FUN-ECM プロジェクト
FUN-ECM Project

1013186　山本 健太　Kenta Yamamoto

1　プロジェクトの目的

本プロジェクトの目的は、楕円曲線法を並列処理できるプログラムを作成し、大きな整数の素因数を見つけ、ECMNET にランクインすることである。現在のインターネットの通信において、公開鍵暗号の 1 つである RSA 暗号が広く用いられている。RSA 暗号は、巨大な数の素因数分解が困難であることを利用した暗号である。よって、RSA 暗号の安全性は、巨大な数の素因数分解の困難さで評価される。つまり、素因数分解のチャレンジは、重要なテーマとなっている。素因数分解の優れたアルゴリズムの 1 つに楕円曲線法がある。私たちは、楕円曲線法による素因数分解のプログラムの並列処理と高速化の実現を目指す。

また、ECMNET[1] という楕円曲線法によって大きな素因数を見つけることを目的とするコンペティションがある。現在記録されている素因数よりも大きな素因数を見つけることで、ランキングに名前を載せることができる。私たちは、このサイトのランキングに載っている素因数よりも大きな素因数を見つけたい。

2　課題

本プロジェクトの目的である ECMNET ランクインのためには大きな素因数を見つける ECM プログラムが必要となる。その ECM プログラムの作成にあたって以下の項目を課題として設定した。

- 楕円曲線法の理解
- ECM プログラムの高速化と並列化の実装
- ECM プログラムの評価

プロジェクト始動時にメンバ全員が楕円曲線法についての理解が乏しかった点から、前期では白勢先生・由良先生の指導のもと、楕円曲線法の基礎的な学習に取り組んだ。後期では課題達成のため高速化の手法の提案、実装、評価を主軸に活動を行った。

3　活動内容

3.1　前期

前期では課題の 1 つの楕円曲線法の理解を中心に活動を行った。担当教員の白勢先生・由良先生に作成していただいた授業計画に従い、楕円曲線法のアルゴリズムや基礎知識などを学んだ。具体的な内容は以下の通りである。

- 楕円曲線法の基本
- 剰余について
- Z/nZ の四則演算
- ラグランジュの定理と点の位数
- 無限遠点と因数発見の関連について

また、学習した内容をもとに PARI/GP[2] を用いて簡単な ECM プログラムを作成した。今回作成したプログラムは学習してきた内容が実際にどのように実装されているのか確認するためのプログラムである。よって大きな素因数を見つけることができなかった。

後期では、残り 2 つの課題、ECM プログラムの高速化と並列化の実装と評価の解決のために、プロジェクト全体を射影班、プログラム班、統計班の 3 班に分け活動を進めた。

3.2　射影班

3.2.1　活動目的

射影班の活動目的は、ECM プログラム内の加算公式部分に射影変換を用いて計算量を削減することである。具体的には今年度から採用したエドワーズ曲線を利用した ECM プログラム内の加算公式、2 倍算公式の中の計算コストを削減することを目標とした。

3.2.2　活動内容

まず昨年度のプログラムの計算コストを削減するために使用した射影変換に着目した。今年度のプログラムにも採用するためにまずは射影変換を書籍、論文、白勢先生の講義等で学習した。その後、各計算の計算コストを算出し、昨年度と今年度の ECM プログラム内の加算公式の計算コストを調べた。

3.2.3　活動成果

乗算を基準として各計算の計算コストを算出した際に、各計算の計算コストは乗算を基準とした際に、二乗算は 0.8 倍、逆元計算は 27.5 倍であることが判明した。また、昨年度と今年度の加算公式一回分の計算コストの比較は以下のようになった。表記の際は乗算を M、二乗算を S、逆元計算を I とした。

- 昨年度射影変換前 : I+2M+S ≒ 30.3M
- 今年度射影変換前 : 2I+6M ≒ 61M
- 昨年度射影変換後 : 12M+2S ≒ 13.6M
- 今年度射影変換後 : 11M+S ≒ 11.8M

