応用代数学入門

市吉　修
2006/12/11

目次

1. 数の体系

自然数(whole number)
これは 1,2,3,,,とものを数えるための数である。その起源は先史時代のはるか彼方にあるが指の数を反映して 10 進法が多い。時計や角度では 12 進法や 60 進法が今日も用いられている。また日本では(ひ、ふ)、(み、む), (よ、や)のように倍数関係にもとづくと思われる数え方もある。自然数においては加算+と乗算・が定義され、交換側 a・b = b・a や分配則 a・(b + c) = a・b + a・c が成り立つ。

整数(integer)
自然数は加算は定義されるが減算は負数が生じ得る為数の体系としては不完全である。自然数に負の数を加えて加減算法に対して群(group)を成す数系を構成するには零が必要である。零を導入すれば加算+に対してある整数 a の逆元-a は–a = 0 - a として定義できる。分配側 a・(b + c) = a・b + a・c において b=-c を代入すると a・0 = 0 なる乗算が成立し, a=1,c=0 を代入すると b+0 = b なる加算が成立する。整数は加算については群を成すが乗算については群を成さない。整数は数系としては環(ring)を成す。

有理数(rational number)
整数に分数を導入することにより拡張した有理数は乗算についても群をなす。すなわち数 a の乗算に関する逆元 a'は a・a' = 1 を満たす数であり a' = 1/a と表記する。有理数は加減算と乗除算が完全にできる数系である。加減乗除が可能な数系は体(fields)と呼ばれる。即ち有理数の全体は有理数体を成す。任意の有理数 a,b の間には例えばその加重平均(a+b)/2 があるからいかなる数も有理数で幾らでも正確に近似できる。即ち有理数の全体は稠密(compact)である。

実数(real number)
有理数に等しくない、即ち分数で表現できない(例$\sqrt{2}$)数が有理数の全体よりも遥かに多く存在することが知られている。これを無理数(irrational number)と呼ぶ。無理数まで拡張した数系が実数体である。

複素数　(complex number)
i^2 = -1 なる虚数(imaginary number) i を含む数の数系を複素数(complex number)と呼ぶ。複素数は実部 x, 虚部 y の二つの成分を有し z = x + iy, = (x, y)と表記される。実部、虚部を整数の範囲に限定すれば複素整数となり、実数の範囲にすれば実複素数となる。

多項式(polynomial)
自然数 n 次の多項式は変数(variable) x に対して　P(x) = a[n].x^n + a[n-1].x^(n-1) + ,,,,+a[1].x + a[0]
で定義される。係数{a[i]}は応用に応じて種々の数系の値をとる。係数が体 F の数であるときその多項式は

体 F 上の多項式であると言う。多項式の集合は整数と同様に環を成す。

代数学の基本定理
実係数の n 次多項式 P(x)は複素数の範囲で一次または二次の多項式の積に因数分解できる。
即ち n 次方程式は複素数の範囲で n 個の根を有し、n 個の一次式の積に因数分解(factorization)できる。

代数的数と超越数
実係数の多項式の根で表される数を代数的数(algebraic number)と呼び、多項式の根で表されない数を超越数(transcendental number)と呼ぶ。円周率πや自然対数の底 e は超越数であることが知られている。

2. 整数論的関数

<オイラーの関数,Totient function>
正整数数 m に対して 1,2,,,,m-1 のうち m と互いに素な数の数として定義される関数はオイラーの関数と呼ばれφ(m)で表される。
m = p(素数)のとき　　φ(p) = p-1、　　但しφ(1) =1 と定義する。
m = p^α　(αは自然数)の時　　φ(p^α) = p^α - p^(α-1)
　これは{1,2,,,p^α}なる,p^α個の数のうち p で割り切れる数が p^α/p= p^(α-1)個ある事より明らか。
M が一般の合成数の場合;　m の素因数分解を m = p[1]^α[1]・p[2]^α[2]・・・p[k]^α[k]とすると
　　φ(m) =φ(p[1]^α[1])・φ(p[2]^α[2])・・・φ(p[k]^α[k])
　　= m.・(1 – 1/p[1])・(1 – 1/p[2])・・・(1 – 1/p[;k])
実際 m = p^α・q^βの場合 p と互いに素でない数は m/p 個、q と互いに素でない数は m/q 個、それらの数のうちダブっている数は m/(.p.q)個ある。従ってφ(m) = m – m/p – m/q + m/(p.q) = m.(1-1/p).(1-1/q)となる。以下素因数の数が多くなる場合にも同様にして証明できる。

<剰余(remainder)>
整数 A を整数 m で割った余りを剰余と呼び A (mod m)と表す。剰余が 0 の時 A は m で割り切れる(A is divisible by m)。集合{0,1,2,,,,,m-1}を m の剰余系と呼ぶ。

<既約剰余系(reduced remainders set)>
正整数 m に対して 1,2,,,,m-1 のうち m と互いに素な数の集合を m の既約剰余系と呼ぶ。既約剰余系の元の数はφ(m)である。特に m = p (素数)の場合の既約剰余系は{1,2,,,,p-1}である。

