

Testing convexity of functions over finite domains

Aleksandrs Belovs¹, Eric Blais², and Abhinav Bommireddi²

¹Faculty of Computing, University of Latvia, Riga, Latvia, aleksandrs.belovs@lu.lv

²Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada,
eblais,vabommir@uwaterloo.ca

August 8, 2020

Abstract

We establish new upper and lower bounds on the number of queries required to test convexity of functions over various discrete domains.

1. We provide a simplified version of the non-adaptive convexity tester on the line. We improve the upper bound $O\left(\frac{\log(\varepsilon n)}{\varepsilon}\right)$ in the usual uniform model, and prove an $O\left(\frac{\log n}{\varepsilon}\right)$ upper bound in the distribution-free setting.
2. We show a tight lower bound of $\Omega\left(\frac{\log(\varepsilon n)}{\varepsilon}\right)$ queries for testing convexity of functions $f : [n] \rightarrow \mathbb{R}$ on the line. This lower bound applies to both adaptive and non-adaptive algorithms, and matches the upper bound from item 1, showing that adaptivity does not help in this setting.
3. Moving to higher dimensions, we consider the case of a stripe $[3] \times [n]$. We construct an *adaptive* tester for convexity of functions $f : [3] \times [n] \rightarrow \mathbb{R}$ with query complexity $O(\log^2 n)$. We also show that any *non-adaptive* tester must use $\Omega(\sqrt{n})$ queries in this setting. Thus, adaptivity yields an exponential improvement for this problem.
4. For functions $f : [n]^d \rightarrow \mathbb{R}$ over domains of dimension $d \geq 2$, we show a non-adaptive query lower bound $\Omega\left(\left(\frac{n}{d}\right)^{\frac{d}{2}}\right)$.

1 Introduction

Let X be a subset of \mathbb{R}^d . A function $f : X \rightarrow \mathbb{R}$ is called *convex* if for every finite collection of points $x_1, x_2, \dots, x_k \in X$ and non-negative reals $\lambda_1, \dots, \lambda_k \geq 0$ satisfying $\sum_i \lambda_i = 1$ and $\sum_i \lambda_i x_i \in X$, we have

$$f\left(\sum_i \lambda_i x_i\right) \leq \sum_i \lambda_i f(x_i).$$

Convex functions are typically considered on convex domains, but for property testing questions, we will be mostly interested in the case when X is a finite (hence, discrete) subset of \mathbb{R}^d . In this case, one can show that f is convex on X if and only if it can be extended to a convex function $\tilde{f} : \mathbb{R}^d \rightarrow \mathbb{R}$ on the entire linear space \mathbb{R}^d .¹

¹I.e., f is convex on X if and only if there exists a function \tilde{f} that is convex on \mathbb{R}^d and satisfies $\tilde{f}(x) = f(x)$ for every $x \in X$. See Section 2 for details.

For a finite set X , we say that a function $g: X \rightarrow \mathbb{R}$ is ϵ -far from convex with respect to some proximity parameter $0 < \epsilon < 1$ if for every convex function $h: X \rightarrow \mathbb{R}$, we have $|\{x \in X : h(x) \neq g(x)\}| \geq \epsilon|X|$.

In this work, we consider the problem of distinguishing convex functions from those that are far from convex in the property testing framework [11, 19]. Formally, an (ϵ, X) -convexity tester is a bounded-error randomized algorithm that queries the values of an unknown function $f: X \rightarrow \mathbb{R}$ on a set of inputs from X and distinguishes the case where f is convex from the one where f is ϵ -far from convex. A tester is *non-adaptive* if it selects all the inputs to query before observing the value of f on any of those inputs; otherwise the tester is *adaptive*.

Our goal is to determine the minimum query complexity of (ϵ, X) -convexity testers for various discrete sets X , and to determine whether the query complexity of adaptive and non-adaptive (ϵ, X) -convexity testers differs for any set X . While there has been work studying the problem of testing convexity of functions in various settings [17, 2, 16, 3, 4], large gaps remain between the best upper and lower bounds. We give new bounds on the number of queries required to test convexity of functions on the line, over a stripe, and over higher-dimensional domains.