これらを全て M に統一して比較すると上記より、昨年度射影変換前が 30.3M、今年度射影変換前が 61M、昨年度射影変換後が 13.6M、今年度射影変換後が 11.8M となった。これらのことから射影変換前の計算コストに比べ射影変換後の計算コスト、昨年度の射影変換後のプログラムの計算コストに比べ今年度の射影変換後の計算コストがそれぞれ少なくなっていることがわかった。

3.3　プログラム班

3.3.1　活動目的

プログラム班の活動目標は 2 つあった。1 つ目は、昨年度の ECM プログラムよりも高速に処理ができる ECM プログラムの実装を行うことであった。ECM プログラムを用いて大きな桁数の素因数を見つけようとした場合、プログラムを長期にわたって動かし続ける必要がある。また、ECM プログラムは、加算公式の計算処理を繰り返すことで素因数分解を行っている。よって、加算公式の計算処理コストを減らすことで、ECM プログラムの処理が高速になり、大きな桁数の素因数発見の効率化を図れると考えた。そこで、プログラム班は昨年度のプログラムの加算公式の計算処理部分を中心に改良を行うことした。具体的には、後期に理論班が作成した射影座標系を導入したエドワーズ曲線を用いた楕円曲線法のアルゴリズムを基に、昨年度の ECM プログラムの改良を行った。2 つ目は、ECM プログラムを XeonPhi 実行するための簡単なマニュアルの作成を行うことであった。ECM プログラムは C 言語で記述したが、OpenMP を導入し XeonPhi 上で並列処理できるようなものにした。よって、XeonPhi 上で ECM プログラムを実行するために、いくつかの特殊な操作を行う必要があった。そこで、特殊な操作の手順をマニュアルにすることで、誰でも XeonPhi 上で ECM プログラムを動かせるようにした。

3.3.2　活動内容

目的達成のために大きく分けて 5 つの活動を行った。1 つ目は GMP の学習である。GMP とは任意精度演算を行える算術ライブラリである。GMP を用いることで、計算の桁数がハードウェアのメモリ容量以外に制限されなくなるため、楕円曲線法のような非常に大きな桁数を扱う計算に適している。インストールや使用方法などを GMP のマニュアルを読み、小規模のプログラムを作成しながら学んでいった。2 つ目は並列処理である。楕円曲線法で大きな桁数の数の素因数分解を行うと処理に莫大な時間がかかる。そこで、並列実行して素因数分解の処理の高速化、効率化を行った。並列処理部分に、OpenMP を用いたプログラムを Xeon Phi 上で実行した。3 つ目はプログラム制作である。アルゴリズム担当とインターフェース担当に分けコーディング、テスト、実装を行った。不具合に適切に対応できるよう気を付けるなど工夫しながらプログラム作成を行った。4 つ目は運用である。効率的に Xeon Phi を使って運用するために、オンライン作業できるような設定をした。5 つ目はマニュアル作成である。Xeon Phi の操作は複雑な手順を要するため、他のプロジェクトメンバや次のプロジェクトのためにわかりやすいマニュアルを作成した。

3.3.3　活動成果

エドワーズ曲線の加算公式に射影変換を適用し、Xeon Phi 上で並列処理できるようなプログラムを完成した。そして、昨年度と今年度のプログラムの比較を行った結果、昨年度よりも 15% の高速化ができた。また、プログラム班以外でも簡単に利用できるような簡単なマニュアルの作成もした。

3.4　統計班

3.4.1　活動目的

統計班は、射影班によって取り入れた理論、プログラム班に実装された機能によりプログラムがどれほど高速化したのかの検証を目的とした。昨年度の ECM プログラムと今年度作成したプログラムの大きな違いは、楕円曲線の形式と射影変換式を変更したことである。この違いがプログラムにどのような効果をもたらすのかを知ることで、今後のプログラムの改良に役立てることができる。

3.4.2　活動内容

どのようにして高速化を検証するのかを話し合い、今回はプログラムの処理性能の評価をしたいと考えた。合成数を入力したときの出力までの時間は素因数の大きさなどに左右されるため正確な処理速度の測定には適さない。作成した ECM プログラムは、素数を入力することによってプログラムのすべ