<剰余類体>
素数 p の剰余類{0,1,2,,,,,p-1}は有限体 F(p)を成す。
実際 F(p)の要素 a に対して加算の逆元 a'は　a' = p – a で与えられる。
乗算に関する零でない要素 a の逆元 a"の存在は次のようにして確認できる。変数 x と a の積 a・x は x が F(p)内の元を回ると順番は異なるが同じく F(p)の元を重複する事無く回る。実際 F(p)の二つの相異なる要素 x, y に対して a・x = a・y (mod p)となったとすると a・(x – y) = 0 (mod p)となり、a =/= 0 であるから x = y となり仮定に反する。従って a に対して a・x = 1 (mod p)なる要素 x が唯一存在する。その x が乗算に関する a の逆元である。

<Fermat の定理>
素数 p に対して任意の数 a をとると
　　a^(p-1) = 1 (mod p)
任意の数 m に対しては m と互いに素な任意の数 a に対して
　　a^φ(m) = 1 (mod m)
実際 m の既約剰余系の要素 x[i] (i=1,2,,,　φ(m))に対して y[i] = a・x[i] (mod m)なる既約剰余が一意に定まる。両辺をすべての i について掛け合わせると∏y[i] = a^φ(m)・∏x[i]となる。 ∏y[i] = ∏x[i]であるから両辺を約分すれば a^φ(m) = 1 (mod m)が得られる。

<原始根(primitive o generator)>
有限体 F(p)の要素αから 1=α^0，α，α^2，α^3,,,,とべき乗を作ると F(p)は有限であるから特定の数 r に対して，α^r = 1 となる。このとき要素αは指数 r に属すると称する。
Fermat の定理によりα^(p-1)=1 であるから r は p-1 の約数となる。
特に指数 r = p-1 となる要素を原子根(primitive)と呼ぶ。原始根 g は g^i (i=0,1,2,,,,p-1)により F(p)の 0 以外のすべての元を重複無く生成するので生成元(generator)とも呼ばれる。

<離散対数(discrete logarithm)>
原始根 g によって F(p)の任意の元 x に対して y = g^x (mod p)なる y が一意に決まる。そこでその逆関数を離散対数と呼び　x = ind[g](y) と表す。今指数 x'に対応する元を y'とすると y・y' = g^(x+x')となるので実数の指数関数とその逆関数の対数に相当する関係にあることが分かる。この時 g は対数の底に対応する役割を果たしている。

<指数表(index table)>
素数 p に対して F(p)の指数と対数表を予め作っておけば計算に便利である。
例えば p=11 に対する指数表は原始根 g=2 に対して

Ind(x)	0	1	2	3	4	5	6	7	8	9	10
X	1	2	4	8	5	10	9	7	3	6	1

応用例
　5.x = 3 (mod 11)を求めよ。
両辺の離散対数をとると
　ind(5) + ind(x) = ind(3) ➔　ind(x) = ind(3) – ind(5) = 8 – 4 = 4 ➔ x=5

3. 有限体概論

上述のごとく整数環から素数 p による剰余類体 F(p)を生成することができるが同様の手法を多項式環に拡張することができる。

<体 F 上の多項式>
n 次多項式
　P(x) = a[n]..x^n + a[n-1]..x^(n-1) + ,,,, + a[1]. x + a[0]
の係数 a[i] (i = 0,1,2,,,,,n) が体 F の要素であるとき P(x)は体 F 上で定義された多項式である。

<既約多項式、Reduced polynomial>
方程式　P(x) = 0 が体 F 上に根を持たないとき P(x)は既約(reduced)であるという。

<拡大体, Extended fields)>
体 F 上で根を持たない方程式 P(x) = 0 は拡大体において根を有する。例えば実数体上では根が無い方程式も複素数体に拡大すると根がある。代数学の基本定理によれば n 次方程式は n 個の根を持つ。

<剰余類体>
多項式 p(x)が体 F 上で規約であるとする。
任意の多項式 P(x)を p(x)で割った余りを r(x) =P(x) (mod p(x))と表記すると r(x)の全体は Galois 体 GF(p(x))を成す。
例　任意の実数 a,b に対して複素数 c = a + ci　(i^2=-1)が定義されるが、これは i の多項式として与えられた数を i^2 + 1 を法とする剰余類に分類するのと等価である。
例えば　(a+bi).(c+di) = a.c +b.d.i^2 +i.(b.c + a.d)　(mod i^2 +1)= ac-bd +i(bc+ad)
.

