Decoding Goppa Codes with MAGMA

Giorgio Faina and Massimo Giulietti *

Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Via Vanvitelli, 1
06123 Perugia, Italy

faina@dipmat.unipg.it giuliet@dipmat.unipg.it

Abstract

The aim of this note is to provide a programme for the Computer Algebra package MAGMA, which is suitable to decode one-point Goppa Codes defined from Hermitian curves.

Key words: Decoding Goppa Codes, Hermitian Curves.

1 Introduction

Ideas from Algebraic Geometry become useful in Coding Theory after Goppa's construction [4]. He had the beautiful idea of associating to an algebraic curve C defined over the finite field with q elements $\mathbf{F}_q$, some linear codes. These codes, nowadays called *algebraic-geometric* or *Goppa* codes, are defined from two divisors D and G on C, where one of them, say D, is the sum of distinct $\mathbf{F}_q$-rational points of C.

The problem of decoding algebraic-geometric codes has been deeply investigated in the past few decades. A first attempt to decode Goppa codes was made by Driencurt [2] for elliptic curves. At the end of the 1980's, Justesen, Larsen, Jensen, Havemose and Høholdt [6], [8] found for algebraic-geometric codes on plane curves a generalization of the decoding

*The research was supported by Italian MURST progetto 40% *Strutture Geometriche, Combinatoria e loro applicazioni*, by CNR Progetto Strategico *Applicazioni della matematica per la tecnologia e la società* sottoprogetto *Calcolo Simbolico* and by GNSAGA

algorithm of Arimoto [1] and Peterson [9] for Reed-Solomon codes. This was generalized to arbitrary curves by Skorobogatov and Vlăduţ [11]. In this way one gets the *basic* and *modified* decoding algorithm [15].

In this note we will mainly concerned with the basic algorithm for Goppa codes which are defined from Hermitian curves $\mathcal{X}$ and are such that the support of the divisor G consists of just one $\mathbf{F}_q$-rational point of $\mathcal{X}$. These codes, called *one-point Hermitian* or simply *Hermitian* codes, will be introduced in Section 2. In Section 3 we describe the basic algorithm for such codes, whereas a MAGMA programme which realizes the algorithm is provided in Section 4.

2 One-point Hermitian codes

2.1 Linear codes

Let $\mathbf{F}_q$ be a finite field with q elements and $\mathbf{F}_q^n$ the vector space of n-tuples over $\mathbf{F}_q$. A q-ary linear code $\mathbf{C}$ of length n and dimension k is a k-dimensional subspace of $\mathbf{F}_q^n$. The number of non-zero positions in a vector $\mathbf{x} \in \mathbf{C}$ is called the Hamming weight $w(\mathbf{x})$ of $\mathbf{x}$; the Hamming distance $d(\mathbf{x}, \mathbf{y})$ between two vectors $\mathbf{x}, \mathbf{y} \in \mathbf{C}$ is defined by $d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} - \mathbf{y})$. The minimum distance of $\mathbf{C}$ is

$$d(\mathbf{C}) := \min\{w(\mathbf{x}) \mid \mathbf{x} \in \mathbf{C}, \mathbf{x} \neq 0\},$$

and a q-ary linear code of length n, dimension k and minimum distance d is indicated as an $[n, k, d]_q$ code. For such codes the Singleton bound holds:

$$d \leq n - k + 1.$$

A *generator matrix* of $\mathbf{C}$ is a $k \times n$ matrix whose rows form an $\mathbf{F}_q$-bases of $\mathbf{C}$.

The dual of a code $\mathbf{C}$ is indicated as $\mathbf{C}^\perp$, and it consists of all the vectors of $\mathbf{F}_q^n$ which are orthogonal to all codewords from $\mathbf{C}$, that is

$$\mathbf{C}^\perp := \{\mathbf{x} \in \mathbf{F}_q^n \mid \langle \mathbf{x}, \mathbf{y} \rangle = 0 \, \text{for any} \, \mathbf{y} \in \mathbf{C}\},$$

where $\langle , \rangle$ denotes the inner product in $\mathbf{F}_q^n$. If G is a generator matrix of $\mathbf{C}$, then

$$\mathbf{C}^\perp = \{\mathbf{x} \in \mathbf{F}_q^n \mid G\mathbf{x}^T = 0\},$$

where $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$; moreover, $\mathbf{C}^\perp$ has dimension $n - k$. A parity check matrix of a code is any generator matrix of its dual.