ての処理を終了するまで時間を計測することができるため素数を入力することにした。実際のプログラムには素数判定の機能が組み込まれているが今回はプログラム班に依頼してその機能を削除して実測に臨んだ。また、処理速度のデータの信頼性を得るためいくつかの統計方法を模索し実行した。

3.4.3　統計方法

今年度と昨年度の ECM プログラムの性能を計る為にそれぞれ 5 ～ 40 桁の素数を入力として実行時間を計測した。各桁の素数はその桁の素数をランダムに抽出するプログラムを作成した。a 桁の素数を抽出する時、その素数の範囲は $[10^{a-1}, 10^{a-1}+10^{\lceil \frac{a-1}{2} \rceil}]$ とした。これによって抽出された素数を用いて計測に使用した。各桁の計測数は今回は 4 人で計測をしたが、1 人 4 回ずつの計 16 回計測した。私たちは今回データを計測するにあたり、4 人で同じ性能の 4 台のパソコンを使用した。パソコンの性能は以下の通りである。

- ホスト OS ： Windows8.1
- ゲスト OS ： CentOS6.3
- プロセッサ ： Intel(R) Core(TM)i3-3120M CPU @2.50Ghz 2.50Ghz
- 実装メモリ (RAM) ： 8.00GB
- システムの種類 ： 64 ビット オペレーティングシステム x64、 ベースプロセッサ

3.4.4　活動成果

今年度と昨年度のプログラムの処理速度を計測した結果、桁数が 10 桁の時の今年度のプログラムの処理速度は 43.562 秒となっている。また、桁数が 40 桁の時の新プログラムの処理速度は 96.555 秒となり、桁数が 10 桁の時と比べると約 2.2 倍処理に時間がかかっている。向上率が最大になったのは桁数が 10 桁の時で、約 18% も処理速度が速くなった。各桁数の向上率の平均を出したところ、約 15% 処理速度が向上した。

射影班が導入した射影変換により今年度の ECM プログラムは昨年度使用した ECM プログラムよりも計算コストが約 15% 削減することに成功したことが分かっている。つまりこの結果は、射影変換によって削減された計算コストが理論通りに処理速度を向上させたことを示している。

3.5　発表

3.5.1　中間発表

中間発表では楕円曲線法の基本的なアルゴリズムなどを説明し、PARI/GP を使った ECM プログラムの実演などを披露した。ホワイトボードを使ってアルゴリズムの流れを説明するなどの工夫も行った。楕円曲線法についての説明は難解なため、聴講者にできるだけわかりやすくするために時間をかけて丁寧に説明した。また、スライドには Prezi[3] を使い聴講者が飽きないような動きのあるスライドを作成した。ポスターは一目で内容が伝わるように見やすい配置などを心掛けた。本番当日ではプロジェクターの不具合などに見舞われたが、準備してきた内容は発表することができた。聴講者の評価は発表技術・発表内容ともに約 6 点だった。専門的な内容は理解が難しいということが聴講者のコメントでよく理解できた。

3.5.2　最終発表

最終発表では、前期の反省を踏まえて聴講者にとって理解が難しいような専門知識や理論の内容は減らした。代わりに、聴講者に伝えたい本プロジェクトの成果の射影変換を用いて計算コストを 15% 削減したこと、処理速度を 15% 向上したということについて詳しく説明をした。スライドはグラフや表を載せるなど見やすくわかりやすいものにした。ポスターではプロジェクト全体について 1 枚、班ごとの活動について 2 枚の計 3 枚作成した。発表と併せて読むことで、より理解してもらえるよう専門的な内容をわかりやすくし画像とともに説明をした。聴講者の評価は発表技術・発表内容ともに 7 点で、前期の評価を共に上回ることができた。明確でわかりやすかったという聴講者からのコメントが多かったため、こちらの狙い通りの発表ができていたと判断できる。

4　成果

4.1　射影班

射影班は射影変換に関する学習を行い、各計算の計算コストを C 言語環境下で各計算を 1 億回行うプログラムを作成し、そのプログラムをそれぞれ動かし調べて比較した。また射影変換前後の計算コストの算出を行った。今年度のプログラムに射影変換を用いた結果、昨年度の完成したプログラムの加算公式一回の計算コストが 13.6M だったのに対し、今年度のプログラムの加算公式一回の計算コストが 11.8M だったことから、理論上プログラムを約 15% の高速化をすることに成功したと考えられる。