<巡回多項式 Cyclic polynomial>
多くの応用は x^N = 1 なる体上で設計される。
n 次の多項式　　p(x)=　　a[n].x^n　+　a[n-1].x^(n-1) + ,,,+　a[1].x　+　a[0]
に対して　　x.p(x)= a[n].x^(n+1)+a[n-1].x^n +　　,,,,　　+ a[0].x
となるように x を掛けると多項式の各項が移動するが x^N=1 であるから x^N.p(x) = .p(x)となり周期 N で巡回する。

<　二進数体　binary field>
ディジタル通信においては二進数体が広く用いられる。要素 0,1 間に加算 0+0=0,0+1=1,1+0=1　1+1=0 及び乗算 0・0 = 0 ,0・1=0, 1・0 = 0, 1・1 = 1　これを位数 2 のガロア体と呼び GF(2)で表す。

< Galois 拡大体　Extension field >
二進数体 GF(2)上の n 次の既約多項式 p(x)を法とする剰余多項式は要素の数が 2^n の拡大体 GF(2^n)を成す。p(x)= x^n + p[n-1].x^(n-1) + ,,, + p[1].x +1 の形を取る。定数項は必ず 1 である。さもなければ x=0 が根となり p(x)が既約であるとの仮定に反する。係数 p[i]　(i=1,2,,,,n-1)は 0 か 1 の値を取るが、p(x)の項の数は奇数である。なぜならもしも項の数が偶数であれば x=1 が p(x)の根となり p(x)が既約であるという仮定に反する。p(x)の剰余類は r(x) = a[n-1].x^(n-1) + a[n-2].x^(n-1) + ,,,, + a[1].x + a[o]　なる高々n-1 次の多項式の集合である。係数 a[i] = 0,1 なる二進数であるから要素の数は 2^n 個ある。p(x)を法とする剰余多項式の集合が体を成す事は前述の整数環において素数 p の剰余類が体を成すのと同様の論理により確認できる。

<原始多項式と原始根　Primitive　>
GF(2^n)を生成する既約多項式 p(x) = 0 の根を g とする。1 = g^0, g, g^2 , g^3,,,,と g のべき乗を作ると有限体であるから或る自然数 r に対して g^r = 1 となる。定義により g^(2^n-1) =1 となるので r は 2^n-1 の約数である。特に r = 2^n -1 のとき、g は GF(2^n)のすべての非零要素を重複無く生成する。そのような g を原始根、g を根とする多項式を原始多項式と呼ぶ。体の要素を生成することから生成多項式(generator polynomial)とも呼ばれる。
例　p(x) = x^3 + x + 1　　p(x)で生成される GF は

g^0 = 1	(0,0,1)
g　= x	(0,1,0)
g^2 = x^2	(1,0,0)
g^3 = x^3 = x +1	(0,1,1)
g^4 = x^2 + x	(1,1,0)
g^5 = x^3 + x^2 = x^2 + x +1	(1,1,1)
g^6 = x^3 + x^2 + x = x^2 +1	(1,0,1)
g^7 = x^3 + x =1	
上の要素に　0	(0,0,0)

を付け加えた集合{0, 1, g, g^2,,,,, g^6}は GF(8)を成す。

< 離散フーリエ変換　Discreet Fourier Transform DFT >
x^N -1 = (x-1).(x^(N-1) + x^(N-2) + ,,,,+ x + 1) であるから x^N -1 = 0 の一つの根は x = 1 である。それ以外の根 g は　g^(N-1) + g^(N-2) + ,,,, + g + 1 = 0　を満足する。
多項式　c(x) = c[N-1].x^(N-1) + c[N-2].x^(N-2) +,,,,+ c[1].x + c[0]で表されるデータの集合{c[i]; i=0,1,,N-1} に対してその Fourier 変換{C[k]; k= 0,1,2,,,,N-1}を次のように定義できる。

C[k] = [i=0, N-1]∑ c[i].g^(k.i)
C[i] = 1/N [k=0, N-1]∑ C[k].g^(-k.i)　　(逆変換)

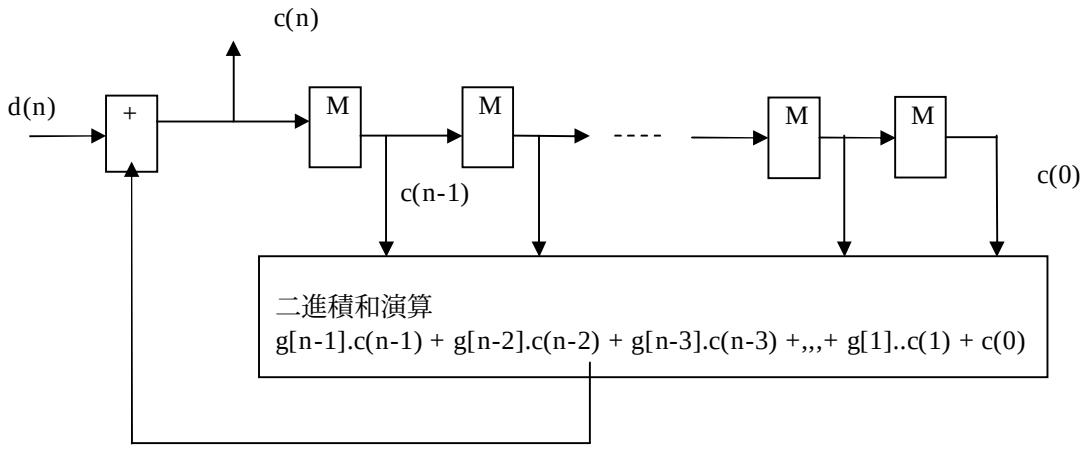
逆変換の成立することは　[i=0,N-1]∑ g^(k.i) = (1- g^(N.k)) / (1-g^k) = 0 (k=/=0) ,または N(k=0)　より明らかである。
これは丁度複素実数に対する離散 Fourier 変換(DFT)と同じ内容の定理である。複素数に対する場合には x^N-1 = 0 の根は g = e^(j2π/N) (j^2 = -1)で与えられる。一般の場合には演算規則はその変数が定義される体の演算規則に従う。