1.1 Testing convexity on the line

The problem of testing convexity of functions $f: [n] \rightarrow \mathbb{R}$ on the line was first considered by Parnas, Ron, and Rubinfeld [17]. They showed that $O(\frac{\log n}{\epsilon})$ queries suffice to ϵ -test convexity in this setting. A slightly better upper bound of $O(\frac{\log(\epsilon n)}{\epsilon})$ was shown by Ben-Eliezer [2]. This follows from his more general algorithm for testing local properties of arrays. We give a more direct algorithm.

Theorem 1.1. *There exists an ϵ -tester for convexity of functions $f: [n] \rightarrow \mathbb{R}$ over the line with complexity $O(\frac{\log(\epsilon n)}{\epsilon})$. The tester is non-adaptive and has 1-sided error.*

We also consider the problem of testing convexity of functions on the line in the distribution-free model of Halevy and Kushilevitz [12]. In this model, the distance of a function $g: X \rightarrow \mathbb{R}$ to convexity is measured with respect to some unknown distribution \mathcal{D} over the domain X . The algorithm can query the target function $f: [n] \rightarrow \mathbb{R}$ as usual, and it can also sample from \mathcal{D} . The tester must distinguish the case where f is convex and the case where f is ϵ -far from a convex function with respect to \mathcal{D} , in that $\Pr_{x \sim \mathcal{D}}[g(x) \neq h(x)] \geq \epsilon$ for every convex function h . The tester must work for any distribution \mathcal{D} , and the complexity measure is the worst-case sum of the number of queries to f and samples from \mathcal{D} . Thus, distribution-free property testing is at least as hard as usual property testing, and for some problems the query complexity is much larger in the distribution-free setting [12].

We show that our algorithm for testing convexity of functions $f: [n] \rightarrow \mathbb{R}$ can be made distribution-free with only a slight loss in the dependence of ϵ .

Theorem 1.2. *There exists a non-adaptive 1-sided algorithm that ϵ -tests a function $f: [n] \rightarrow \mathbb{R}$ for convexity with respect to an unknown distribution \mathcal{D} using $O(\frac{\log n}{\epsilon})$ queries to f and $O(\frac{1}{\epsilon})$ samples from \mathcal{D} .*

The algorithms that establish Theorems 1.1 and 1.2 are both *triple testers*: they repeatedly draw triples of points from a natural probability distribution over $[n]^3$ and test that the function is convex on those three points.² This has a number of consequences. First, both our algorithm admit

²This is similar to the situation for the well-known *pair testers* for monotonicity [8] that sample pairs of points from a natural distribution and test them for monotonicity. Note also that while it is not presented as such, the convexity tester of Parnas, Ron, and Rubinfeld [17] can also be reformulated as a triple tester.

time-efficient implementation. Second consequence is for quantum testers (see [14] for introduction to quantum property testing). Using quantum amplitude amplification [5], we can achieve quadratic improvement. Thus, in the standard property testing model, the quantum query complexity for ϵ -testing convexity is $O\left(\sqrt{\epsilon^{-1} \log(\epsilon n)}\right)$, and in the distribution-free setting the quantum query complexity of the problem is $O\left(\sqrt{\epsilon^{-1} \log n}\right)$. Again, both of the algorithms can be implemented time-efficiently.

Blais, Raskhodnikova, and Yaroslavtsev [4] showed that the bound in Theorem 1.1 on the query complexity of *non-adaptive* convexity testers is optimal when $\epsilon > 0$ is a constant. For adaptive algorithms, this only gives a lower bound of $\Omega(\log \log n)$ by the standard conversion between adaptive and non-adaptive algorithms. We close this gap and show that the bound in Theorem 1.1 is optimal for all values of $\epsilon \leq \frac{1}{9}$, even when the testers are allowed to be adaptive.

Theorem 1.3. *For every $\frac{1}{n} \leq \epsilon \leq \frac{1}{9}$, any ϵ -tester for convexity of functions $f: [n] \rightarrow \mathbb{R}$ has query complexity $\Omega\left(\frac{\log(\epsilon n)}{\epsilon}\right)$.*

In particular, the lower bound in Theorem 1.3 implies that adaptivity does *not* help to reduce query complexity when testing convexity of functions over the line. This is analogous to the situation for testing monotonicity of functions over the line [10]. This result, combined with the distance approximation algorithm of Fattal and Ron [9], also shows that approximating the distance to convexity is essentially no harder than testing convexity.

