Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 134 (2009)

# On matrix rigidity and locally self-correctable codes

Zeev Dvir [*]

**Abstract**

We describe a new approach for the problem of finding rigid matrices, as posed by Valiant [Val77], by connecting it to the, seemingly unrelated, problem of proving lower bounds for linear locally self-correctable codes. This approach, if successful, could lead to a non-natural property (in the sense of Razborov and Rudich [RR97]) implying super-linear lower bounds for linear functions in the model of logarithmic-depth arithmetic circuits.

Our results are based on a lemma saying that, if the generating matrix of a locally decodable code is **not** rigid, then it defines a locally self-correctable code with rate close to one. Thus, showing that such codes cannot exist will prove that the generating matrix of *any* locally decodable code (and in particular Reed Muller codes) is rigid.

## 1 Introduction

One of the main challenges of computational complexity is proving lower bounds for interesting families of functions in natural models of computation and in particular *circuit* lower bounds. The 'holy grail' is to show that there are languages in NP that cannot be computed by polynomial size boolean circuits, thus separating $P$ from $NP$. One can also formulate this question for arithmetic circuits (circuits using basic field operations that compute polynomials) and try to prove that there are polynomials in VNP (Valiant's arithmetic analog of NP [Val79]) that do not have polynomial size arithmetic circuits. A notable obstacle for pursuing this goal (at least in the boolean setting) was given by Razborov and Rudich [RR97] who showed that a large class of possible proof strategies, called *natural proofs*, cannot separate $P$ from $NP$.

A special case of the above challenge is to prove circuit lower bounds for *linear* functions. Say $A : \mathbb{F}^n \mapsto \mathbb{F}^n$ is a linear mapping over a field $\mathbb{F}$. A natural way to compute this mapping on a given input is by an arithmetic circuit whose gates compute linear functions

[*] Institute for advanced study, Princeton, NJ. Research partially supported by NSF grants CCF-0832797 and DMS-0835373.

ISSN 1433-8092

of two inputs. That is, we start with the inputs $x_1, \ldots, x_n$ and, at each gate, compute a linear combination of two terms already computed. Eventually we would like to reach a point in the computation where all $n$ outputs of $A$ have been computed. The *size* of the circuit is the number of gates used and its *depth* is the longest path from an output to an input (when we view the computation as a directed acyclic graph). Even though it is easy to show that almost all linear mappings require nearly quadratic size circuits, no explicit examples of such mappings are known. Even more surprising, after more than three decades of attempts, there are no explicit examples of linear mappings that cannot be computed by circuits of size $O(n)$ and depth $O(\log(n))$. Of particular interest are mappings such as the discrete fourier transform which are used heavily in practice and for which the existence of linear size circuits would have an enormous impact.

In the late 70's Valiant [Val77] defined the notion of *matrix rigidity* and showed that a linear mapping defined by a rigid matrix cannot be computed by linear size and logarithmic depth circuits. In this work we give a further reduction from the problem of finding rigid matrices to the, seemingly unrelated one, of proving lower bounds for a special kind of error correcting codes. Informally, we show that lower bounds for linear locally self-correctable codes will imply rigidity for a large (and explicit) family of matrices. More specifically, *any* matrix which is the generating matrix of a linear locally-decodable code (with sufficiently good parameters) will be rigid. In particular, our approach (if successful) can potentially lead to showing that computing constant rate Reed Muller encodings cannot be done with a linear size and logarithmic depth circuit. Since the property of being a good locally decodable code is not a 'natural' property our reduction raises the possibility of proving 'non-natural' circuit lower bounds [1]. The lower bounds we require for locally self-correctable codes are for a range of parameters not considered before (large number of queries and rate close to one) and we find them to be consistent with current knowledge.

We start by defining matrix rigidity and locally decodable/correctable codes and then proceed to give the formal statement of our results.

## 1.1 Matrix rigidity

In the following $\mathbb{F}$ denotes a field. We denote by $M_{m \times n}(\mathbb{F})$ the vector space of matrices with $m$ rows and $n$ columns with entries in $\mathbb{F}$. We call a matrix $S \in M_{m \times n}(\mathbb{F})$ *$s$-sparse* if each row in $S$ contains at most $s$ non zero entries. All logarithms are taken in base 2.

**Definition 1.1** (Matrix rigidity)**.** *Let $A \in M_{m \times n}(\mathbb{F})$. We say that $A$ is $(r, s)$-*rigid *if $A$ cannot be written as a sum of two matrices $A = L + S$ such that $L$ has rank at most $r$ and $S$ is $s$-sparse.*

[1] The notion of natural properties and natural proofs were defined in [RR97] only for boolean functions but one can make the analogy in the linear setting by defining a natural property of a matrix to be a property that holds for most matrices and can be verified efficiently.