4. PN 信号への応用

Pseudo Noise 信号は擬似雑音信号の意味である。従来試験用データ系列発生器、ビット誤り率測定器、暗号化回路、測位など広汎に用いられて来た。最近では CDMA 方式の移動通信に広く用いられている。

<PN 符号発生器>

n 段シフトレジスタを用いて　c(n) = .c(n-1) + g[n-2].c(n-2) + g[n-3].c(n-3) +,,,+ g[1]..c(n) + c(0)
なる式で符合を発生する。ここで演算は二進数体である。係数 g(i)は 1,0 のいずれかの値をとる。これは次図に示す回路で実現される。ここで d(n)は外部より初期条件を設定するための入力である。



上の図より

c(n) = d(n) + g[n-1].c(n-1) + g[n-2].c(n-2) + g[n-3].c(n-3) +,,,+ g[1]..c(n-(n-1)) + c(n-n)

あるいは二進演算では-a = +a なることに注意して

c(n)+ g[n-1].c(n-1) + g[n-2].c(n-2) + g[n-3].c(n-3) +,,,+ g[1]..c(n-(n-1)) + c(n-n) = d(n)

両辺の z 変換をとると　C(x).G(x) = D(x)　となる。ここで

G(x) = x^n + g[n-1]. x^(n-1) + g[n-2].x^(n-2) + g[n-3].x^(n-3) +,,,+ g[1]..x +1

また　C(x) = [i]∑ c(i).x^i

である。通常の z 変換に対して x=1/z となることに注意。それは単に述語の定義のちがいである。

<初期条件と無限系列の発生>

D(x) = 1 の場合、すなわち最初に 1 という初期値を入力すると　C(x) = 1/G(x)　となり、入力に何も入れなくても自励発振して無限系列を発生する。

<特性方程式と最長符号系列>

通常無限系列が発生し始めると入力は 0 にするので C(x)は G(x) = 0 なる特性方程式(Characteristic equation)の条件を満たしつつ発生される。即ち系列 c(x)は G(x)を法とする巡回多項式となる。生成多項式が n 次でシフトレジスタは二進 n 段であるから法 G(x)の拡大体の要素の数は 2^n – 1 である。従って巡回周期 r(x^r=1)は 2^n-1 の約数である。特に r= 2^n-1 なるものを最長符号系列(Maximum Length code)と呼ぶ。

<原始根と拡大体>

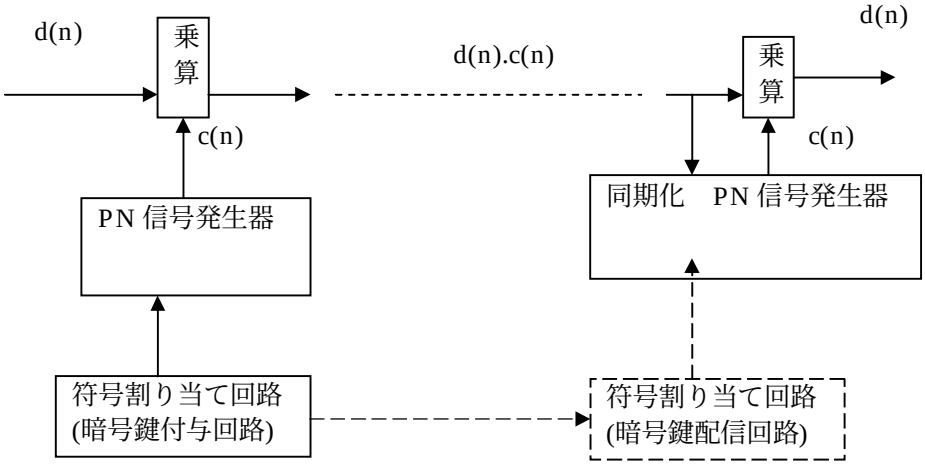
n 段シフトレジスタによる最長符号系列の周期は L = 2^n – 1 である。シフトレジスタの内容をベクトルと見ると< c(m)] = (c(m-1), c(m-2),,,　c(m-n+1), c(m-n)),とすると< c(m+1)] = < c(m)].G となる。ここで G は遷移行列(transition matrix)であり下表の成分を有す。今初期ベクトルを< c(0)]とすると m 回の遷移ごのベクトルは< c(m)] = <c(0)].G^m となる。これは L=2^n-1 回後に初めて元にもどるから G^(2^n-1) = I(単位行列)となる。すなわち G のべき乗により取りうる 2^n-1 個のベクトル(シフトレジスタの内部状態)

を重複無く生成する。即ち G は原始根である。

g[n-1]	1	0	0		0
g[n-2]	0	1	0		0
g[n-3]	0	0	1		0
g[n-4]	0	0	0		0
---	0	0	0		0
---	0	0	0		0
g[1]	0	0	0		1
1	0	0	0		0