2.2 Goppa codes

Throughout this section C will be a curve defined over $\mathbf{F}_q$. For background facts on curves we refer to [3], [5], [10], [12].

We fix the following notation.

- $\mathbf{F}_q(C)$ denotes the field of $\mathbf{F}_q$-rational functions of C.
- For $f \in \mathbf{F}_q(C) \setminus \{0\}$, $\mathrm{div}(f)$ denotes the divisor associated to f.
- For E divisor of C, $\mathcal{L}$ is the following $\mathbf{F}_q$-vector space

  $$\mathcal{L} = \{f \in \mathbf{F}_q(C) \setminus \{0\} \mid E + \mathrm{div}(f) \geq 0\} \cup \{0\};$$

  moreover we let $\ell(E) = \dim_{\mathbf{F}_q}(\mathcal{L}(E))$.

Let $P_1, \ldots, P_n$ be n distinct $\mathbf{F}_q$-rational points of C and let G be a divisor of C defined over $\mathbf{F}_q$ and such that $v_{P_i}(G) = 0$ for $i = 1, \ldots, n$. Let e be the following $\mathbf{F}_q$-linear map

$$e : \mathcal{L}(G) \to \mathbf{F}_q^n, \qquad f \mapsto (f(P_1), \ldots, f(P_n)),$$

and set $D = P_1 + P_2 + \ldots + P_n$.

Definition 2.1 *The Goppa code associated with D and G is* $\mathbf{C}_{D,G} := e(\mathcal{L}(G))$.

Lemma 2.2 *Let $k := \dim_{\mathbf{F}_q}(\mathbf{C}_{D,G})$ and d be the minimum distance of $\mathbf{C}_{D,G}$. Then*

1. $k = \ell(G) - \ell(G - D)$;
2. $d \geq n - \deg(G)$.

Lemma 2.3 *Let g be the genus of C, and let k and d as above. Then*

1. *if $n > \deg(G)$ then $k = \ell(G)$; moreover, a generator matrix of $\mathbf{C}_{D,G}$ is given by*

   $$M := \begin{pmatrix} f_1(P_1) & \cdots & f_1(P_n) \\ \vdots & \vdots & \vdots \\ f_k(P_1) & \cdots & f_k(P_n) \end{pmatrix}$$

   where $f_1, \ldots, f_k$ is an $\mathbf{F}_q$-basis of $\mathcal{L}(G)$;

2. *if $n > \deg(G) > 2g - 2$, then $k = \deg(G) + 1 - g$.*

Definition 2.4 *If $G = \gamma P$ for some P $\mathbf{F}_q$-rational point of C and some $\gamma \in \mathbf{Z}$, then $\mathbf{C}_{D,G}$ is called* one-point Goppa code.

The parameters of a one-point Goppa code are closely related to the Weierstrass semigroups of the underlying curve.

Definition 2.5 *The Weierstrass semigroups $H(P)$ of C at a point $P \in C$ is defined as*

$$H(P) := \{a \in \mathbf{Z} \mid \exists f \in \mathbf{F}_q(C) \text{ such that } (f)_\infty = aP\},$$

where $(f)_\infty$ denotes the polar divisor of f.
The elements in $H(P)$ are called non-gaps *at P.*

We have that $\mathcal{L}(\gamma P) = \mathcal{L}(\gamma^* P)$ where γ^* is the biggest non-gap at P less than or equal to γ. Therefore for a one-point Goppa code it is usually assumed that the integer γ is a non-gap at P.