1.2 Testing convexity over 2-dimensional domains

Parnas, Ron, and Rubinfeld [17] asked whether convexity can be tested efficiently for functions over 2-dimensional domain. The first non-trivial upper bound on the query complexity for testing the convexity of functions mapping $[n]^2$ to \mathbb{R} was obtained by Ben-Eliezer [2], who showed that $O(n)$ queries suffice for non-adaptive testing of convexity—a number of queries that is sublinear (in fact, quadratically smaller) than the size of the domain.

The only previous lower bound for non-adaptive testing of convexity of functions $f: [n]^2 \rightarrow \mathbb{R}$ was again $\Omega(\log n)$ [4], so it remained open whether it is possible to test convexity non-adaptively using a number of queries that is *exponentially* smaller than the size of the domain. We show that it is not, and that the Ben-Eliezer bound is optimal for all non-adaptive algorithms when ϵ is a constant.

Theorem 1.4 (Special case of Theorem 1.6 below). *Any non-adaptive $\Omega(1)$ -tester for convexity of $f: [n]^2 \rightarrow \mathbb{R}$ has query complexity $\Omega(n)$.*

Note that Theorem 1.4 does not eliminate the possibility that convexity of functions on $[n]^2$ can be tested with polylog(n) queries by *adaptive* algorithms. Based on the results for testing convexity in 1D, one may be tempted to guess that adaptivity does not help in this setting either and that the bound in the theorem could be strengthened to apply to adaptive algorithms as well. To test this intuition, we consider an intermediate domain between 1-dimensional and full 2-dimensional case: the *stripe* $[3] \times [n]$. The same intuition from the 1-dimensional case would suggest that adaptivity does not help in testing convexity of functions over the stripe. We show, however, that here adaptivity can be used to obtain an *exponential* improvement on the query complexity of convexity testers.

Theorem 1.5. *There exists a 1-sided-error algorithm that ε -tests a function $f: [3] \times [n] \rightarrow \mathbb{R}$ for convexity in the distribution-free testing model using $O\left(\frac{\log^2 n}{\varepsilon}\right)$ queries to f and $O\left(\frac{1}{\varepsilon}\right)$ samples from \mathcal{D} . By contrast, any non-adaptive $\Omega(1)$ -tester for convexity of $f: [3] \times [n] \rightarrow \mathbb{R}$ (in the standard testing model) has query complexity $\Omega(\sqrt{n})$.*

The exponential gap between the adaptive and non-adaptive query complexity of convexity testing in Theorem 1.5 stands in stark contrast to the situation for the related problem of testing monotonicity: there it is known that adaptivity does not yield *any* reduction in query complexity, as there is a non-adaptive monotonicity tester for functions $f: [n]^d \rightarrow \mathbb{R}$ with query complexity $O(d \log n)$ [6] and every monotonicity tester (adaptive or not) has query complexity $\Omega(d \log n)$ [7].

1.3 Testing convexity over high-dimensional domains

Ben-Eliezer’s upper bound for testing convexity [2] also carries over to high-dimensional settings. When the dimension d is large, however, the bound is quite weak: it shows that $O(dn^{d-1})$ queries suffice to test convexity non-adaptively. This is (barely) sublinear in the domain size n^d when $d = o(n)$.

Blais, Raskhodnikova, and Yaroslavtsev [4] previously showed that non-adaptive algorithms that test *linear convexity* of functions over the hypergrid $[n]^d$ have query complexity $\Omega(d \log n)$. (Linear convexity is a slightly different notion of convexity than the one studied here; see Appendix A for details.) We show that a much stronger lower bound holds for the problem of testing convexity: any non-adaptive algorithm for testing convexity of functions over $[n]^d$ has query complexity that is linear in n and exponential in d .