<相関特性>
最長符号系列は零ベクトル以外のすべての状態をとるから PN の一周期内の 1 の数は 2^n/2 個、0 の数は 2^n/2-1 個ある。また G は原始根であるから任意の数 k,l に対して G^k + G^l = G^m なる m がある。即ち位相の異なる二つの最長符号系列を二進加算すると同じ生成行列から生成される PN 信号になる。
通常の応用では 1 を-1,0 を+1 に対応させて用いる。このとき二進加算は乗算に対応する。
1x1 = 1, -1x(-1) =1 1x(-1) = -1 ←→ 0 + 0 = 0, 1+1 = 0, 1+0 = 1
今最長符号系列 C(x) = [i]∑c(i).x^I　(c(i) = 1,-1) の自己相関関数(auto-correlation function)は R[C](m)は R[C](m) =[i = 0, 2^n-1] ∑c(i).c(i+m)として定義される。上述の事から R[C](m) = 2^n-1 (m=0), -1(m=/=0) となる。すなわち周期 2^n-1 なる最長符合系列は自己相関性が非常に高い。同じ PN 信号でも 1 シンボル(chip)以上ずれると無相関に近くなる。
また同じ周期でも生成多項式の異なる最長符号系列間の相互相関(cross-correlation)は零になる。これらは雑音に類似した性質なので最長符号系列は擬似雑音、PN 信号と呼ばれる。

< CDMA>
上述の擬似雑音特性を利用して通信の多重化を行うのが符号分割多重(Code division multiple access, CDMA)方式である。同様に暗号化にも応用できる。両者とも回路構成は類似している。



<ML 符号の数>
n 段の最長符号系列の数はφ(2^n-1)/n 個ある。n=10 段で 60 個、15 段で 1800 個ある。各段の生成多項式は文献で表にまとめられている(S.W.Golomb)。

<Gold 符号>
異なる周期 L,L'の PN 符号を乗算する、即ち二段に PN 変調を行うと周期 L.L'の PN 符号が得られる。こ

れを Gold 符号と呼ぶ。

<暗号化>
PN 符号は CDMA と同様の回路構成で暗号にも用いることができる。Gold 符号化により PN 符号の数は無尽蔵に増やせる上に周期が長大化できるので暗号には特に有効である。この場合符号の組み合わせと初期条件を復号化の鍵として受信者に配信する必要が生じる。

5. 暗号への応用

上述の暗号法は送信側と受信側が共通の鍵を共有しかつその秘密が保たれなくてはならない。それを秘密鍵共有方式(shared secret key system)と称するが遠隔通信においては実行困難である。

< 公開鍵暗号法の必要性>
伝統的な暗号方法は通信文列に対して換字、即ち文字の置き換えと転置、即ち順番の入れ替えを行なうものである。現代暗号理論は文字を始め、映像、音声すべてを数値化して扱う所に特長がある。また伝統的な暗号は殆ど一対一通信であった。この場合は当然送信と受信とで共通の秘密鍵を用いる。他方インターネットは開放的な通信網である。HP を開設すれば不特定多数の人に発信できるし、逆に情報を収集する事ができる。この際、投書の中味を部外者に読まれないようにするには暗号化のための鍵を公開する必要がある。公開鍵暗号法の代表例として RSA 暗号法が広く用いられている。

< RSA 暗号法 >
(1) 数値ｎとｅを公開する。
(2) 通信(平文 Plain-Text)は数値系列に変換される。その数値の一つをＰとする。
(3) 暗号化はＣ　＝　Ｐ＾ｅ　(mod　n)なる演算を行なう。
(4) 受信側においては秘密鍵 d を用いて
　Ｃ＾ｄ　（mod　ｎ）なる演算を行なうと
　Ｃ＾ｄ　＝Ｐ＾(ｄｅ)（mod　ｎ）＝　Ｐ
　となり復号される。

上の鍵ｅ，ｄと数ｎとは次の様にして決められる。
(a)　大きな素数ｐ，ｑを選び n　＝　p･q
(b)　ｅ・ｄ　＝１　(mod　(p-1)・(q-1))
　　なるｅ，ｄを選ぶ。
(C)　ｎとｅを公開し、ｄ，ｐ，ｑを秘す。

RSA の原理は次の Fermat の（小）定理である。素数ｐと任意の数Ａについて
　Ａ＾(p-1)　＝　１　(mod　p)

暗号語Ｃに対して
Ｃ＾ｄ　＝　Ｐ＾ed＝　Ｐ＾　{1+t(p-1)・(q-1)}
＝　Ｐ・（Ｐ＾ｔ）＾(p-1)(q-1)
Fermat の定理により第二因子は１となり
＝ P (mod n) (n=pq)

RSA 公開鍵暗号法は n を公開してもそれを因数分解して p,q を求める事が計算上困難であることにもとづく。そのために p,q,n は非常に大きな数から選ぶ必要がある。

< Diffie-Hellman 鍵交換法 >
インターネット上で Alice, Bob なる二者間で暗号化通信を行うには共通の使い捨て秘密鍵を公開通信路上で設定できれば好都合である。Alice は公開されている大きな素数表の中から素数 n,g を選び自分で勝手

