
ALGEBRAIC GEOMETRIC CODES ON SURFACES

by

José Felipe Voloch & Marcos Zarzar

Abstract. — We study error-correcting codes constructed from projective surfaces over finite fields using the generalized Goppa construction. We obtain bounds for the minimal distance of these codes by understanding how the zero sets of functions on a surface decompose into irreducible components. We also present a decoding algorithm for these codes based on the Luby-Mitzenmacher algorithm for LDPC codes.

1. Introduction

Given a divisor G on a projective surface X , both defined over the finite field \mathbf{F}_q , and rational points P_1, \dots, P_n on X but not on G , we can define a code C consisting of the vectors $(f(P_1), \dots, f(P_n))$ where f varies in $L(G)$. This is a well-known construction, generalizing Goppa's classical construction for codes coming from curves over finite fields. Unlike curves, however, not much is known about codes from surfaces.

The natural questions are what are the possible (or best) length, dimension and minimal distance of these codes and decoding algorithms for them.

The problem of length is tied up with the problem of finding the number of points on surfaces over finite fields, of which much less is known than in the case of curves. For a survey up to 1996, see [4], for some recent results, see [5].

The dimension is the easiest to compute, once we know that the evaluation map $f \mapsto (f(P_1), \dots, f(P_n))$ is injective. Then it is just the dimension of $L(G)$ which can be estimated using the Riemann-Roch theorem for surfaces. Proving that the evaluation map is injective is often a by-product of lower bounds for the minimal distance of C .

Keywords: error-correcting codes, algebraic geometric codes, surfaces.

In this paper we focus our attention on bounds for the minimal distance of C and decoding algorithms.

2. Minimal distance

To obtain bounds for the minimal distance of C note that the weight of $(f(P_1), \dots, f(P_n))$ is bounded below by $n - \#Z_f(\mathbf{F}_q)$ where Z_f is the support of the divisor of zeros of f , so Z_f is a reduced curve defined over \mathbf{F}_q , if f is non-constant. Thus, we needed to understand how zero sets of functions on surfaces decompose in irreducible components. For instance, the following result can be proved as a natural extension of the main result of [1].

Theorem 2.1. — *A curve over \mathbf{F}_q of arithmetic genus g and with r irreducible components defined over \mathbf{F}_q has at most $r(q+1) + 2gq^{1/2}$ rational points.*

Of course, the above bound ultimately depends on the Weil bound. Alternatively, other bounds on the number of rational points on irreducible curves over finite fields can be used. Still, the number of irreducible components needs to be bounded, since it will greatly affect the ultimate bound for $\#Z_f(\mathbf{F}_q)$.

Here are some geometric results bounding the number of irreducible components.

Lemma 2.2. — *If the Néron-Severi group of X is generated by the ample divisor H and $G = mH$, then the zero set of a non-zero element of $L(G)$ has at most m irreducible components.*

Proof: If $f \in L(G)$, $(f)_0 = k_1A_1 + \dots + k_rA_r$, then A_i is algebraically equivalent to a_iH for some $a_i > 0$, by hypothesis. So

$$rH^2 \leq \sum k_i a_i H^2 = (f)_0 H = (f)_\infty H \leq mH^2,$$

so $r \leq m$.

For a sharper result for surfaces in \mathbf{P}^3 , see [6].

Lemma 2.3. — *Suppose that H is an ample divisor irreducible over the ground field but decomposing on a Galois extension of prime degree p as a sum of p conjugate irreducible components such that the intersection points are also moved by Galois. If $G = mH$, then the zero set of a non-zero element of $L(G)$ has at most mH^2/p absolutely irreducible components defined over the ground field.*

Proof: If $f \in L(G)$, $(f)_0 = k_1A_1 + \dots + k_rA_r + D$, where the A_i are irreducible and defined over the ground field, then $A_iH \geq p$ by the Galois action. So

$$rp \leq \sum k_i A_i H \leq (f)_0 H = (f)_\infty H \leq mH^2,$$

so $r \leq mH^2/p$.

A cubic surface in \mathbf{P}^3 over \mathbf{F}_9 with 100 points is constructed in [6] where both lemmas apply and H is a suitable plane section, thus $H^2 = 3$ and if $G = mH$, then the zero set of a non-zero element of $L(G)$ has at most m absolutely irreducible components defined over the ground field. For $m = 2$ this gives a code of length 100, dimension 10 and minimal distance 68.