In other words, a matrix $A$ is $(r, s)$-rigid if we cannot decrease its rank to less than $r$ by changing at most $s$ of its entries in every row. We refer the reader to the excellent survey [Lok09] for a complete discussion of past work on matrix rigidity. Our definition of rigidity is slightly non-standard since it allows $m$ and $n$ to be different. This formulation, which was used also in the recent work of Alon et al [APY09], allows one to think of a rigid $m \times n$ matrix as a set of (row) vectors such that, for any $r$-dimensional subspace, there exists a vector in the set which is of Hamming distance at least $s$ from the subspace. This formulation allows us also to fix the sparsity parameter, $s$, and to try and minimize $m$ as a function of $n$ (hopefully reaching $m = O(n)$). It is trivial to construct an $(n/2, \Omega(n))$-rigid matrix with $m = \exp(n)$ by taking as the rows of the matrix all vectors in $\mathbb{F}^n$. The task becomes significantly harder if one tries to get the same rigidity with $m = O(n)$ or even $m = 2^{o(n)}$.

Most of the interest in constructing explicit rigid matrices comes from the following theorem of Valiant connecting rigidity to arithmetic circuit complexity [2].

**Theorem 1.2** (Valiant [Val77]). *Suppose $A \in M_{m\times n}(\mathbb{F})$ is $(\alpha n, n^\beta)$-rigid for constant $\alpha, \beta > 0$ and $m = O(n)$. Then a linear arithmetic circuit $C : \mathbb{F}^n \mapsto \mathbb{F}^m$ computing $C(x) = A \cdot x$ cannot have both size $O(n)$ and depth $O(\log(n))$.*

The best explicit upper bound on $m$ in terms of $n$ and $s$ (we fix $r$ to be $n/2$ for convenience) is by Alon et al [APY09] and gives, for every $s$, explicit $(n/2, s)$-rigid matrices with

$$m \leq n \cdot \exp(s).$$

Non explicitly, one can easily show the existence of $(n/2, \Omega(n))$-rigid matrices with $m = n$. Over fields of characteristic zero, some constructions of 'semi-explicit' matrices are known that have almost optimal rigidity. For example, the square matrix with entries $\sqrt{p_{ij}}$, with $p_{ij}$ the first $n^2$ primes, is known to be $(\epsilon n, \Omega(n))$-rigid [Lok06]. These matrices, while having a nice compact mathematical description, are not strongly explicit in the sense that their entries require infinite (or exponential) precision in bits.

We now formulate two open problems (stated as a strong and a weak version of the same problem) related to the construction rigid matrices. A solution to the strong version will result in super-linear circuit lower bounds (as was shown by Valiant). A solution to the weak version of the problem, while not saying anything about circuits, will still signify, in our opinion, a major breakthrough on the way to proving matrix rigidity. We will define 'explicit' to be any family of matrices that can be generated in polynomial time by a deterministic Turing machine. Since we will be mostly concerned with finite fields, we can, for now, not worry about issues of precision and bit representations (we will discuss infinite fields only in Section 5).

**Problem 1.** *Give an explicit family $A_n \in M_{m(n)\times n}(\mathbb{F})$ (for infinitely many $n$'s) such that:*

[2] Valiant's proof actually allows $\alpha$ and $\beta$ in the theorem to be slightly sub-constant.

- *(Strong version)* $m(n) = O(n)$ *and* $A_n$ *is* $\left(\alpha n, n^{\beta}\right)$*-rigid for constants* $\alpha, \beta > 0$.
- *(Weak version)* $m(n) = poly(n)$ *and* $A_n$ *is* $\left(\alpha n, \log^{1+\beta}(n)\right)$*-rigid for constants* $\alpha, \beta > 0$.

Notice that if we replace $\log^{1+\beta}(n)$ with $O(\log(n))$ in the weak version, we are already in the range of parameters obtained by [APY09] and so the weak version of Problem 1 represents the 'barrier' to current techniques. We now turn to discuss locally decodable/correctable codes and their connection to the above discussion.

## 1.2 Locally decodable and self-correctable codes

We will be interested only in *linear* codes. All of the definitions in this section can, however, be extended also to the non linear case. For a vector $v \in \mathbb{F}^n$ we denote by $w(v)$ the number of non zero entries in $v$ (i.e its Hamming weight). We start by defining locally decodable codes. Informally, an LDC is an error correcting codes that allows one to recover a single message symbol by reading a small number of entries in a corrupted encoding. These codes were first formally defined in the work of Katz and Trevisan [KT00].