に選択した数 x を用いてα= g^x (mod n)を計算する。Alice は Bob に(n,g,α)を送る。Bob は送られてきた情報をもとに勝手な数 y を選択しβ= g^y (mod n)を計算しβを Alice に返信する。Alice は受け取ったβを用いてγ= β^x を行い、Bob はγ'=α^y を行う。するとγ=γ' = g^(x.y) (mod n)となり秘密鍵を共用できる。第三者は n,g,α、βを盗み見できるがそれから秘密の数 x, y を計算することが困難である。即ち第二章で解説した離散対数の計算は数値が大きくなると極めて困難である。この事情は原始根の指数表で数字の並びが一様でないことから理解されると思う。

6　誤り訂正技術への応用

誤り訂正符号は大別すると畳み込み符号(convolutional codes)とブロック符号(block codes)の二種類がある。ここでは代表的なブロック符号である巡回符号(Cyclic codes)について解説する。

< 巡回符号>

Block 長 n (語、words 又はシンボル、symbols)の巡回符号は下の多項式で表現される。

c(x) = c[n-1].x^(n-1) + c[n-2].x^(n-2) + ,,,, + c[1].x + c[0]

実用上広く用いられている巡回符号(Cyclic codes)とは　x^n =1　なる条件を満足する符号である。
上の巡回符号においては

x.c(x) = c[n-2].x^(n-1) + ,,,,, + c[1].x^2 + c[0].x + c[n-1]

となり巡回する。x^i を掛けると i 語だけ巡回シフトする。

<符号間の距離と最小重み>

二つの符号　c(x) = [i= 0, n-1]∑ c(i).x^i と　d(x) =[i= 0, n-1]∑ d[i].x^i　の差 c(x) –d(x)は項別の差を取る多項式である。c(x) – d(x) = 0 とはすべての項が 0 になる事である。差異が 0 とならない項の数が D 個ある時符号 c(x)と d(x)の距離は D(語)であると言う。また c(x) – d(x)の重み(weight)が D であるとも言う。二つの巡回符号の差はまた一つの巡回符号であるから符号間の最小距離とはその巡回符号のうち重みが最小となる符号の重みに等しい。ただし重みが 0 となる符号(入力が all 0 の場合に相当)は除く。

< 巡回符号の符号化法>

符号長が n でそのうち情報語が k 個ある符号を(n,k)符号と表記する。k 個の情報語に r = n-k 個の検査語(parity check words)を加えて符号化を行う。:検査語の発生は r 次の生成多項式 g(x)によって行う。よく用いられるのは次の組織的符号(systematic codes)である。

情報系列　d(x) = d[k-1].x^(k-1) + d[k-2].x^(k-2) + ,,,　+ d[1].x + d[0]

符号　　c(x) = x^r.d(x) – Rem[x^r.d(x)/g(x)]

但し Rem[P(x)/Q(x)]は P(x)を Q(x)で割った余りを意味する。
即ち情報系列を r 個高次にシフトし次にそれを g(x)で割った余りを引くので c(x)を g(x)で割ると 0 になる。組織符号の利点は　c = (d[k-1], d[k-2], ,,,d[1], d[0], p[r-1],p[r-2], ,,,,p[1], p[0])のように符号の前部に情報系列がそのまま入っているので分かりやすいことである。

<巡回符号の復号法>

伝送路で誤り e が付加されて受信信号は c(x) + e(x)となる。そこで受信信号を生成多項式 g(x)で和って余りを取る事により Syndrome 多項式 s(x)が得られる。符号 c(x)を g(x)で割ると余りは 0 であるから s(x)は伝送路で生じた誤りの情報を含んでいる。もし s(x) = 0 であれば誤りは生じなかったと解釈される。

<生成多項式のつくり方>

上の生成多項式 g(x)は巡回符号は x^n -1 = (x-1).(x-ω).(x-ω^2).,,,,.(x-ω^(n-1))　なる因数多項式から生成される。ここでωは原始多項式 p(x) = 0 の根、原始根(primitive)である。原始多項式の次数を r とすると原始根ωのべき乗によって 1,ω，ω^2 ,,,,　ω^(2^r-1)が重複無く生成される。0 と合わせてガロア各大体 GF(2^r)が生成される。GF(2^r)上で巡回符号を生成するには n = 2^r-1 なる原始多項式を選定する。

<BCH 限界>

上の因数多項式のうち引き続く 2t 個の因数多項式を選んで生成多項式(generator　polynomial)g(x)を作

るとそれから生成される巡回符号は最小重みを 2t+1 にすることができる。即ち t 重誤り訂正符号を構成できる。これを BCH 限界(BCH bound)定理と呼ぶ。

