数理解析研究所講究録
第1568巻 2007年 162-168

Niederreiterアルゴリズムとその実装

山中亜希子
AKIKO YAMANAKA
神戸大学総合人間科学科*

長坂耕作
KOSAKU NAGASAKA
神戸大学発達科学部

1 Niederreiterアルゴリズムのサーベイ

今回取り上げたのは，有限体上の因数分解アルゴリズムの1つであるNiederreiterアルゴリズムである．有限体上の因数分解法としてはBerlekampアルゴリズムがよく知られて用いられているが，それとは違う方法で，似たような行列を用いる因数分解法としてNiederreiterアルゴリズムは紹介されている．

Niederreiterの論文は文献リストにあるように多数見られる．後のほうの論文ではNiederreiterの改良点などが述べられている．よって，すべてを読まなくてもアルゴリズムの理解は可能である．後に述べるが，このアルゴリズムは標数が小さいときに効率よく働くが，大きくなるとBerlekampのほうが早く動く．

1.1 Niederreiterアルゴリズムの概要

この因数分解法では有利関数体上の微分方程式 $y^{(p-1)}+y^p=0$ を用いる．これを線形化して作られるベクトル空間 (2.1節で述べる) に対し，Berlekampアルゴリズムのように行列 (2.1.1節で述べる) の零空間を求め，そこで見つかった因子とのGCDを取ること (2.1.2節で述べる) で因数分解を行う．この因数分解法はBerlekampとは違い無平方でも因数分解可能であり，行列の構築がBerlekampの行列に比べ，元の多項式から簡単に構築できる．また，この方法は標数が小さいときに有効である．次の節ではNiederreiterの因数分解法について，詳しく説明する．

2 Niederreiter因数分解法

2.1 Niederreiter因数分解法の詳細

$\mathbf{F}_p$ を標数が素数 p である有限体，$f\in\mathbf{F}_p[x]$ を無平方でモニックな次数 $d\geq 1$ の多項式とする．また，f は $\mathbf{F}_p$ 上で．$f=g_1\cdots g_m(g_1,\cdots,g_m\in\mathbf{F}_p[x])$ に因数分解されるとする．f の自明でない因子を見つけるため，有理関数体 $\mathbf{F}_p(x)$ での次の $p-1$ 階の微分方程式を用いる．

$$y^{(p-1)}+y^p=0 \tag{2.1}$$

$L(y)=y^{(p-1)}+y^p$ は $\mathbf{F}_p$ 上のベクトル空間 $\mathbf{F}_p(x)$ での線形演算子である．よって (2.1) 式の解は $\mathbf{F}_p(x)$ の線形部分空間を形成する．因数分解のためには，$\mathbf{F}_p(x)$ の部分空間をなす $y=h/f\,(h\in\mathbf{F}_p[x])$ なる (2.1) 式の解を求める．

*065f739f@stu.kobe-u.ac.jp

ここで，$h(x) \in \mathbb{F}_p[x]$ とすると，(2.1) 式は次のように書ける．

$$f^p \left(\frac{h}{f}\right)^{(p-1)} = -h^p \tag{2.2}$$

Theorem 1 ([13, theorem1]) 分母を $f = g_1 \cdots g_m$ に固定するとき，(2.2) 式の解は次の式で与えられる．

$$y = \sum_{i=1}^{m} c_i \frac{g_i'}{g_i}, \quad c_1, \cdots, c_m \in \mathbb{F}_p$$

◁

両辺が次数 $(d-1)p$ 以下の $\mathbb{F}_p$ 上の多項式となっており，どちらも x^p の多項式となっている．$M_p(f)$ を (2.2) 式の左辺の $d \times d$ 係数行列とすると，(2.2) 式は h の係数ベクトル $\boldsymbol{h} \in \mathbb{F}_p^d$ を未知数とする次の線形方程式になる．

$$(M_p(f) + I_d)\boldsymbol{h}^T = \mathbf{0} \tag{2.3}$$

Theorem 2 ([13, theorem2])

$$rank(M_p(f) + I_d) = d - m$$

◁

この定理より，もし $rank(M_p(f)+I_d) = d-1$ ならば，f は $\mathbb{F}_p$ 上で既約となる．$rank(M_p(f)+I_d) \leq d-2$ と仮定すると，定理 1 より (2.2) 式の解 h は $c_i \in \mathbb{F}_p$ と $b_i = g_i' \frac{f}{g_i} \in \mathbb{F}_p[x] (1 \leq i \leq m)$ で次のように与えられる．