4.2　プログラム班

プログラム班は初めにコーディングを担当する箇所を振り分けることでソースコードの共有における食い違いを防いだ。また、コーディングの際には今後の改良やメンテナンスがしやすいよう書くことを心掛けた。こうして新しいアルゴリズム

を適用した素因数分解プログラムを完成させた。主な改良点は新アルゴリズムによる高速化、ループ実装による効率化である。また、学内ネットワークを通じてプログラムを実行できるようにしたことで効率的な素因数探しを行えるようになった。

4.3 統計班

昨年度のプログラムと今年度のプログラムを比較する為に、5 ～ 40 桁の素数を 5 桁毎に 16 個ずつ入力し、プログラム内部で 1 万回計算を行った時の処理速度の結果を図と表に出力した。その結果、昨年度のプログラムよりも、今年度のプログラムの方が約 15% 処理速度が向上した。射影班が算出した、昨年度のプログラムと今年度のプログラムの計算コストを比較した時に、計算コストが今年度のプログラムの方が 15% 削減されていたことがわかる。よって、射影変換によって、理論通りに計算処理速度が向上したことが言える。

5 まとめ

本プロジェクトの今年度の目的は楕円曲線を理解し、ECMNET へランクインすることであった。目的達成のために楕円曲線法の理解、 ECM プログラムの高速化と並列化の実装、ECM プログラムの性能評価の 3 つを主軸において活動を進め、以下のことを達成した。1 つ目は前期に基礎学習を行い楕円曲線への理解を深めた。2 つ目は従来のヴァイエルシュトラス方程式からエドワーズ曲線に変更し射影変換を導入することで計算コストを約 15% 削減し、Xeon Phi 上で並列化処理を施した ECM プログラムを完成させた。3 つ目は昨年度と今年度の ECM プログラムを比較し、約 15% 処理速度を向上したことを実証した。

6 反省と展望

基本的な理論の学習、プログラムの学習、発表の準備などが主な活動だったため、他の活動に手が回らなかった。来年度のプロジェクト学習に向け、今後の反省と展望を以下に示す。

1 つはステージ 2 の実装である。素因数分解を 2 つのステージに分けて行う方法があり、その場合、今年度行っている ECM はステージ 1 で使用される。ステージ 2 では、誕生日攻撃法を利用したロー法による素因数分解を行う。まず、ステージ 1 で素因数分解に失敗した場合、座標が算出される。その座標をステージ 2 で使用することで、素因数分解が成功する場合がある。それにより、素因数分解が成功する確率を高めることができる。今年度の活動では手を付けることができなかったため、今後の活動ではステージ 2 のプログラムを実装し、ステージ 1 と連携をさせたい。

2 つ目は論文の輪読である。論文を読む時間があまりとれなかったため、特に楕円曲線法に関する論文をプロジェクトメンバ全員で輪読し、理解を深めたい。

3 つ目はプログラムの試行回数である。プログラムを実際に動かす時間があまりとれなかったため、プログラムの動作回数を稼ぎ、目標とする桁数の素因数分解など、ECMNET への登録に向けた活動をしたい。

4 つ目は社会との関連性である。成果発表会で何のために素因数分解をするのかわからないという意見が多かったため、楕円曲線法や素因数分解がどのように現代社会に関係しているかをよく学びたい。また、プロジェクト学習を行う背景や概要などに関係することについて簡潔にまとめ、発表する。特に、公開鍵暗号や RSA 暗号は素因数分解と密接に関係しているため、成果発表会でプロジェクト学習との関連性を示せるようにしたい。

参考文献

[1] ECMNET
`http://www.loria.fr/~zimmerma/records/ecmnet.html` (最終アクセス 2016 年 1 月 5 日)

[2] PARI/GP Development Headquarters
`http://pari.math.u-bordeaux.fr/` (最終アクセス 2015 年 12 月 11 日)

[3] プレゼンソフト — オンラインプレゼンツール — Prezi
`https://prezi.com/` (最終アクセス 2015 年 12 月 11 日)