2.3 One-point Hermitian codes

We present here certain one-point Goppa codes constructed from the Hermitian curve $\mathcal{X}$ which is defined for q squared by the equation

$$Y^{\sqrt{q}}Z + YZ^{\sqrt{q}} = X^{\sqrt{q}+1}.$$

This curve is non-singular of genus $g = \sqrt{q}(\sqrt{q}-1)/2$ and it has $q\sqrt{q}+1$ $\mathbf{F}_q$-rational points. These points are $P_\infty = (0,1,0)$ and $P_{a,b} = (a,b,1)$ where $a \in \mathbf{F}_q$ and $b^{\sqrt{q}} + b = a^{\sqrt{q}+1}$. Set

$$D := \sum_{a \in \mathbf{F}_q, b^{\sqrt{q}}+b=a^{\sqrt{q}+1}} P_{a,b},$$

and

$$G := mP_\infty, \qquad m \in \mathbf{Z}.$$

We consider the one-point Goppa code

$$\mathbf{C}_m := \mathbf{C}_{D,G} \subseteq \mathbf{F}_q^n,$$

whose length is $n := q\sqrt{q}$. Let $x = X/Z$ and $y := Y/Z$ so that $y^{\sqrt{q}} + y = x^{\sqrt{q}+1}$. Thanks to the following proposition it is possible to calculate exactly the dimension and the minimum distance of both $\mathbf{C}_m$ and its dual (see [13] and [16]).

Proposition 2.6 *An $\mathbf{F}_q$-basis of $\mathcal{L}(G)$, $m \geq 0$, is given by*

$$\{x^i y^j \,:\, i\sqrt{q} + j(\sqrt{q}+1) \leq m,\, i \geq 0,\, 0 \leq j \leq \sqrt{q}-1\}$$

3 Decoding Hermitian codes

We keep the notation of the previous section. From now on, we let $\mathbf{C} = \mathbf{C}_m^{\perp}$, the dual of the one-point Goppa code constructed from the Hermitian curve as in Subsection 2.3.

Let d be the minimum distance of $\mathbf{C}$. Suppose that $\mathbf{x} \in \mathbf{C}$ is a transmitted codeword from which we receive $\mathbf{y} = \mathbf{x} + \mathbf{e}$. Notice that $\mathbf{x}$ is unique whenever $\mathbf{y}$ has distance at most $(d-1)/2$ to $\mathbf{C}$.

Definition 3.1 *The vector* $\mathbf{e} = (e_1, \ldots, e_n)$ *is called the* error vector *of* $\mathbf{y}$. *The* e_i*'s are called the* error values *of* $\mathbf{y}$ *and the weight of* $\mathbf{e}$ *is the* number of errors *of* $\mathbf{y}$. *The set* $\{i \in \{1, \ldots, n\} \mid e_i \neq 0\}$ *is the set of* error positions *of* $\mathbf{y}$.

Lemma 3.2 *[7, Prop. 6.1] Let H be a parity check matrix of $\mathbf{C}$. Suppose that $\mathbf{y} = \mathbf{x} + \mathbf{e}$, $\mathbf{x} \in \mathbf{C}$, and that $J \subseteq \{1, \ldots, n\}$ is a set with at most $d-1$ elements which contains the set of error positions. Then $\mathbf{e}$ is the unique solution of the following linear equations in $\mathbf{z} = (z_1, \ldots, z_n)$:*

$$H\mathbf{z}^T = H\mathbf{y}^T \qquad \text{and} \qquad z_k = 0 \quad \text{for all } k \notin J.$$

Proof. Certainly $\mathbf{e}$ satisfies the equations. Let $\mathbf{z}$ be another solution. Then $H(\mathbf{z} - \mathbf{e})^T = 0$ and so $\mathbf{z} - \mathbf{e} \in \mathbf{C}$. Moreover, the weight of $\mathbf{z} - \mathbf{e}$ is less than or equal to $\#J \leq d - 1$. Therefore, $\mathbf{z} = \mathbf{e}$. ◇

Now, let

$$H(P_\infty) = \{\rho_1 = 0, \rho_2, \ldots\}, \qquad m = \rho_l.$$

For $\mathbf{y} \in \mathbf{F}_q^n$, $i, j \in \mathbf{N}$ such that $\rho_i + \rho_j \leq \rho_l$ and $J \subseteq \{1, \ldots n\}$, we set

$$K_{ij}(\mathbf{y}) := \{f \in \mathcal{L}(\rho_j P_\infty) \mid \langle \mathbf{y}, e(fg) \rangle = 0, \text{ for all } g \in \mathcal{L}(\rho_i P_\infty)\},$$

and

$$L_j(J) := \{f \in \mathcal{L}(\rho_j P_\infty) \mid e(f)_k = 0 \text{ for all } k \in J\},$$

where $e(f)_k$ is the kth-coordinate of $e(f) = (f(P_1), \ldots, f(P_n))$.