**Definition 1.3** (Locally decodable code (LDC)). *A* $(q, \delta, \epsilon)$*-LDC over a field* $\mathbb{F}$ *is a linear mapping* $C : \mathbb{F}^n \mapsto \mathbb{F}^m$ *such that there exists a probabilistic procedure* $D : \mathbb{F}^m \times [n] \mapsto \mathbb{F}$ *with the following properties:*

1. *For all* $x \in \mathbb{F}^n$*, for all* $i \in [n]$ *and for all* $v \in \mathbb{F}^m$ *with* $w(v) \leq \delta m$ *we have that* $D\left(C(x) + v, i\right) = x_i$ *with probability at least* $1 - \epsilon$ *(the probability is taken only over the internal randomness of* $D$*).*
2. *For every* $y \in \mathbb{F}^m$ *and* $i \in [n]$*,* $D(y, i)$ *reads at most* $q$ *positions in* $y$*.*

*We refer to the* $m \times n$ *matrix computing the mapping* $C$ *as the* generating matrix *of the code.*

LDC's are very useful (both in practice and in theory) and their properties are the subject of many works (see the survey article [Tre04] for more details). For a long time the best constructions of LDC's were variants of Reed Muller (RM) codes [HB98]. These are codes based on multivariate polynomials where the queries are taken, in general, from a random line through a point (see Section 3 for details). In recent years there have been a number of works giving new and surprising upper bounds for constant query LDC's [BIKR02, Yek08, Efr09]. However, despite this progress, when the number of queries is roughly logarithmic in $n$, the best upper bounds are still obtained using RM based codes.

Exponential lower bounds (i.e $m \geq \exp(n)$) for LDC's were proven for two-query codes in [GKST06, KdW04] both in the linear and the non-linear case. When $q > 2$ the only known lower bounds are slightly super-linear. The first bound for $q > 2$ is due to Katz and Trevisan [KT00] and is of the form $m = \Omega\left(n^{1+1/(q-1)}\right)$. Notice that this bound deteriorates rapidly when when $q$ approaches $\log(n)$. In [KdW04, Woo07] this bound was improved slightly, for constant $q$, to $m = \Omega\left(n^{1+2/(q-2)}\right)$.

We now define locally self-correctable codes. These codes have stronger properties than LDC's and one can even verify formally that an LCC implies and LDC with similar parameters. In an LCC we want to locally decode not message symbols but rather *code word* symbols. Since there are no message symbols in the definition, it is easier to think of the code as a subspace.

**Definition 1.4** (Locally self-correctable code (LCC)). *A $(q, \delta, \epsilon)$-LCC is a subspace $C \subset \mathbb{F}^m$ such that there exists a probabilistic procedure $D : \mathbb{F}^m \times [m] \mapsto \mathbb{F}$ with the following properties:*

1. *For all $x \in C$, for all $i \in [m]$ and for all $v \in \mathbb{F}^m$ with $w(v) \leq \delta m$ we have that $D(x + v, i) = x_i$ with probability at least $1 - \epsilon$ (the probability is taken only over the internal randomness of $D$).*

2. *For every $y \in \mathbb{F}^m$ and $i \in [m]$, $D(y, i)$ reads at most $q$ positions in $y$.*

*The* dimension *of an LCC is simply its dimension as a subspace of $\mathbb{F}^m$.*

LCC's originated in works on program checking [BK95, Lip90] and, as LDC's, have found many applications. Unlike LDC's, the known techniques for constructing LCC's amount to basically using RM codes. The more 'modern' LDC techniques that beat RM do not seem to give LCC's. In fact, at our current state of knowledge, it is certainly possible that RM codes give the best parameters for LCC's (at least asymptotically). One work that addresses this issue and connects it to well-studied conjectures in design theory is [BIW07].

Even though LCC's are stronger objects than LDC's, the lower bounds known for them are the same ones known for LDC's. In particular, there are no non trivial bounds when $q > log(n)$.

## 1.3 Our results

Our main theorem, stated below, shows that certain lower bounds on the encoding length of LCC's will imply a solution to the rigidity problem (Problem 1). Since we believe these lower bounds to be true we state them in the form of a conjecture.

**Conjecture 1.** *There exist constants $\alpha, \beta, \gamma, \epsilon, > 0$ such that, for sufficiently large $n$:*

- *(Strong version) There does not exist an $\left(n^\alpha, 1/n^\beta, \epsilon\right)$-LCC $C \subset \mathbb{F}^n$ with dimension $(1 - \gamma)n$.*

- *(Weak version) There does not exist a* $\left(\log^{2+\alpha}(n), 1/\log^{1+\beta}(n), \epsilon\right)$*-LCC* $C \subset \mathbb{F}^n$ *with dimension* $(1-\gamma)n$.

**Theorem 1.** *A proof of the strong (weak) version of Conjecture 1 over a finite field* $\mathbb{F}$ *will solve the strong (weak) version of Problem 1 over the same field. Moreover, the required rigid matrices can be taken to be the generating matrices of* **any** *LDC which matches the parameters of Reed Muller codes.*

The proof of the theorem will rely on one main lemma (proved in Section 2) that shows that, if the generating matrix of an LDC is *not* rigid, then one can construct an LCC from it, with rate close to one and with similar query complexity. The main degeneration is in the error parameter $\delta$, which explains the need for sub-constant $\delta$ in the two variants of Conjecture 1. Notice, however, that formulating the conjectures with constant $\delta$ makes them trivially true, since a code (LCC or not) that can correct a constant fraction of errors cannot have rate arbitrarily close to one. So our conjecture can be rephrased as saying that the restriction of being locally correctable prevents the code from having optimal rate/distance dependency.