<Reed-Solomon 符号>
生成多項式 g(x) = (x-1).(x-ω).(x-ω^2).,,,,.(x-ω^(2t-1))により生成される巡回符号を Reed-Solomon 符号と呼び(n, n-2t, t)符号により t 重誤り訂正が可能である。Reed-Solomon 符号は GF(2^r)上の巡回符号であり語は r ビットの語である。従って最大 r.t ビットの二進バースト誤りまで訂正可能である。
Reed-Solomon 符号は符号化効率が高く誤り訂正能力も高いので広く用いられている。例えば直接衛星放送では(208,188)が用いられている。2t=208-188=20 であるから n=208 の中の t=10 語までのランダム誤りを訂正可能である。ここで語は Byte,即ち 8 ビットである。自然な数値は n = 2^8-1 =255 (bytes)であるがシステムの構成上の都合から情報部の 47 bytes は予め 0 としている。これを短縮化(shortening)と呼び符号化効率は落ちるが種々のシステム要求に合わせた設計が可能となる。

< n=15 の Reed-Solomon 符号　>
n = 15 の巡回符号は　　x^15 = 1 を満足する原始根から生成できる。
p[1](x) = x^4 + x + 1 = 0 の根をωとする。

ω^0 =1	(0,0,0,1)
ω^1 =ω	(0,0,1,0)
ω^2	(0,1,0,0)
ω^3	(1,0,0,0)
ω^4 = ω+1	(0,0,1,1)
ω^5 = ω^2 +ω	(0,1,1,0)
ω^6 = ω^3 +ω^2	(1,1,0,0)
ω^7 = ω^4 +ω^3 = ω^3 +ω +1	(1,0,1,1)
ω^8 = ω^4 +ω^2 + ω = ω^2 +1	(0,1,0,1)
ω^9 = ω^3 +ω	(1,0,1,0)
ω^10 = ω^4 +ω^2 = ω^2 + ω+1	(0,1,1,1)
ω^11 = ω^3 +ω^2 + ω	(1,1,1,0)
ω^12 = ω^4 +ω^3 + ω^2 =ω^3 +ω^2 +ω+1	(1,1,1,1)
ω^13 = ω^4 +ω^3 + ω^2+ω =ω^3 +ω^2 +1	(1,1,0,1)
ω^14 = ω^4 +ω^3 + ω =ω^3 +1	(1,0,0,1)
ω^15 = ω^4 +ω = 1	

t = 2 誤り訂正可能な Reed-Solomon 符号としては根として 1,ω,ω^2,,ω^3 なる項をとって
g(x) = (x-1).(x-ω).(x-ω^2).(x-ω^3) を作ればよい。これにより(15,11)二重誤り訂正符号が形成できる。

< n=15 の BCH 符号>
Reed-Solomon 符号は GF(2^r)上の符号であり、演算が多少複雑である。BCH 符号は GF(2)上の符号である。即ち二進論理回路で実現できる。GF(2)において(x+1)^2 = x^2+1 である。一般の多項式について P(x)^2 =P(x^2)であるから上の原始根ωを根とする最小多項式 m[1](x)はω,ω^2,ω^4,ω^8 をも根とする。

根	最小多項式
ω^0 = 1	m[0]{x} = x+1
ω, ω^2, ω^4, ω^8	m[1](x) = (x+ω).(x+ω^2).(x+ω^4).(x+ω^8) = x^4 + x +1
ω^3, ω^6, ω^12, ω^9	m[3](x) = (x+ω^3).(x+ω^6).).(x+ω^12).(x+ω^9) = x^4 + x^3 + x^2 + x +1
ω^5, ω^10,	m[5](x) = (x+ω^5).(x+ω^10) = x^2 + x + 1
ω^7, ω^14, ω^13, ω^11	m[7](x) = (x +ω^7).(x + ω^14).(x +ω^13). (x + ω^11) = x^4 +x^3 + 1

一重誤り訂正符号は g(x) = m[0](x).m[1](x)とすればよい。この時 g(x)は 5 次であるから(15, 10)ブロック

符号が形成される。二重誤り訂正符のためには g(x) = m[0].m[1].m[3](x)もしくは m[3].m[7]号をを用いて (15, 6), (15, 7)ブロック符号を形成することができる。
各種の符号(n,k,t)に対する BCH 符号の表が文献化されている(Lucky, et.al)。

上の例が示すように BCH 符号に比べて Reed-Solomon 符号は遥かに効率的で誤り訂正能力も高い。しかし演算は GF(2^r)での演算であり乗算が多少面倒である。それに対して BCH 符号は GF(2)上の符号であり演算は二進論理回路で構成できる。1980 年代までは BCH 符号が主力であったがディジタル信号処理技術の発展によりそれ以降は Reed-Solomon 符号が使われるようになった。特に Reed-Solomon 符号のバースト訂正能力を畳み込み符号/Viterbi 復号法と組み合わせた連接符号(concatenated codes)が直接衛星放送などに広く用いられている。