$$h = \sum_{i=1}^{m} c_i b_i$$

定理 2 での証明から $b_1, \cdots, b_m$ は解 h の空間の基底であることが導ける．解 h に対して，次のようにおく．

$$J(h) = \{1 \leq j \leq m : c_j = 0\},$$

$$h = \sum_{i=1, i \notin J(h)}^{m} c_i b_i = \sum c_i g_i' \frac{f}{g_i} = \left(\prod_{j \in J(h)} g_j\right) \sum_{i=1, i \notin J(h)}^{m} c_i g_i' \frac{f}{g_i \prod_{j \in J(h)} g_j}$$

この式から次の式が導かれ，$\gcd(f, h)$ の計算によってすべての f のモニックな因子が求まることがわかる．

$$\gcd(f, h) = \prod_{j \in J(h)} g_j$$

2.1.1 Niederreiter 行列の作成方法

(2.2) 式の係数行列 $M_p(f)$ は直接求めることもできるが，S.Jeong と Y.Park [8] や P.Fleichmann と P.Roelse [5] らにより効率的な作成方法が述べられている．ここでは，S.Jeong と Y.Park らによる方法を紹介する．まず，行列 $M_p(f)$ を構成するのに必要となる (2.1) 式の微分方程式は p が素数でないと使えないので，一般化するため，Hasse-Teichmüller derivatives に基づいた微分方程式を用いている．

$\mathbb{F}_q$ を q 個の要素を持つ有限体とする．q は素数の標数 p のべき乗である．それぞれの整数 $n \geq 0$ に対して，階数 n の Hasse-Teichmuller derivative $H^{(n)}$ は，F_q 上で x^{-1} を変数とした Laurent(ローラン) 級数の体 $\mathbb{F}_q((x^{-1}))$ で次のように定義されている．w は任意の整数である．

$$H^{(n)}(\sum_{i=w}^{\infty} c_i x^{-i}) = \sum_{i=w}^{\infty} \begin{pmatrix} -i \\ n \end{pmatrix} c_i x^{-i-n}$$

すると，(2.1) 式は有理関数体 $\mathbb{F}_q(x)$ 上の次の微分方程式となる．ここで，$\mathbb{F}_r$ が $\mathbb{F}_q$ の標数 r の部分体になるように，整数 $r > 1$ を制限する．例えば，$r = q$ 又は $r = p(\mathbb{F}_q$ の標数$)$ など．

$$H^{(r-1)}(y) = y^r \tag{2.4}$$

この式を用いて，$N_r(f)h^T = (h^{[r]})^T$ なる行列は次のように構築できる．$f^{p-1} = \sum_{i=0}^{(p-1)d} a_i x^i$ とする．

$$N_r(f) = \begin{pmatrix} a_{r-1} & \cdots & a_0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 \\ a_{2r-1} & \cdots & a_r & \cdots & a_0 & 0 & \cdots & 0 & \cdots & 0 \\ a_{3r-1} & \cdots & a_{2r} & \cdots & a_r & a_{r-1} & \cdots & 0 & \cdots & 0 \\ \cdot & \cdots & \cdot & \cdots & \cdot & \cdot & \cdots & \cdot & \cdots & \cdot \\ 0 & \cdots & 0 & \cdots & 0 & \cdot & \cdots & a_{(r-1)d} & \cdots & a_{(r-1)d-r} \\ 0 & \cdots & 0 & \cdots & 0 & \cdot & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

2.1 節の $M_p(f)$ と $N_r(f)$ との関係は $M_p(f) = -N_p(f)$ である．

2.1.2 Niederreiter アルゴリズムの後半のステップの効率化

初期の $\gcd(f, h)$ の取り方は，線形方程式の解となる p^m 個の多項式 $h \neq 0$ に対して，f や f の自明でない因子との GCD を計算することで求めていた．Niederreiter は [16] で次の定理をベースにした後半のステップの効率化を行っている．

Theorem 3 ([16, theorem2]) $f = g_1 \cdots g_m$ を定数でないモニックな多項式 f を $\mathbb{F}_q$ 上で因数分解したものとする．その時，分母を f に固定した $y = \frac{h}{f}$ の $\mathbb{F}_r$-空間の基底は $\{\frac{g_1'}{g_1}, \cdots, \frac{g_m'}{g_m}\}$ で与えられる． ◁