This brings us to the question of whether we should believe Conjecture 1 to be true or not (in any of its variants). It is quite easy to be convinced that known constructions of LCC's, based on low degree polynomials, do not contradict the conjecture. To go deeper we recall the lower bound for LCC's with $q > 2$ (in fact for LDC's) given by Katz and Trevisan. Their theorem was stated originally only for binary codes, but their techniques extend also for any constant size field (at least for linear codes).

**Theorem 1.5** (Rephrased from [KT00]). *Let* $C \subset \mathbb{F}^n$ *be a* $(q, \delta, \epsilon)$*-LCC. Then*

$$\dim(C) \leq O\left((\epsilon \cdot \delta)^{-\frac{1}{q}} \cdot n^{1-\frac{1}{q}}\right).$$

Notice that setting $q = \alpha \cdot \log(n)$ and $\delta = n^{-0.99}$ (and thinking of $\epsilon$ as a constant) we still get an upper bound of $c(\alpha) \cdot n$ with $c(\alpha) \mapsto 0$ as $\alpha \mapsto 0$. Thus, in the range of $q = \Omega(\log(n))$ we have a sufficiently good (for our purposes) upper bound on the dimension of LCC's, even for extremely small values of $\delta$. If we consider the weak version of Conjecture 1 we see that there $\delta$ can be taken to be much larger (inverse of poly-log instead of inverse polynomial) at the cost of increasing $q$ from logarithmic to poly-logarithmic[3]. This seems to indicate that the value of $\delta$, be it constant or sub constant, might not play as important of a role as does the query complexity.

Since the range of parameters appearing in Conjecture 1 is one that was not looked at before, it could be that there is some undiscovered construction of *LCC*'s that disproves

[3] It would have been nice to have $q = \log^{1+\alpha}(n)$ instead of $q = \log^{2+\alpha}(n)$ in the weak version of our conjecture. This would have paralleled the situation in Problem 1 where the weak version represents the 'barrier' to current techniques.

the conjecture. If true, we would find that to be very interesting since the only construction of LCC's we know of are based on RM codes and these do not seems to come close to achieving rate one (for any $\delta$).

The rest of the paper is organized as follows: In Section 2 we prove our main lemma mentioned above. We proceed in Section 3 to establish some (well known) upper bounds on LDC's based on RM codes. We prove Theorem 1 in Section 4 and conclude in Section 5 with a discussion and some directions for future work.

# 2 The main lemma

The main ingredient in the proof of Theorem 1 is the following lemma which shows that, if the generating matrix of an LDC is not rigid, then there exists an LCC with rate close to one, and with certain bounds on its query complexity and error parameter.

**Lemma 2.1** (Main Lemma)**.** *For every real $\rho > 0$ the following holds: Let $C : \mathbb{F}^n \mapsto \mathbb{F}^m$ be a $(q, \delta, \epsilon)$-LDC and let $A_C \in M_{m \times n}(\mathbb{F})$ be its generating matrix. If $A_C$ is not $(r, s)$-rigid then there exists a $(qs, (\rho\delta)/s, \epsilon)$-LCC $C' \subset \mathbb{F}^n$ of dimension at least $n(1 - \rho) - r$.*

The outline of the proof is as follows: Suppose $A_C$ is not rigid. Then, the mapping $C$ can be 'approximated' by a sparse mapping $S$ (a mapping in which each output depends on a small number of inputs) in the sense that there exists a large subspace on which $C$ agrees with $S$. Next, we observe that this subspace is an $LCC$, since we can correct each coordinate in a corrupted code-word $y$ by invoking the local decoder for $C(y)$ and simulating each query to $C(y)$ using $s$ queries to the original string $y$. A detailed proof follows.

## 2.1 Proof of Lemma 2.1

**Preprocessing:** Suppose $A_C$ is not $(r, s)$-rigid. Then it can be written as a sum

$$A_C = L + S \tag{1}$$

with rank$(L) \leq r$ and such that $S$ is $s$-sparse ($S$ has at most $s$ non zeros in every row). We would like $S$ to have the additional property of being sparse also in each column. We can get this by 'moving' all the columns with too many elements to $L$. Since there cannot be too many 'heavy' columns, we will need to modify $L$ in a small number of columns, thus increasing its rank by a small factor. More formally: The total number of non zeros in $S$ is at most $s \cdot m$. This means that the average number of non zeros in a column of $S$ is at most $(s \cdot m)/n$. By Markov's inequality, we have that there are at most $\rho \cdot n$ columns with more than $(s \cdot m)/(n \cdot \rho)$ non zeros. In view of Eq. 1, We can replace these columns with zero columns in $S$ and move them to $L$, increasing its rank by at most $\rho \cdot n$. Thus, after this modification we have

$$A_C = L' + S' \tag{2}$$

with $\mathrm{rank}(L') \leq r + \rho \cdot n$ and such that $S'$ has at most $s$ non zeros in every row and at most $(s \cdot m)/(\rho \cdot n)$ non zeros in every column.

