1D-01　FFTによるモジュラー多項式の計算

堀 幸雄
神奈川大学理学研究科情報科学専攻
EMail: *horiyuki@goto.info.kanagawa-u.ac.jp*

1 はじめに

高速フーリエ変換 (Fast Fourier Transform:FFT) は科学技術計算において広く用いられるアルゴリズムである．係数が膨大な多項式の乗算を数多く繰り返すモジュラー多項式の計算において計算の高速化を考える上で有効なアルゴリズム [1] である．大規模な桁数を必要とする計算，π や e のような数学定数の高精度計算において高速に計算するのに広く用いられている [2].

またモジュラー多項式は Schoof による群位数計算する多項式時間アルゴリズム [3] の中で重要な役割を果たしている．楕円曲線上の群位数は重要なパラメタであり，安全な楕円曲線暗号を作成する上で高速に計算されなくてはならない [4].

本稿ではモジュラー多項式の計算において基礎となる乗算アルゴリズムについて FFT アルゴリズム，Karatsuba 法などの計算 [5] の結果について比較し報告する．

2 モジュラー多項式

モジュラー多項式とは，楕円モジュラー関数 $j(z)$ と素数 l に対して以下の式で定義される多項式である．

$$\begin{aligned}\Phi_l(X,j) &= (X-j(lz))\prod_{k=0}^{l-1}\left(X-j\left(\frac{z+k}{l}\right)\right)\\ &= X^{l+1}+j^{l+1}+\sum_{m,n=0}^{l}a_{mn}X^m j^n\end{aligned}$$

これは 2 変数の整数係数多項式であり，X, j の対称式となるここでモジュラー多項式は X の 1 変数多項式となり次のように表される．

$$\Phi_l(X,j)=X^{l+1}+\sum_{k=1}^{l+1}(-1)^k s_k(j)X^{l-k+1}$$

$$J_l=\left\{j(lz), j\left(\frac{z}{l}\right), j\left(\frac{z+l}{l}\right),\ldots, j\left(\frac{z+l-1}{l}\right)\right\}$$

と見なすこともできる．定義に従い普通に計算していくと係数の膨張が激しく Φ_l を求めるのが困難である．

3 FFT アルゴリズム

良く知られているように n 桁の乗算計算は通常の筆算方法では n^2 回の基本演算が必要になり，多倍長の計算では非常に遅くなる．しかし FFT アルゴリズム を利用して計算すると $O(nlog(n))$ ですむ．

FFT による x と y の乗算概略を簡単に述べれる．体の上でのフーリエ変換であり，ある数についてモデュロ計算でフーリエ変換を行ない，その結果の畳み込み (Convolution Product) をして，求める結果を構築する．これはフーリエ変換を行なった結果の畳み込みは，通常の乗算結果をフーリエ変換したものに等しいという事実を利用する．

1. x, y の各桁を数列 x_i, y_i とおいて，それに対する離散フーリエ変換 X_i, X_i を FFT で計算する．
2. $Z_i = A_i * B_i$ を計算して Z_i に対して逆 FFT を行い畳み込み z_i を得る．
3. z_i を元の法にし乗算 z を得る．

ここで FFT による畳み込みでは，巡回しないような処理が必要となる．

Calculate modular polynomial by FFT algorithm
Yukio HORi
Department of Information Science, Faculty of Science, Kanagawa University

4 モジュラー多項式の計算

計算は定義式に従い，$j(q)$ が必要な次数まで求められているとし，J_l の各元を求め，

$$\begin{aligned} j(lz) &= \frac{1}{q^l} + c_0 + c_1 q^1_+ \cdots \\ j\left(\frac{z+k}{l}\right) &= \frac{1}{\zeta^k q^{1/l}} + c_0 + c_1 \zeta^k q^{1/l} + \cdots \end{aligned}$$

$1 \leq k \leq l-1, \zeta$ は 1 の l 乗根である．

ここで $s_k(j)$ が対称式で表わされるので，対称式に関する Newton の公式を利用する．

J_l の元のうち $j(lz)$ だけ他の元と性質が異なるので，J_l から $j(lz)$ を除いた集合を

$$\overline{J_l} = \left\{ j\left(\frac{z}{l}\right), \left(\frac{z+l}{l}\right), \ldots, j\left(\frac{z+l-1}{l}\right) \right\}$$