基底の変換を行うことにより，GCD 計算を高速化できる．元の基底は $B_0 = \left\{\frac{g_1'}{g_1}, \cdots, \frac{g_m'}{g_m}\right\}$ となっている．アルゴリズムの前段階で基底 $B = \left\{\frac{h_1}{f}, \cdots, \frac{h_m}{f}\right\}$ が求められている．B を簡約して基底 $B_1 = \left\{\frac{u_1}{v_1}, \cdots, \frac{u_m}{v_m}\right\}$ が得られるとする．基底 B_1 を用いることで GCD 計算の高速化を図る．

$f = g_1 \cdots g_m$ の定数でないモニックな因子 w が与えられているとする．$w = g_1 \cdots g_k \quad (1 \leq k \leq m)$ と書くことができる．Basic Splitting Step の目的は w の自明でない因子，または，w の既約性を証明することである．Basic Splitting Step は $\gcd(w, v_i) \quad (1 \leq i \leq m)$ の計算により開始する．GCD の 1 つが w の自明でない因子ならば目的を達したことになる．

そうでない場合は $\gcd(w, v_i) \in \{1, w\}$ を得る．集合 $I(w) = \{1 \leq i \leq m : w \mid v_i\}$ とする．$i \in I(w)$ と $\beta \in \mathbb{F}_r$ に対して次を考える．

$$\gcd(u_i + \beta w' \frac{v_i}{w}, v_i) \tag{2.5}$$

この GCD はいつも w の 1 つの因子である．もし，w が可約ならば $i \in I(w)$ と $\beta \in \mathbb{F}_r$ に対して，(2.5) は w の自明でない因子である．この方法により，GCD を取る回数が p^m 回から rm^2 回になる．

2.1.3 Niederreiter アルゴリズムのまとめ

Algorithm 1 Niederreiter 因数分解法 [13]

[入力] モニックな多項式 $f \in \mathbb{F}_p[x]$

[出力] f の互いに素な既約因子 $g_1, \cdots, g_m \in \mathbb{F}_p[x]$

Step 1. 行列 $N_p(f)$ の計算．$m = 1$ なら f を返して終了．

Step 2. 線形方程式 $(N_p(f) - I)\boldsymbol{h}^T = 0$ を解く．解 $\boldsymbol{h}$ から (2.2) 式を満たす p^m 個の多項式 h を構成する．

Step 3. $h \neq 0$ を取り出し，$\gcd(f, h) \neq 1$ となるまで計算する．そこで出てきた $\gcd(f, h)$ が f の自明でない因子となる．既約因子を求めるためにはそれぞれの因子に対して $h \neq 0$ との GCD 計算を繰り返す．

◁

アルゴリズム 1 のステップ 3 を改良したものが次のアルゴリズムである．

Algorithm 2 Basic Splitting Step(GCD 計算)[19]

[入力] f の因子 w

[出力] w の既約因子

Step 1. $\gcd(w, v_i)$ の計算．w の自明でない因子なら w を分割し，それぞれに対して Basic Splitting Step を再帰的に適用する．

Step 2. $I(w) = \{1 \leq i \leq m : w | v_i\}$ を求め，次の GCD を計算．$\beta \in \mathbb{F}_p$ に対して $\gcd(u_i + \beta \frac{w' v_i}{w}, v_i)$ が自明でない因子なら w を分割してステップ 1 に戻る．自明な因子であれば w は既約となる．

◁

2.2 Niederreiter と Berlekamp の比較

Niederreiter アルゴリズムと Berlkamp アルゴリズムはよく似た行列を用いている．しかし，この行列の構築にかかる計算量は異なっている．Niederreiter 行列の構築には $O(d^\omega + (d^2 + d \log r)(\log d) \log \log d)$ ($d = \deg(f)$ かつ $\omega < 2.38$ は高速行列乗算の指数) の計算量である．一方，Berlekamp 行列の構築には高速化された乗算アルゴリズムを用いると $O((d^2 + d \log q)(\log d) \log \log d)$ の計算量である [16]．よって標数が小さい場合は Niederreiter 行列の構築のほうが早い．