Lemma 3.3 ([14]) *Let $\mathbf{y} = \mathbf{x} + \mathbf{e}$, $\mathbf{x} \in \mathbf{C}$, and let I be the set of error positions of $\mathbf{y}$. Then*

(1) $K_{ij}(\mathbf{y}) = K_{ij}(\mathbf{e})$;

(2) $L_j(I) \subseteq K_{ij}(\mathbf{y})$;

(3) $L_j(I) = K_{ij}(\mathbf{y})$ *provided that the minimum distance of $\mathbf{C}_{\rho_i}^{\perp}$ is greater than the weight of* $\mathbf{e}$.

Then the following proposition holds.

Proposition 3.4 ([14]) *Suppose that $l \geq 2g + 2$, where $g = \sqrt{q}(\sqrt{q} - 1)$ is the genus of $\mathcal{X}$. For l even, set $i = l/2$, $j = l/2 - g + 1$, $t = l/2 - g$; for l odd, set $i = (l-1)/2$, $j = (l+1)/2 - g + 1$, $t = (l-1)/2 - g$. Then*

i) $\rho_i + \rho_j \leq \rho_l$;

ii) *if* $\mathbf{y} = \mathbf{x} + \mathbf{e}$ *with* $\mathbf{x} \in \mathbf{C}$ *and* $w(\mathbf{e}) \leq t$, *then* $K_{ij}(\mathbf{y}) = L_j(I)$;

iii) $L_j(I) \neq \{0\}$;

iv) *For any* $f \in L_j(I)$, $\#\{k \mid f(P_k) = 0\} \leq d - 1$.

Therefore, we have the so-called *basic algorithm* for the code $\mathbf{C}$, i.e. giving $\mathbf{y} = \mathbf{x} + \mathbf{e}$ with $\mathbf{x} \in \mathbf{C}$ we can compute $\mathbf{e}$ whenever $w(\mathbf{e})$ is less than or equal to $l/2 - g$.

Basic algorithm.

Given $\mathbf{C} = \mathbf{C}_{\rho_l}$ with $l \geq 2g + 2$.

Step 1. Fix i and j fulfilling the hypothesis of Proposition 3.4. Calculate $\mathbf{F}_q$-basis for $\mathcal{L}(\rho_l P_\infty)$, $\mathcal{L}(\rho_i P_\infty)$ and $\mathcal{L}(\rho_j P_\infty)$.

Step 2. Once the (possibly altered) message $\mathbf{y}$ has been received, calculate a function f in $L_j(I) = K_{ij}(\mathbf{y})$, $f \neq 0$.

Step 3. Set $J = \{k \mid f(P_k) = 0\}$. Then calculate $\mathbf{e}$ such that

$$H\mathbf{e}^T = H\mathbf{y}^T \qquad \text{and} \qquad e_k = 0 \quad \text{for all } k \notin J.$$

Step 4. Put $\mathbf{x} = \mathbf{y} - \mathbf{e}$.

4 The MAGMA programme

4.1 Preliminary settings

Suppose that the prime power $\sqrt{q}$ is contained in the variable `sq`, and that $l \geq 2g + 2$ is contained in the variable `l`.

```
q:=sq^2;
K:=GF(q);
R<x,y>:=PolynomialRing(K,2);
n:=sq^3;
g:=((sq)*(sq-1)) div 2;
```

4.2 Step 1

First we construct a function which computes the dimension of an $\mathbf{F}_q$-space $\mathcal{L}(\mathrm{t}P_\infty)$. We refer to Proposition 2.6.

```
dim:=function(t)
   if t eq 0 then
      return 1;
   else
      base:={};
      for r in [0..t] do
          for s in [0..sq-1] do
              if r*sq+s*(sq+1) le t then
                 base:=base join {r*q+s*(q+1)};
              end if;
          end for;
      end for;
      return #base;
   end if;
end function;
```