**Defining the code $C'$:** We let

$$C' \triangleq \ker(L') \subset \mathbb{F}^n.$$

Notice that $C'$ satisfies

$$\forall x \in C', \;\; C(x) = S' \cdot x. \tag{3}$$

we now turn to showing that $C'$ is a $(qs, (\rho\delta)/s, \epsilon)$-LCC. This will prove the lemma since we already know that

$$\dim(C') = n - \mathrm{rank}(L') \geq n(1-\rho) - r.$$

**Local correction procedure for $C'$:** Let

$$D : \mathbb{F}^m \times [n] \mapsto \mathbb{F}$$

be a decoding procedure for the code $C$ satisfying the two conditions of Definition 1.3. We need to describe a local correcting procedure

$$D' : \mathbb{F}^n \times [n] \mapsto \mathbb{F}$$

for $C'$. Let

$$\delta' \triangleq \frac{\rho \cdot \delta}{s}.$$

Let $x \in C'$ and let $v \in \mathbb{F}^n$ be such that $w(v) \leq \delta' n$. Given $i \in [n]$ and query access to $x + v$, $D'$ will need to recover $x_i \in \mathbb{F}$ by looking at at most $q \cdot s$ positions of $x + v$. We start with a simple claim

**Claim 2.2.** *Let $v' = S' \cdot v$. Then*

$$w(v') \leq \delta \cdot m$$

.

*Proof.* We know that the matrix $S'$ has at most $(s \cdot m)/(\rho \cdot n)$ non zeros in every column. This means that every index in $v$ can influence at most $(s \cdot m)/(\rho \cdot n)$ positions in $S' \cdot v$. Therefore, since $w(v) \leq \delta' \cdot n$ we have that

$$w(v') \leq (\delta' \cdot n) \cdot \frac{s \cdot m}{\rho \cdot n} = \left(\frac{\rho \cdot \delta}{s} \cdot n\right) \cdot \frac{s \cdot m}{\rho \cdot n} = \delta \cdot m.$$

□

From Claim 2.2 we see that, given query access to $C(x) + v'$, $D$ can recover $x_i$ (w.p $1 - \epsilon$) using at most $q$ queries. The crucial observation is that we can simulate a single query to $C(x) + v'$ using at most $s$ queries to $x + v$. This is because, by Eq. 3,

$$C(x) + v' = S' \cdot x + S' \cdot v = S' \cdot (x + v)$$

and since computing the $j$'th coordinate of the product of $S'$ with a vector can be done by looking at at most $s$ positions (since $S'$ has at most $s$ non zero entries in every row). Therefore, $D'$ can simulate $D$ on the input $(C(x) + v', i)$ using $q \cdot s$ queries to $x + v$ and we already know that, on this input, $D$ will return $x_i$ with probability at least $1 - \epsilon$. □

# 3 LDC's using low degree extensions

In order to use Lemma 2.1 to prove Theorem 1 we need to use certain (well known) constructions of LDC's. Since the parameters we require are quite specific, and for the sake of completeness, we sketch below the way to get these upper bounds. These are simple low degree extension codes concatenated with good linear error correcting codes. Readers familiar with these constructions can safely skip to the next section.

**Lemma 3.1** (Low-degree-extension codes). *Let $\mathbb{F}$ be a finite field of size $t$. For every $\epsilon > 0$ there exists $\delta = \delta(\epsilon) > 0$ such that for every sufficiently large integers $d$ and $k$ there exists a $(q, \delta, \epsilon)$-LDC*

$$C : \mathbb{K}^n \mapsto \mathbb{K}^m$$

*with $n = d^k$, $m \leq (2tdk)^k$, $q \leq 2tdk$ and with $\mathbb{K}$ being an extension field of $\mathbb{F}$ of size $t^\ell$ with $\ell \leq \log(q)$. Moreover, the generating matrices of these codes can be generated (deterministically) in time polynomial in $m$ given $d, k$ and $\epsilon$.*

*Proof.* Let $\ell$ be the smallest integer such that $t^\ell \geq 2dk$ and let $\mathbb{K}$ be the (unique) extension field of $\mathbb{F}$ of size $t^\ell$. We will set

$$m = |\mathbb{K}|^k = \left(t^\ell\right)^k \leq (2dtk)^k.$$

Notice that, since $|\mathbb{K}| \leq m$, we can construct the field $\mathbb{K}$ in (deterministic) time $\mathrm{poly}(|\mathbb{K}|) \leq \mathrm{poly}(m)$.

The code $C$ is defined as follows: Let $H \subset \mathbb{K}$ be some fixed set of size $d$. Since

$$n = d^k = |H|^k,$$

we can view each message $x \in \mathbb{K}^n$ as a function $F_x : H^k \mapsto \mathbb{K}$. It is well known that for each such $F_x$ there is a unique polynomial $g_x \in \mathbb{K}[u_1, \ldots, u_k]$ with individual degrees at most $d - 1$ such that $g_x(u) = F_x(u)$ for all $u \in H^k$. The encoding $C(x) \in \mathbb{K}^m$ is the list of evaluations of this polynomial $g_x$ on the entire space $\mathbb{K}^k$. The encoding matrix can be generated efficiently using standard results on multivariate polynomial interpolation.