For another example consider X , the zero set of

$$x^3 + y^3 + z^3 - zx^2 - xy^2 - yz^2 + xz^2 + w^3$$

in characteristic 3. Then 2.3 applies both over \mathbf{F}_3 and \mathbf{F}_9 with H the plane $w = 0$. Over \mathbf{F}_3 we can take $m = 1$ and get a code of length 13, dimension 4 and minimal distance 7, which is best possible. Over \mathbf{F}_9 we can take $m = 2$ and get a code of length 91, dimension 10 and minimal distance 61. In both cases the actual minimal distance is slightly bigger than what general estimates give. The results were obtained by computer calculation.

3. Decoding Algorithm

The decoding algorithm uses a divide-and-conquer technique and is based on the Luby-Mitzenmacher decoding algorithm presented in [3], section IV-B. The idea is to split the parity check conditions for C into several sets of parity check conditions for codes on curves on the surface.

Let X and $\{P_1, \dots, P_n\} \subset X$ as in 1. Let $Y_1, \dots, Y_r \subset X$ be curves on X , and let $\{P_i \mid i \in I_k\}$ be the set of points in $Y_k \cap \{P_1, \dots, P_n\}$. In practice, it is useful to minimize the size of the pairwise intersection of the curves Y_k . For instance, we fix a projective embedding of X and take the Y_k to be hyperplane sections the number of points on the intersection of any two such curves is bounded by the degree of the surface.

For each $k \in \{1, \dots, r\}$, define the map

$$\begin{aligned} \pi_k : C &\longrightarrow \mathbf{F}_q^{n_k} \\ (c_1, \dots, c_n) &\longmapsto (c_{l_1}, \dots, c_{l_{n_k}}) \end{aligned}$$

where $I_k = \{l_1, \dots, l_{n_k}\}$ and thus π_k is the projection of C on the coordinates that correspond to points in Y_k . Let M_k be a parity check matrix for $\pi_k(C)$ and define \bar{M}_k by

$$(\bar{M}_k)_{i,j} = \begin{cases} (M_k)_{i,j} & \text{if } j \in I_k; \\ 0 & \text{otherwise.} \end{cases}$$

Note that the parity check equations coming from \bar{M}_k are satisfied by the elements of C . Let M denote the matrix obtained by concatenating all the \bar{M}_k , so that the parity check equations coming from M are satisfied by the elements of C also. Assume we can choose Y_1, \dots, Y_r so that $\ker M = C$.

Unfortunately we cannot yet prove that such a set exists but in practice (we give an example below) such sets are plentiful. For future reference, we will say that $\{Y_1, \dots, Y_r\}$ satisfies the kernel condition if $\ker M = C$. Note that M has sparse rows if X has many more rational points than each Y_k and it has sparse columns when any pair of Y_k 's intersect in few points.

Under these conditions, the decoding algorithm is the following:

1. Given a word w , generate the subword $\pi_i(w)$, where i is chosen at will;
2. Use a decoding algorithm for codes on curves to decode $\pi_i(w)$, and replace the corresponding coordinates of w by those of the decoded word;
3. Repeat steps 1 and 2 as many times as necessary so that for each i in $\{1, \dots, r\}$, $\pi_i(w)$ is a word in $\pi_i(C)$.

Observe that the ideal condition in 3 might not always be satisfied, so it is useful to introduce a bound on the number of iterations or test for some stationary situation.

This procedure is close to that of [3] where it is described for LDPC codes over large alphabets, having projections contained in Reed-Solomon codes. In our case, codes from surfaces are often LDPC codes and have projections contained in codes from curves.

It is interesting to note that Goppa codes coming from curves are seldom LDPC, since their duals are also Goppa codes coming from curves and, as such, have large minimal distance, whereas the dual of an LDPC code has small minimal distance, by definition.

3.1. Experimental results. — Using a cubic surface in \mathbf{P}^3 over \mathbf{F}_9 and $G = 2H$, [6] constructs a code of length 100, dimension 10 and minimal distance 68. We used this code to run some simulations of the decoding algorithm. Using random search, we were able to find several sets of 15 plane sections satisfying the kernel condition (but no less than 15).

For comparison we did two searches for plane sections. The first was random and the second we only looked at plane sections with many points. Curves with few points will give codes with small minimal distance and poor error correcting capacity. Also, they contribute few parity check conditions, meaning that we will need more curves to satisfy the kernel condition, affecting the overall performance of the algorithm. We thus selected only curves with at least 15 points, which is very close to the Weil bound for such a curve.