3 包括的因数分解への応用

ここでは Niederreiter アルゴリズムの応用として，包括的因数分解への応用を考える．まず，包括的因数分解とは $f(x) \in F[a, b, \cdots][x]$ に対して，パラメータに値を入力するなどして，環準同型で既約分解が保持される分解を求めるものである．

$$\rho : F[a, b, \cdots] \to F^*$$

その前段階として，既約性が保たれる，パラメータの条件を求める．

3.1 既約判定の導出例

Example 1 次の $f(x) \in \mathbb{F}_2[a][x]$ について.

$$f(x) = x^2 + a$$

このとき，任意の a に対して可約になり

$$f(x) = (x+a)^2$$

◁

Example 2 次の $g(x) \in \mathbb{F}_2[a][x]$ について.

$$g(x) = x^3 + x + a$$

- 可約条件：$a = 0$, $g(x) = x(x+1)^2$
- 既約条件：$a = 1$

◁

3.2 Niederreiter 行列による既約判定法

無平方な場合，行列の階数判定により判定ができる．またこの行列は多項式の係数のべき乗のみを使うので，行列の要素が分数にならず，パラメータが入っていても行列の構成が可能である．階数が落ちる条件は小行列式で判定することができる．無平方かどうかの判定は $f(x) = (x^{d_1} + s)^{d_2}$ であるかの確認だけで十分である.

3.3 実際の条件の導出例

Example 3 次の $f(x) \in \mathbb{F}_3[a][x]$ について.

$$f(x) = x^2 + a$$

- $(x+s)^2$ より $s = 0$ のとき無平方でない.
- 既約性の確認
  (Niederreiter 行列の小行列式の集合)
  $= \{2 + 2a\}$
- 結果
  $a = 0, 2$ ならば可約
  $a = 1$ ならば既約

◁

Example 4 次の $g(x) \in \mathbb{F}_7[a, b][x]$ について.

$$g(x) = x^5 + ax^3 + bx + 1$$

- 無平方性の確認
  確認すべきは $(x+s)^5$ のみだが解は存在しない.
- 既約性の確認
  Niederreiter 行列の小行列式の集合による既約条件の導出
  $\{a, b\} = \{0, 3\}, \{0, 4\}, \{1, 1\}, \{1, 2\}, \{2, 2\}, \{3, 0\}$
  $, \{4, 5\}, \{5, 4\}, \{6, 2\}, \{6, 4\}$

◁

3.4 パラメータ条件導出のアルゴリズム

Algorithm 3 パラメータ条件導出法

[入力] $f(x) \in \mathbb{F}_p[a, b][x]$

[出力] 可約となる $a, b, \cdots \in \mathbb{F}_p$ の条件

Step 1. Niederreteir 行列の構築

Step 2. $d-1$ 次小行列式の Gröbner 基底 → gb1

Step 3. $f(x) = (x^{d_1} + s)^{d_2}$ であるかの確認 (ただし, $d_1 d_2 = d$)

3-1. 多項式の剰余を計算

3-2. 各係数を取り出し Gröbner 基底 →gb2

Step 4. gb1 と gb2 の和集合を返す

◁

4 まとめ

Niederreiter 因数分解法については過去の研究論文が多数あるので，サーベイをより進めて行こうとしている最中である．多くの論文があるため，重複した内容や変えられているところが複雑になってわかりにくくなっているので，今後これを取りまとめいていきたいと思っている．

また，Niederreiter 因数分解法の応用についてであるが，Niederreiter 行列を用いた既約判定法は部分部分，実装も進んでいる．しかし，この方法は発見的方法よりも時間がかかる．今後は，このアルゴリズムの厳密化とその証明を進めていこうと思っている．

なお，多数の研究者による多くの論文があるが，少なくとも論文 [8]，[13]，[19] に目を通すことを推奨する．

[1] F. K. Abu Salem. A new sparse Gaussian elimination algorithm and the Niederreiter linear system for trinomials over F_2. *Computing*, 77(2):179–203, 2006.

[2] P. Fleischmann. Connections between the algorithms of Berlekamp and Niederreiter for factoring polynomials over $\mathbf{F}_q$. *Linear Algebra Appl.*, 192:101–108, 1993. Computational linear algebra in algebraic and related problems (Essen, 1992).

[3] P. Fleischmann, M. C. Holder, and P. Roelse. The black-box Niederreiter algorithm and its implementation over the binary field. *Math. Comp.*, 72(244):1887–1899 (electronic), 2003.