Next we calculate ρ_l, i, j, ρ_i and ρ_j, and we put them in the variables `Rl, i, j, Ri, Rj` respectively.

```
nongaps:=[];
for x in [0..l+g-1] do
  if dim(x+1) eq dim(x)+1 then
    Append(~nongaps,x);
  end if;
end for;
Rl:=nongaps[l];
if IsEven(l) then
  i:=l div 2;
  j:=l div 2-g+1;
    else
  i:=(l-1) div 2;
  j:=(l+1) div 2 -g+1;
end if;
Ri:=nongaps[i];
Rj:=nongaps[j];
```

We denote by `BaseRl` an $\mathbf{F}_q$-base of $\mathcal{L}(\rho_l P_\infty)$, by `BaseRi` an $\mathbf{F}_q$-base of $\mathcal{L}(\rho_i P_\infty)$, and by `BaseRj` an $\mathbf{F}_q$-base of $\mathcal{L}(\rho_j P_\infty)$.

```
BaseRl:=[];
for s in [0..sq-1] do
  for r in [0..Rl] do
    if r*sq+s*(sq+1) le Rl then
      Append(~BaseRl,(x^r)*(y^s));
    end if;
  end for;
end for;
BaseRi:=[];
for s in [0..sq-1] do
  for r in [0..Rl] do
    if r*sq+s*(sq+1) le Ri then
      Append(~BaseRi,(x^r)*(y^s));
    end if;
  end for;
end for;
BaseRj:=[];
for s in [0..sq-1] do
  for r in [0..Rl] do
    if r*sq+s*(sq+1) le Rj then
      Append(~BaseRj,(x^r)*(y^s));
    end if;
  end for;
end for;
k:=#BaseRl;
```

By `Points` we denote the sequence of all $\mathbf{F}_q$-rational affine points of $\mathcal{X}$.

```
Points:=[];
for u in K do
  for v in K do
    if Evaluate(x^(sq+1)+y^sq+y,[u,v]) eq 0 then
      Append(~Points,[u,v]);
    end if;
  end for;
end for;
```

Finally we construct the parity check matrix of $\mathbf{C}$, denoted as `H`, its transpose `Ht` and an $\mathbf{F}_q$-basis of $\mathbf{C}$, denoted as `CodeBase`.

```
seq:=[];
k:=dim(Rl);
for I in [1..k] do
  for v in [1..n] do
    ing:=Evaluate(BaseRl[I],Points[v]);
    Append(~seq,ing);
  end for;
end for;
M1:=KMatrixSpace(K,k,n);
H:=M1![seq[t]:t in [1..k*n]];
Ht:=Transpose(H);
W1:=VectorSpace(K,k);
O:=W1![0:t in [1..k]];
Par,Gen:=Solution(Ht,O);
CodeBase:=Basis(Gen);
```

4.3 Step 2

Suppose that the sequence `Y` contains the message $y \in \mathbf{F}_q^n$. We have to find a non-zero function in $L_j(I) = K_{ij}(y)$, i.e. a non-zero $\mathbf{F}_q$-linear combination $f = \sum \alpha_i f_i$, $f_i \in$ `BaseRj` such that

$$\sum_i \alpha_i \left(\sum_{l=1\ldots n} y_l g(P_l) f_i(P_l) \right) = 0$$

for all g in `BaseRi` and for all f_i in `BaseRj`. Then to find the α_i's we have to solve an appropriate homogeneous linear system over $\mathbf{F}_q$.

```
comp:=[];
for t in [1..#BaseRi] do
  for s in [1..#BaseRj] do

sind:=&+[Evaluate(BaseRi[t],Points[v])*Evaluate(BaseRj[s],Points[
     Y[v]: v in [1..n]];
    Append(~comp,sind);
  end for;
end for;
M2:=KMatrixSpace(K,#BaseRi,#BaseRj);
X:=M2![comp[t]:t in [1..#BaseRi*#BaseRj]];
Xtr:=Transpose(X);
W2:=VectorSpace(K,#BaseRi);
```

```
Z:=W2![0:I in [1..#BaseRi]];
D,A:=Solution(Xtr,Z);

N:=Basis(A);
f:=R!&+[N[1][s]*BaseRj[s]:s in [1..#BaseRj]];
```