< BCH 定理の証明>
GF(n)上の x^n - 1 = 0 の原始根ωのべき乗のうち引き続く 2t 個の根を有する多項式を生成多項式とする巡回符号の最小重みは 2t+1 であり、t 重誤り訂正可能である。
[証明]
符号　　c(x) = c[n-1].x^(n-1) + c[n-2].x^(n-2) + ,,,, + c[1].x + c[0]　　に対して
フーリエ変換　C[k] = c(ω^k)　　(k = 0,1,2,,,,n-1)
そこで　　C(y) = [k = 0, n-1]∑ C[k].y^k　　なる多項式を定義すると
フーリエ逆変換　c[l] = 1/n. C(ω^(-l))　　(l = 0,1,2,,,,,n-1)
仮定により引き続く 2t 個の C[k]は 0 である。巡回符号の性質から C[k+k']なる k'だけのシフトをしたものも同じ符号に対応するから、C[k] = 0　(k = n-1, n-2, ,,,, n-2t)となるようにできる。
即ち C(y) = [k = 0, n -2t-1]∑ C[k].y^k は高々n-2t-1 次式である。代数学の基本定理により C(y)は高々n-2t-1 個の零しか持ち得ない。従って少なくとも 2t+1 個の c[l]は非 0 である。

< Reed-Solomon 符号の復号法>
受信信号を r(x)とすると　r(x) = c(x) + e(x)である。ただし+は二進加算である。
誤り多項式　e(x) = e[n-1].x^(n-1) + e[n-2].x^(n-2) + ,,,, + e[1].x + e[0]　において誤りが生じた項に対して e[l] = 1, 生じなかった項に対して e[l] = 0 である。
受信信号のフーリエ変換をとると　R[k] = C[k] + E[k] (k = 0,1,2,,,,n-1)
ここで C[k] = 0 (k = n-1,n-2,,,,n-2t)であるから　　R(k) = E[k] (k = n-1,n-2,,,,n-2t)　　この 2t 個の量は誤りに関する情報を与えるω領域での Syndrome である。
誤り位置特定多項式(locator polynomial)　λ(x)= λ[n-1].x^(n-1) + λ[n-2].x^(n-2) +,,,,+ λ[1].x + λ[0]
の各項は対応する受信信号の誤り位置で 1,誤りが無い位置では 0 となるものとして定義される。
従ってすべての l に対して　λ[l].e[l] = 0　　(l = 0, 1, 2, ,,,,n-1)
{λ[l] ; l = 0, 1, 2, ,,,, n-1}のフーリエ変換をΛ[k]とする。　Λ[k] = λ(ω^k)　(k= 0, 1, 2, ,,, n-1). そこで Λ(x) = Λ[n-1].x^(n-1)+,,,+Λ[1].x + Λ[0]　　を定義すると　　λ[l] = 1/n.Λ(1/ω^l)　となる。
ここで誤り数は t 以下であるものと仮定する。するとΛ(x)の根は t 個以下であるからΛ(x)は高々t 次の多項式である。Λ[0]= 1 に正規化すると　Λ(x) = 1 + Λ[1].x +,,,,,, + Λ[t].x^t
他方フーリエ変換の定理から

0 = [l = 0, n-1]∑ e[l].λ[[l] x^l ←

➔ Λ(x).E(x) = [k= 0, t][m = 0, n-1]∑ (∑Λ[k].E[m-k]).x^m

これより次の連立一次方程式が得られる。

[k= 0, t]∑Λ[k].E[m-k] = 0　(for all m)

ここで前述の Syndrome から得られる{E[m]; m= n-1,n-2,,,,,,n-2t }について連立方程式を書き下すと

E[n-2].Λ[1] + E[n-3].Λ[2] +,,,,,,,,,,,,+ E[n-1-t].Λ[t]　= E[n-1]
E[n-3].Λ[1] + E[n-4].Λ[2] +,,,,,,,,,,,+ E[n-2-t].Λ[t]　= E[n-2]

E[n-1-t].Λ[1] +E[n-2-t].Λ[2] +,,,,,,,+ E[n-2t].Λ[t]　= E[n-t]

これは未知数が t 個、式の数が t 個であるから解くことができる。

この方程式を解いてΛ(x)を求めるとλ[l] = 1/n.Λ(1/ω^l)により誤りの位置(と値)が求められるので誤り

訂正を行うことができる。
誤りの数 v は仮定により t は越えないが 0 から t までの任意の値を取りうる。この時は Λ(x)は v 次式となるが上の連立方程式の標準的な解法で v = < t なる限り解くことができる。

上の連立方程式を解く方法として Berlekamp-Massey 方式が用いられる。またフーリエ変換を取らずすべて時間領域で演算する復号法も用いられる。詳細は文献[1](R.E.Blahut)を参照されたし。

参考文献

[1] Richard E. Blahut , Algebraic Methods for Signal Processing and Communication Coding Springer-Verlag 1992
[2] Solomon W. Golomb, Shift Register Sequences Holden-Day, Inc. 1967
[3] Andrew S. Tannenbaum, Computer Networks third edition, Prentice-Hall International,Inc. 1996
[4] 一松 信 「暗号の数理」 ブルーバックス 1980
[5] Neal Koblitz A Course in Number Theory and Cryptography second edition Springer 1994
[6] I.M. Vinogradoff, 三瓶与右衛門、山中健訳 「整数論入門」 共立全書
[7] R.W.Lucky, J.Salz, E.J.Weldon,Jr Principles of Data Communication Mcgrow Hill 星子幸男他訳 「データ通信の原理」 ラティス刊
