Xeon E5-2640 とした.

そして, 次に B_1 を 4572523 とし, 試行回数を 100 として平均を取り, 速度比較を行った結果, Stage 1 の実行時間は funecm よりも速かった. しかし, Stage 2 では遅くなった. この時 PARI/GP の実行環境を Core i3-7100U とし, 昨年度のプログラムの実行環境を Xeon E5-2640 とした. 次に実行環境をどちらとも Xeon E5-2640 としたところ, Stage 1 では funecm より遅くなり, Stage 2 では速くなった.　そして, STUDIO KAMADA に載っている合成数 $\frac{(26\times10^{189}-71)}{9}$ の素因数分解に成功した. 素因数分解を行う前は以下のようになっていた. 括弧内の値は桁数を示す.

$149 \times 82484045852903 < 14 > \times 785112311669773 < 15 > \times$
$29939381713135627933606580026414583060193269180371491954502028349323946037260584$
$19696015605412401386075047400441066132226058524561306823269920850294409587305 51$
$< 159 >$

159 桁の部分は合成数であるため, この部分の素因数分解を行った. 昨年度のプログラムを実行した結果, 以下のように素因数分解を行うことができた.

$726994173236431055598202555147195053651075 23 < 44 > \times$
$41182423209599541236032772952263752805172941019494643249743505499380706241248137$
$32022125598731354966984604402141437 < 115 >$

プログラムの実行は VPS で行った. また, プログラムのパラメータ B_1, B_2 は ECMNET の Optimal Parameter を参考に素因数の桁数が 40 桁に対応したものにした.

（文責: 松崎伸彦）

5.1.2 Web 班

既存の Web サイトをもとに不要な情報を削除, 追加する情報の決定を行った. 次に, 対象者を誰にするか, どのような Web サイトを目指すかを決めた. その後, 実装に向けワイヤーフレームを作成し, レイアウトを確認した. 最後に, カテゴリの分担を行った. 下地が HOME と暗号, 谷山が楕円曲線法 (ECM) とその他を担当した. 活動内容と Link の作成は 2 人で行った.

（文責: 谷山真子）

5.2 成果の評価

PARI/GP での Stage 1, Stage 2, そして *Baby-step Giant-step*, 素数テーブルの実装が完了した. その結果, ランダムな 10 桁の素数同士の積である合成数の因数の発見に成功し, 更に昨年度のプログラムにより STUDIO KAMADA に載っている合成数の分解に成功した. よって目的であった PARI/GP での実装及び昨年度のプログラムを用いて STUDIO KAMADA に名前を載せるということは達成できた. しかしこの PARI/GP のプログラムでは, 未だに STUDIO KAMADA や ECMNET に載っている合成数を分解していない. 加えて, 昨年度のプログラムの改良を一切していない. 一方, STUDIO KAMADA に名前を載せることができたが現時点では 1 つの合成数しか分解できていない. そして, ECMNET にはまだ挑戦していない. よって, 昨年度のプログラムの改良, PARI/GP の改良, PARI/GP や昨年度のプログラムを用いての STUDIO KAMADA への更なるランクイン, ECMNET へのランクインが今後の課題である.

（文責: 下地健介）

第 6 章　まとめ

6.1　前期活動の成果

前期は全員で ECM の学習から始め, ECM の概要を理解した. その後は PARI/GP での実装や昨年度のプログラムの運用に取り組んだ.

PARI/GP では Stage 1 の実装を行った. また効率を良くするための改良を行い, 高速化に関しての考察を行った. そして, 昨年度のプログラムを運用し, STUDIO KAMADA に載っている 1 つの合成数の分解に成功した.

（文責: 谷山真子）

6.2　後期の展望

PARI/GP でプログラムを作成し, PARI/GP の実行環境を Core i3-7100U とし, 昨年度のプログラムの実行環境を Xeon E5-2640 とした際に, Stage 1 では funecm よりも速い実行時間となった. しかし, 試行回数は 200 回と少ないので更なる試行を重ねて検証することが重要である. そして Stage 2 の実装はまだ未完成であるので, 実装することが重要である. 加えて, Stage 1 のみで, ある確率での合成数の因数を発見できたがその確率は調べていないため, Stage 1 での素因数の発見確率を調べることが今後の課題である. また, 現在このプログラムは PC 上で動かしているため, PC を起動していないとプログラムが動かない. 目標としては, PARI/GP の更なる高速化と Xeon サーバ上で動かすことが挙げられる.