To decode a message symbol $x_i = F_x(u), u \in H^k$ from a corrupted encoding $C(x) + v$ with $w(v) \leq \delta m$ we first pass a random line through $u \in \mathbb{K}^k$ and consider the restriction of $g_x$ to this line. This restriction is a univariate polynomial of degree at most $k(d-1)$ and so, since $|\mathbb{K}| \geq 2dk$, we can use the Berlekamp-Welch algorithm [MS77] to correct the values on the line (and so recover $g_x(u) = F_x(u)$) as long as their number is smaller than, say $dk/4$. Since the line was random, this will happen with probability at least $1 - \epsilon$ as long as we take $\delta$ to be sufficiently small (say $\delta < \epsilon/8$).

The number of queries done by the decoder is at most $|\mathbb{K}| \leq 2tdk$ as was required. The bound $\ell \leq \log(q)$ follows from the inequality $t^\ell \leq 2tdk = q$. □

The next lemma shows that we can replace the extension field $\mathbb{K}$ from Lemma 3.1 with the original field $\mathbb{F}$ at a small additional cost.

**Lemma 3.2.** *Let $C : \mathbb{K}^n \mapsto \mathbb{K}^m$ be a $(q, \delta, \epsilon)$-LDC and let $\mathbb{F}$ be a subfield of $\mathbb{K}$ such that $|\mathbb{K}| = |\mathbb{F}|^\ell$. Then there exists a $(q\ell, \Omega(\delta), \epsilon)$-LDC $C' : \mathbb{F}^{n'} \mapsto \mathbb{F}^{m'}$ with $n' = n\ell$ and $m' = O(m\ell)$. Moreover, the generating matrix of $C'$ can be generated in deterministic polynomial time from the generating matrix of $C$.*

*Proof sketch.* We can view $C$ as an $\mathbb{F}$-linear code mapping $\mathbb{F}^{n\ell}$ to $\mathbb{F}^{m\ell}$ (this is possible because $\mathbb{F}$ is a sub-field of $\mathbb{K}$). Now, concatenate $C$ with any good linear error correcting code $E : \mathbb{F}^\ell \mapsto \mathbb{F}^{O(\ell)}$ which can correct a constant fraction of errors (such codes can be constructed explicitly). Call the concatenated code $C'$. To decode a message symbol in $C'$ we will recover the entire 'block' (in $\mathbb{F}^\ell$) to which it belongs. This is done by invoking the decoder for $C$, and for each one of its queries (over $\mathbb{K} = \mathbb{F}^\ell$) correcting this query using the decoding for $E$ (we omit the details since this is standard). The bound on the success probability is obtained using Markov's inequality which implies that at most a $\delta$ fraction of the blocks of the encoding (each of size $2\ell$) will have more errors than the inner code $E$ can handle. □

Combining the two previous lemmas we obtain the following two corollaries which gives us the parameters we will require.

**Corollary 3.3** (LDC with $q = n^\alpha$). *Let $\mathbb{F}$ be a finite field. For every $\alpha, \epsilon > 0$ there exists $\delta = \delta(\epsilon) > 0$ and an explicit family of $(n^\alpha, \delta, \epsilon)$-LDC's $C_n : \mathbb{F}^n \mapsto \mathbb{F}^m$ with $m = O(n)$, where $n$ ranges over an infinite set of integers.*

*Proof.* Start by applying Lemma 3.1 with

$$k = \lceil 4/\alpha \rceil$$

to get a $(q', \delta, \epsilon)$-LDC $C' : \mathbb{K}^{n'} \mapsto \mathbb{K}^{m'}$ such that $n' = d^k$, $m' \leq (2tdk)^k \leq O(n')$ and with $q' \leq 2tdk \leq (n')^{(\alpha/2)}$ (notice that $t = |\mathbb{F}|$ can be viewed as a constant). The field $\mathbb{K}$ is an extension of $\mathbb{F}$ of degree $\ell \leq \log(q)$. Now, using Lemma 3.2, we replace $\mathbb{K}$ with $\mathbb{F}$ and get a code $C : \mathbb{F}^n \mapsto \mathbb{F}^m$ with $n = \ell n'$ and $m = \ell m'$. Therefore, we still have $m = O(n)$.