On step 1 of the algorithm, we picked i randomly among the elements of the set $\{1, \dots, r\}$ but no repetition was allowed until all elements had been picked. We used random choice of i to possibly minimize undesired stationary situations.

For the step 2, we used the Zimmermann minimum weight algorithm [2] to decode the code on a curve. This algorithm is already implemented in the MAGMA package.

We ran experiments with 11000 words for each noise weight, and the results are shown in figures 1 and 2. For each k in $\{1, \dots, 34\}$ we computed the percentage of effective correction when a random noise of weight k was added to a random codeword. On figure 1, the circles correspond to the performances of the algorithm using curves with at least 15 points, and the crosses correspond to the performances when no non-trivial constraint was imposed on the number of points on each curve. The performances were substantially distinct. In figure 2 we compare the effectiveness of the algorithm according to how i (as in step 1) is picked. The circles correspond to random choice of i and the crosses to sequential choice of i , both using the plane sections with many points. As we can see, there was no noticeable difference on the effectiveness. Finally we remark that the decoding algorithm implemented in MAGMA was unable to perform decoding on this code.

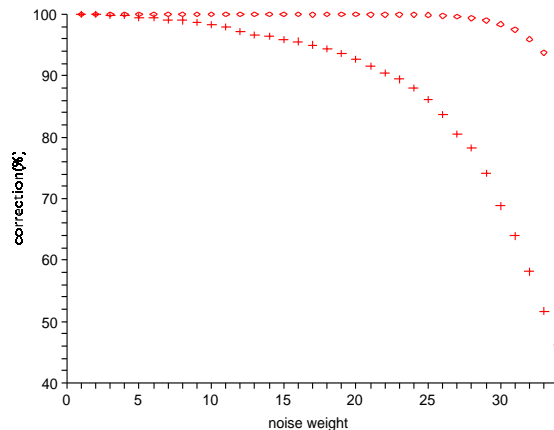
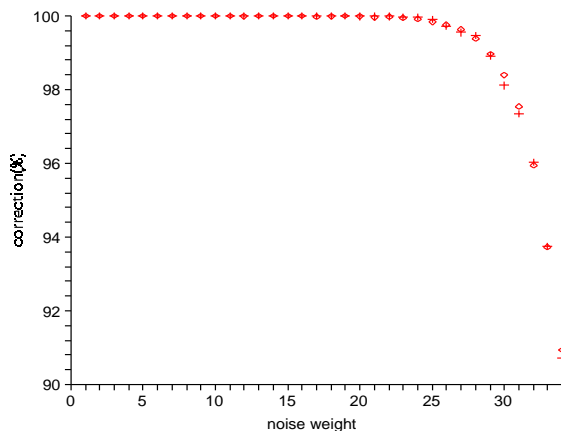


FIGURE 1. Curves with many points versus any curves

Acknowledgments

The computations in this paper were made using the MAGMA Computational Algebra System.

FIGURE 2. Sequential versus random choice of i

References

- [1] Y. Aubry; M. Perret, A Weil theorem for singular curves. Arithmetic, geometry and coding theory (Luminy, 1993), 1–7, de Gruyter, Berlin, 1996.
- [2] A. Betten, H. Fripertinger, A. Kerber, A. Wassermann, and K.-H. Zimmermann, Codierungstheorie - Konstruktion und Anwendung linearer Codes, Springer-Verlag, Berlin-Heidelberg-New York, 1998.
- [3] M. G. Luby, M. Mitzenmacher, Verification-Based Decoding for Packet-Based Low-Density Parity-Check Codes, IEEE Transactions on Information Theory, vol. 51, no. 1, 2005.
- [4] M. A. Tsfasman, Nombre de points des surfaces sur un corps fini. Arithmetic, geometry and coding theory (Luminy, 1993), 209–224, de Gruyter, Berlin, 1996.
- [5] J. F. Voloch, Surfaces in \mathbf{P}^3 over finite fields, in Topics in Algebraic and Non-commutative Geometry: Proceedings in Memory of Ruth Michler, C. Melles et al. eds., Contemporary Math. 324 (2003) 219–226.
- [6] M. Zarzar, Error correcting codes on low rank surfaces, preprint available at <http://www.ma.utexas.edu/users/zarzar/>

JOSÉ FELIPE VOLOCH, Dept. of Mathematics, Univ. of Texas, Austin, TX 78712-0257

E-mail : voloch@math.utexas.edu

MARCOS ZARZAR, Dept. of Mathematics, Univ. of Texas, Austin, TX 78712-0257

E-mail : zarzar@math.utexas.edu