加えて, 昨年度のプログラムを用いて STUDIO KAMADA に名前を載せることができたが, ECMNET にはまだ名前が載っていないため, ECMNET の合成数に対する素因数分解も行っていく. そして昨年度のプログラムの改良は未だ行っていないため, プログラムの改善点を探し, 改良を行うことが今後の課題である.

以上より, 昨年度のプログラムの改良, Stage 1 の性能の確認, Stage 2 の実装, プログラムの更なる高速化とサーバー上でのプログラムの動作, ECMNET に名前を載せるということが今後の展望である.

（文責: 松崎伸彦）

6.3　後期活動の成果

6.3.1　プログラム班

後期は昨年度のプログラムを運用しつつ, 楕円曲線法の改良について調査した. その後は PARI/GP での改良を行った.　PARI/GP では Stage2 の実装を行い, *Baby-step Giant-step* を実装することで改良した.

また PARI/GP のプログラムをサーバ上に移したことで, 起動している PC に負荷がかからなく

なった.

（文責: 松崎伸彦）

6.3.2 Web 班

Web 班の後期活動の成果は既存の Web サイトを閲覧し, どのような内容にするかを話し合った. それに基づいて, 新しく作成する Web サイトのカテゴリを決めた. 次に, Web サイトの全体のレイアウトをワイヤーフレームで作成した. 次に, HTML と CSS の学習を開始した. その後, Ameba Ownd を用いて Web サイトを作成しながら, ロゴの制作に取り組んだ. 後期から活動を始めたため, Web サイトを作成する時間が限られていた. 互いに協力し合うことによって, 毎週着実に進歩することができた. 操作しやすく, 高校生が理解しやすいような Web サイトの作成はほぼ完成できたと考えている.

（文責: 谷山真子）

6.4 今後の展望

6.4.1 プログラム班

PARI/GP で Stage 1, Stage 2 の実装が終わり, *Baby-step Giant-step* の実装も完了した. Stage 1, Stage 2 を実装した結果, ランダムな 10 桁の素数同士の積である合成数を分解することができた. 異なる環境下では Stage 2 では PARI/GP のプログラムの方が遅かったものの, Stage 1 では速かった. また同じ環境下では PARI/GP のプログラムの方が Stage 1 について遅くなったが, Stage 2 は速くなった. このように様々な場合について速度比較を行った. ランダムな 10 桁の素数同士の積である合成数は分解できたが, STUDIO KAMADA にあるような長大桁整数の分解は未だ出来ていない. また, 昨年度のプログラムを用いた素因数分解は今年度は未だ ECMNET のランクインは出来ていない. そして, Stage 2 の改良法としては誕生日のパラドックスがあるため, まだまだ高速化の余地はあると予測される. よって, 今後の課題としては PARI/GP のプログラムによる STUDIO KAMADA 及び ECMNET のランクイン, 昨年度のプログラムの ECMNET のランクイン, 更なるプログラムの高速化が挙げられる.

（文責: 松崎伸彦）

6.4.2 Web 班

最終成果発表会までに Web サイトはほぼ完成したが, 改善点やできなかったことがいくつかある. はじめに完成した Web サイトをサーバにアップロードし, 多くの人に閲覧してもらえる環境を作ることである. また, 内容に関しては数式をできるだけ少なくしたため, 楕円曲線法の説明を言葉ですることとなった. その文章が長くなってしまったので, もう少し簡潔にまとめる. さらに, 実際に高校生に見てもらう機会を設け, 見やすさや理解のしやすさを確かめることが挙げられる.

（文責: 谷山真子）

参考文献

[1] ECMNET. https://members.loria.fr/PZimmermann/ecmnet/ (最終アクセス 2019 年 7 月 22 日)

[2] STUDIO KAMADA. https://stdkmd.net/ (最終アクセス 2019 年 7 月 22 日)

[3] 伊豆哲也. 楕円曲線暗号入門. https://researchmap.jp/mulzrkzae-42427/ (最終アクセス 2019 年 7 月 22 日)

[4] Joseph H. Silverman and Jon Tate. 楕円曲線論入門第 2 版. 丸善出版, 2012.