4.4 Step 3

We put the error positions in a variable J.

```
J:=[v:v in [1..n]|Evaluate(f,Points[v]) eq 0];
```

Note that

$$He^T = Hy^T \qquad \text{and} \qquad e_k = 0 \quad \text{for all } k \notin J$$

is equivalent to

$$eH2^T = yH^T \qquad \text{and} \qquad e_k = 0 \quad \text{for all } k \notin J,$$

where $H2$ is obtained by deleting the columns of H corresponding to the positions $k \notin J$. Therefore the error E can be calculated as follows.

```
seq:=[];
for v in J do
  for I in [1..k] do
    entr:=Evaluate(BaseRl[I],Points[v]);
    Append(~seq,entr);
  end for;
end for;
M3:=KMatrixSpace(K,#J,k);
H2t:=M3![seq[t]:t in [1..k*#J]];

W3:=VectorSpace(K,n);
YM:=W3![Y[I]:I in [1..n]];
YHt:=YM*Ht;

Err:=Solution(H2t,YHt);


E:=[K|];
```

```
u:=1;
for I in [1..n] do
  if I in J then
    E[I]:=Err[u];
    u:=u+1;
  else
    E[I]:=0;
  end if;
end for;
```

4.5 Step 4

Finally, the transimetted codeword `X` can be easily calculated.

```
X:=[K|];
for I in [1..n] do
  X[I]:=Y[I]-E[I];
end for;
```

References

[1] S. Arimoto, Encoding and decoding of p-ary group codes and the correction system, *Inform. Processing in Japan* **2** (1961), 320–325.

[2] Y. Driencourt, Some properties of elliptic codes over a field of characteristic 2, *Proc. AAECC-3, Grenoble 1985, Lect. Notes Comp. Sc.* **229** (1986), 185–193.

[3] W. Fulton, *Algebraic Curves. An introduction to Algebraic Geometry*, W.A. Benjamin, New York, 1969.

[4] V.D. Goppa, Algebraic-Geometric Codes, *Math. USSR-Izv.* **21**(1) (1983), 75–93.

[5] R. Hartshorne, *Algebraic Geometry*, Grad. Texts in Math. Vol. 52, Springer-Verlag, New York/Berlin, 1977.

[6] A. Havemose, Decoding algebraic geometric codes, Ph.D Dissertation, Danmarks Tekniske Hojskole, Denmark, 1989.

[7] T. Høholdt, J.H. van Lint and R. Pellikaan, *Agebraic geometric codes*, in Handbook of Coding Theory (V.S. Pless, W.C. Huffman and R.A. Brualdi Eds.), vol. 1, 871–961, Elsevier, Amsterdam 1998.

[8] J. Justesen, K.J. Larsen, H.E. Jensen A. Havemose and T. Høholdt, Construction and decoding of a class of algebraic geometric codes. *IEEE Trans. Inform. Theory* **35** (1989), 811–821.

[9] W.W. Peterson, Encoding and error-correction procedures for the Bose-Chauduri codes, *IRE Trans. Inform. Theory* **IT-6** (1960), 459–470.

[10] A. Seidenberg, *Elements of the Theory of Algebraic Curves*,

[11] A.N. Skorobogatov and S.G. Vlăduţ, On the decoding of algebraic-geometric codes, *IEEE Trans. Inform. Theory* **36** (1990), 1051–1060.

[12] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer Verlag, Berlin/Heidelberg/New York, 1993.

[13] H. Stichtenoth, A note on Hermitian codes over $GF(q^2)$, *IEEE Trans. Inform. Theory* **34**(5) (1988), 1345–1348.

[14] F. Torres, Notes on Goppa codes, *Quaderno del seminario di Geometrie Combinatorie G. Tallini* **136** (2000), Dipartimento di Matematica Istituto G. Castelnuovo, La Sapienza, Rome.

[15] M.A. Tsfasman and S.G. Vlăduţ, *Algebraic-Geometric Codes*, Kluwer, Amsterdam, 1991.

[16] K. Yang and P.V. Kumar, On the true minimum distance of Hermitian codes, *Coding theory and algebraic geometry*, Lecture Notes in Math. Vol. 1518, 99–107, Springer-Verlag, Berlin-Heidelberg, 1992.