$\overline{J_l}$ による対称式を

$$t_l = j\left(\frac{z}{l}\right), \left(\frac{z+l}{l}\right), \ldots, j\left(\frac{z+l-1}{l}\right)$$

ただし $t_0 = 1, t_{l+1} = 0$．このとき s_k と t_k で以下のように表わされる．

$$s_k = j(lz) t_{k-1} + t_k$$

ここで，$j^r(z)$,$\overline{J_l}$ の元の r 乗和 u_k を以下のように置く．

$$\begin{aligned} j^r(z) &= \sum_{n=-r}^{\infty} c_n^{(r)} q^n \\ u_r &= \sum_{k=0}^{l-1} j^k \left(\frac{z+k}{l}\right) \quad (1 \leq r \leq l) \end{aligned}$$

このとき Newton の公式より以下の式が成立する．

$$\begin{aligned} u_r &= l \sum_{n=0}^{\infty} c_{ln}^{(r)} q^r \quad (1 \leq k \leq l-1) \\ u_l &= l\left(\frac{1}{q} + \sum_{n=0}^{\infty} c_{ln}^{(l)} q^n\right) \\ t_{l-1} &= -\frac{1}{l-1}(u_{l-1} - t_1 u_{l-1} + \ldots - t_{l-2} u_1) \end{aligned}$$

よってモジュラー多項式の計算は，実際には $j^r(q) \quad (r = 1, 2, \ldots, l+1), u_r, t_k, s_k, \Phi_l(X, j)$ の順に求め，計算していく．

5 評価比較

以下に $l = 73$ までのモジュラー多項式について Karatsuba 法と FFT で求めた評価図を示す．

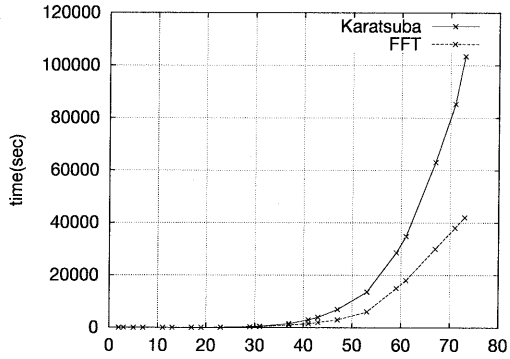

図 1:FFT と Karatsuba 法での評価結果

図 1 のように FFT アルゴリズムを用いた場合には高い性能が得られることが確認した．これはモジュラー多項式はべき乗計算，係数の膨大な多項式の乗法を繰り返しているためである．また複数計算機を用いた並列計算であるが，今回は実験は行なわなかったがこれは分割可能な問題あるゆえ確実な効果が見込める．

6 まとめ

本稿では，モジュラー多項式の計算において乗算アルゴリズムの評価を行なった．今後はいくつかの基底，6 基底，8 基底 [6]，12 基底，16 基底における実現と分析，評価が課題である．

参考文献

[1] D.E. Knuth: 準数値算法 4, 中川圭介訳, サイエンス社

[2] 金田康正, 田村良明: 円周率-高速計算法と統計性 (2)-, 情報処理学会第 28 回プログラミングシンポジウム報告集, 1987 年 1 月, pp.251-262

[3] Schoof,R.: Elliptic curves over finite fields and the computation of square roots mod p, *Math. Comp.* 44 (1985) pp.483-494

[4] T. Izu, J. Kogure, M.Noro, K. Yokoyama: "Efficient Implementation of Schoof's Algorithm", Advances in Cryptology - *ASIACRYPT'98 Lecture Note in Computer Science*, 1514, pp. 66-79, 1998.

[5] 伊豆哲也: Risa/Asir による modular polynomial の計算, 京都大学数理解析研究所講研録,「数式処理における理論と応用の研究」1997

[6] 高橋大介 金田康正: 積和演算命令に向いた 8 基底 FFT カーネルの提案, 情報処理学会論文誌 Vol. 41 No. 7 pp. 2018-2026 July 2000