The number of queries grew from $q = (n')^{(\alpha/2)}$ to $q\ell \leq q \cdot \log(q)$ which is at most $n^\alpha$ for sufficiently large $n$. $\square$

**Corollary 3.4** (LDC with $q = \log^{1+\alpha}(n)$)**.** *Let $\mathbb{F}$ be a finite field. For every $\alpha, \epsilon > 0$ there exists $\delta = \delta(\epsilon) > 0$ and an explicit family of $\left(\log^{1+\alpha}(n), \delta, \epsilon\right)$-LDC's $C_n : \mathbb{F}^n \mapsto \mathbb{F}^m$ with $m = poly(n)$, where $n$ ranges over an infinite set of integers.*

*Proof.* Start by applying Lemma 3.1 with

$$k = \left\lceil d^{4/\alpha} \right\rceil$$

to get a $(q', \delta, \epsilon)$-LDC $C' : \mathbb{K}^{n'} \mapsto \mathbb{K}^{m'}$ such that $n' = d^k$, $m' \leq (2tdk)^k \leq \mathrm{poly}(n')$ and with $q' \leq 2tdk \leq \log(n')^{1+\alpha/2}$. The field $\mathbb{K}$ is an extension of $\mathbb{F}$ of degree $\ell \leq \log(q)$. Now, using Lemma 3.2, we replace $\mathbb{K}$ with $\mathbb{F}$ and get a code $C : \mathbb{F}^n \mapsto \mathbb{F}^m$ with $n = \ell n'$ and $m = \ell m'$. Therefore, we still have $m = \mathrm{poly}(n)$. The number of queries grew from $q = \log(n')^{1+\alpha/2}$ to $q\ell \leq q \cdot \log(q)$ which is at most $\log(n)^{1+\alpha}$ for sufficiently large $n$. $\square$

# 4 Proof of Theorem 1

The proof of Theorem 1 will follow by combining Lemma 2.1 with the upper bounds given in Section 3 (Corollaries 3.3 and 3.4).

We start by showing that the strong version of Conjecture 1 implies the solution of the strong version of Problem 1. Suppose the strong version of Conjecture 1 holds. That is, there exists $\alpha, \beta, \gamma, \epsilon > 0$ such that there are no $\left(n^\alpha, n^{-\beta}, \epsilon\right)$-LCC's in $\mathbb{F}^n$ of dimension at least $(1-\gamma)n$. Set

$$\lambda \triangleq \min\left\{\alpha/2,\ \beta/2\right\}.$$

Let $C : \mathbb{F}^n \mapsto \mathbb{F}^m$ be given by Corollary 3.3 such that $C$ is an $(n^{\alpha/2}, \delta, \epsilon)$-LDC with $m = O(n)$, $\delta = \delta(\epsilon)$ and with an explicit generating matrix $A_C \in M_{m\times n}(\mathbb{F})$. We claim that

$$A_C \text{ is } \left((\gamma/2)\cdot n, n^\lambda\right)\text{-rigid.}$$

Suppose not, then we can apply Lemma 2.1 with the parameter $\rho = \gamma/2$ to get that there exists a $(q, \delta', \epsilon)$-LCC $C' \subset \mathbb{F}^n$ with

$$q \leq n^{\alpha/2} \cdot n^\lambda \leq n^\alpha,$$

$$\delta' = \frac{\gamma \cdot \delta}{2 \cdot n^\lambda} \geq \frac{1}{n^\beta}$$

and dimension at least

$$n(1-\gamma/2) - (\gamma/2)n = (1-\gamma)n.$$

contradicting our assumption. Therefore, $A_C$ is an explicit rigid matrix as is required by the strong version of Problem 1.

The proof of the weak case goes along the same lines, replacing Corollary 3.3 with Corollary 3.4. Suppose the weak version of Conjecture 1 holds. That is, there exists $\alpha, \beta, \gamma, \epsilon > 0$ such that there are no $\left(\log^{2+\alpha}(n), 1/\log^{1+\beta}(n), \epsilon\right)$-LCC's in $\mathbb{F}^n$ of dimension at least $(1-\gamma)n$. As before, set $\lambda \triangleq \min\{\alpha/2, \beta/2\}$. Let $C : \mathbb{F}^n \mapsto \mathbb{F}^m$ be given by Corollary 3.4 such that $C$ is a $\left(\log^{1+\alpha/2}(n), \delta, \epsilon\right)$-LDC with $m = \mathrm{poly}(n)$, $\delta = \delta(\epsilon)$ and with an explicit generating matrix $A_C$. We claim that

$$A_C \text{ is } \left((\gamma/2)\cdot n, \log^{1+\lambda}(n)\right)\text{-rigid.}$$

Suppose not, then we can apply Lemma 2.1 with the parameter $\rho = \gamma/2$ to get that there exists a $(q, \delta', \epsilon)$-LCC $C' \subset \mathbb{F}^n$ with

$$q \leq \log^{1+\alpha/2}(n) \cdot \log^{1+\lambda}(n) \leq \log^{2+\alpha}(n),$$

$$\delta' = \frac{\gamma \cdot \delta}{2 \cdot \log^{1+\lambda}(n)} \geq \frac{1}{\log^{1+\beta}(n)}$$

and dimension at least $(1-\gamma)n$, contradicting our assumption. Therefore, $A_C$ is an explicit rigid matrix as is required by the weak version of Problem 1.

# 5 Discussion and directions for future work

The main contribution of this work is a conceptual one - we show a connection between two well studied and seemingly unrelated problems. We hope this connection will lead to a better understanding of both problems. Another contribution is the formulation of Conjecture 1, which deals with a range of parameters not considered before for locally self-correctable codes.