[4] P. Fleischmann, G. O. Michler, P. Roelse, J. Rosenboom, R. Staszewski, C. Wagner, and M. Weller. *Linear algebra over small finite fields on parallel machines*, volume 23 of *Vorlesungen aus dem Fachbereich Mathematik der Universität GH Essen [Lecture Notes in Mathematics at the University of Essen]*. Universität Essen Fachbereich Mathematik, Essen, 1995.

[5] P. Fleischmann and P. Roelse. Comparative implementations of Berlekamp's and Niederreiter's polynomial factorization algorithms. In *Finite fields and applications (Glasgow, 1995)*, volume 233 of *London Math. Soc. Lecture Note Ser.*, pages 73–83. Cambridge Univ. Press, Cambridge, 1996.

[6] S. Gao. Factoring multivariate polynomials via partial differential equations. *Math. Comp.*, 72(242):801–822 (electronic), 2003.

[7] R. Göttfert. An acceleration of the Niederreiter factorization algorithm in characteristic 2. *Math. Comp.*, 62(206):831–839, 1994.

[8] S. Jeong and Y.-H. Park. A note on the factorization method of Niederreiter. *Finite Fields Appl.*, 11(2):269–277, 2005.

[9] T. C. Y. Lee and S. A. Vanstone. Subspaces and polynomial factorizations over finite fields. *Appl. Algebra Engrg. Comm. Comput.*, 6(3):147–157, 1995.

[10] R. Lidl and H. Niederreiter. *Finite fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1983. With a foreword by P. M. Cohn.

[11] H. Niederreiter. Finite fields and their applications. In *Contributions to general algebra, 7 (Vienna, 1990)*, pages 251–264. Hölder-Pichler-Tempsky, Vienna, 1991.

[12] H. Niederreiter. Factorization of polynomials and some linear-algebra problems over finite fields. *Linear Algebra Appl.*, 192:301–328, 1993. Computational linear algebra in algebraic and related problems (Essen, 1992).

[13] H. Niederreiter. A new efficient factorization algorithm for polynomials over small finite fields. *Appl. Algebra Engrg. Comm. Comput.*, 4(2):81–87, 1993.

[14] H. Niederreiter. Recent advances in the theory of finite fields. In *Finite fields, coding theory, and advances in communications and computing (Las Vegas, NV, 1991)*, volume 141 of *Lecture Notes in Pure and Appl. Math.*, pages 153–163. Dekker, New York, 1993.

[15] H. Niederreiter. Factoring polynomials over finite fields using differential equations and normal bases. *Math. Comp.*, 62(206):819–830, 1994.

[16] H. Niederreiter. New deterministic factorization algorithms for polynomials over finite fields. In *Finite fields: theory, applications, and algorithms (Las Vegas, NV, 1993)*, volume 168 of *Contemp. Math.*, pages 251–268. Amer. Math. Soc., Providence, RI, 1994.

[17] H. Niederreiter. Nets, (t, s)-sequences, and algebraic curves over finite fields with many rational points. In *Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998)*, number Extra Vol. III, pages 377–386 (electronic), 1998.

[18] H. Niederreiter and R. Göttfert. Factorization of polynomials over finite fields and characteristic sequences. *J. Symbolic Comput.*, 16(5):401–412, 1993.

[19] H. Niederreiter and R. Göttfert. On a new factorization algorithm for polynomials over finite fields. *Math. Comp.*, 64(209):347–353, 1995.

[20] P. Roelse. Factoring high-degree polynomials over $\mathbf{F}_2$ with Niederreiter's algorithm on the IBM SP2. *Math. Comp.*, 68(226):869–880, 1999.

[21] P. L. A. Roelse. *Linear methods for polynomial factorization over finite fields*, volume 25 of *Vorlesungen aus dem Fachbereich Mathematik der Universität GH Essen [Lecture Notes in Mathematics at the University of Essen]*. Universität Essen Fachbereich Mathematik, Essen, 1997. Theory and implementations, Dissertation, Universität-Gesamthochschule-Essen, Essen, 1997.

[22] C. Xing and H. Niederreiter. Applications of algebraic curves to constructions of codes and almost perfect sequences. In *Finite fields and applications (Augsburg, 1999)*, pages 475–489. Springer, Berlin, 2001.