We now turn to discuss the meaning of our results over fields of characteristic zero. We then conclude with a short section concerning the possibility of approaching Conjecture 1 by considering the special case of cyclic codes.

## 5.1 Fields of zero characteristic

Even though the notion of ‘explicitness’ is slightly ill defined over infinite fields, one can still try to come up with ‘interesting’ families of rigid matrices. Say we are working over the complex numbers. We could try to use Lemma 2.1 in conjunction with LDC upper bounds and LCC lower bounds to show that the generating matrices of sufficiently good LDC’s (over the complex numbers) are rigid. The task of proving LCC lower bounds certainly seems easier when the characteristic is zero. This advantage, however, becomes a disadvantage when we try to find LDC upper bounds. To the best of our knowledge, LDC’s that match the parameters of RM codes (as they appear, say, in the two corollaries of Section 3) are not known to exist over fields of characteristic zero. Recently, it was shown in [Gop09, DGY10] that LDC's emerging from earlier works of [Yek08, Efr09] can be made to work also over the complex (or real) numbers. Even though these codes have an encoding length which is super-polynomial in $n$, they could still be used (in conjunction with suitable LCC lower bounds) to get interesting results on the rigidity of their generating matrices. We note that we do not know of **any** non trivial LCC's over the complex numbers. We believe that trying to prove bounds on LCC's over the complex (or the real) field is a good starting point for approaching Conjecture 1.

### 5.2 Lower bounds for cyclic LCC's

A cyclic code $C \subset \mathbb{F}^n$ is a subspace which is invariant under cyclic shifts of the coordinates. That is, if we denote by $\pi : \mathbb{F}^n \mapsto \mathbb{F}^n$ the map which cyclically permutes the coordinates of vectors, we have that for all $x \in C,\ \pi(x) \in C$. Cyclic codes have many applications and many of the codes used in practice (including Reed Muller codes) are cyclic. The question of whether there are good families of cyclic codes is still open. In [BSS05] it was shown that cyclic codes cannot be both good (having constant rate and relative distance) and *locally testable*. The notion of local testability bears some resemblance to local decodability and requires the existence of a local procedure that can distinguish (w.h.p) between code words and vectors which are far (in Hamming distance) from the code. A possible line of attack towards proving Conjecture 1 is to prove it first for cyclic LCC's. Since the LCC constructions we know of are all cyclic this type of result will certainly be of interest.

## 6 Acknowledgements

I am grateful to Boaz Barak, Amir Yehudayoff and Avi Wigderson for many helpful discussions on the topic of matrix rigidity that inspired this work. I thank Sergey Yekhanin for helpful comments.

## References

[APY09] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *APPROX '09 / RANDOM '09: Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 339–351, Berlin, Heidelberg, 2009. Springer-Verlag.

[BIKR02] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-François Raymond. Breaking the o(n1/(2k-1)) barrier for information-theoretic private information retrieval. In *FOCS*, pages 261–270. IEEE Computer Society, 2002.

[BIW07] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private pir. In *APPROX '07/RANDOM '07: Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 311–325, Berlin, Heidelberg, 2007. Springer-Verlag.

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995.

[BSS05] László Babai, Amir Shpilka, and Daniel Stefankovic. Locally testable cyclic codes. *IEEE Transactions on Information Theory*, 51(8):2849–2858, 2005.

[DGY10] Z. Dvir, P. Gopalan, and S. Yekhanin. Matching vector codes. Manuscript, 2010.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In Michael Mitzenmacher, editor, *STOC*, pages 39–44. ACM, 2009.

[GKST06] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

[Gop09] Parikshit Gopalan. A note on efremenko's locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, (069), 2009.

[HB98] W. C. Huffman and Richard A. Brualdi. *Handbook of Coding Theory*. Elsevier Science Inc., New York, NY, USA, 1998.

[KdW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Syst. Sci.*, 69(3):395–420, 2004.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*, pages 80–86, 2000.

[Lip90] Richard J. Lipton. Efficient checking of computations. In Christian Choffrut and Thomas Lengauer, editors, *STACS*, volume 415 of *Lecture Notes in Computer Science*, pages 207–215. Springer, 1990.

[Lok06] Satyanarayana V. Lokam. Quadratic lower bounds on matrix rigidity. In Jin yi Cai, S. Barry Cooper, and Angsheng Li, editors, *TAMC*, volume 3959 of *Lecture Notes in Computer Science*, pages 295–307. Springer, 2006.

[Lok09] Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1-2):1–155, 2009.

[MS77] F.J. MacWilliams and N.J.A. Sloane. *The theory of error correcting codes.* 1977.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, 13:2004, 2004.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In Jozef Gruska, editor, *MFCS*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.

[Val79] L. G. Valiant. Completeness classes in algebra. In *STOC '79: Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 249–261, New York, NY, USA, 1979. ACM.

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. Electronic Colloquium on Computational Complexity (ECCC) TR07-006, 2007.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), 2008.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de