Theorem 1.6. *For every $d \geq 2$ and any $\epsilon \leq \frac{1}{10}$, any bounded-error non-adaptive ϵ -tester for convexity has query complexity $\Omega\left(\left(\frac{n}{d}\right)^{\frac{d}{2}}\right)$.*

Note that the trivial upper bound for testing convexity (or any other property) of functions over $[n]^d$ is n^d , so Theorem 1.6 shows that non-adaptive convexity testers cannot do significantly better (qualitatively) than the naïve brute-force testing algorithm.

This result also implies a general lower bound of $\Omega(d \log n)$ queries for adaptive convexity testers of convexity for functions over the hypergrid $[n]^d$. This is the first general lower bound for convexity testing which shows that the query complexity must scale as the product of the dimension and the logarithm of the length of hypergrids.

1.4 Discussion and open problems

Our results suggest two main open problems.

Open Problem 1. Is it possible to $\Omega(1)$ -test convexity of functions $f: [n] \times [n] \rightarrow \mathbb{R}$ with $\text{polylog}(n)$ queries?

Parnas, Ron, and Rubinfeld [17] also raised the problem of determining the query complexity for testing convexity in $d \geq 2$, and the upper bound in Theorem 1.5 provides the first suggestion that the query complexity of the problem might be exponentially smaller than—and not just sublinear in—the domain size. As the lower bound in the same theorem shows, however, any algorithm that would provide a positive answer to this question would have to be adaptive.

We can also generalize Open Problem 1 to ask whether convexity testing of $f: [n]^d \rightarrow \mathbb{R}$ can be done with query complexity $\text{polylog}(n)$ for every constant value of d . For high-dimensional settings, it is also natural to ask about the dependence on d .

Open Problem 2. Must every $\Omega(1)$ -tester for convexity of functions $f: [n]^d \rightarrow \mathbb{R}$ have query complexity $2^{\Omega(d)}$?

Theorem 1.6 gives a positive answer to this question for non-adaptive algorithms, but it still allows for the possibility that there is a convexity tester with query complexity that is polynomial in d . It is also possible that the best query complexity of convexity testers is subexponential in d , even if it is not polynomial in d . (C.f., for instance, the submodularity testing problem, where it is known that $2^{\tilde{O}(\sqrt{d})}$ queries suffice to test submodularity of functions $f: \{0, 1\}^d \rightarrow \mathbb{R}$ [20]. It is possible that a similar bound holds for testing convexity as well.)

1.5 Organization

We introduce some basic facts about convexity in Section 2, establish our algorithmic results in Sections 3 and 4, and give the proofs for our hardness results in Sections 5 and 6.

Specifically, the proofs of Theorems 1.1 and 1.2 for testing convexity over one-dimensional domains are presented Section 3. The upper bound in Theorem 1.5 for testing convexity of functions on the stripe is established in Section 4.

The lower bound in Theorem 1.6 for testing convexity over high-dimensional domains is presented in Section 5; the lower bound for the stripe in Theorem 1.5 is found in Section 5.6; and the optimal lower bound for testing convexity on the line in Theorem 1.3 is presented in Section 6.

2 Basic facts about convexity

In this section, we establish some basic facts about convex functions over finite subsets of \mathbb{R}^d . We use the notation $[n] = \{0, 1, \dots, n - 1\}$ and $[a..b] = \{a, a + 1, \dots, b - 1\}$. All the results in this section are standard; we provide the missing proofs in Appendix B for completeness.

The *restriction* of a function $f: X \rightarrow \mathbb{R}$ to a domain $Y \subseteq X$ is the function $f|_Y: Y \rightarrow \mathbb{R}$ defined by $f|_Y(y) = f(y)$ for each $y \in Y$. Our first basic observation is that restriction preserves convexity.

Lemma 2.1. *Let $f: X \rightarrow \mathbb{R}$ be a convex function and $Y \subseteq X$. Then the function $f|_Y: Y \rightarrow \mathbb{R}$ restricted to Y is also convex.*

To define the extension of convex functions, we first need the notion of a centred simplex.

Definition 2.1. A *simplex* in \mathbb{R}^d is a set of affinely independent points. A *centred simplex* in \mathbb{R}^d is a collection of points x_1, \dots, x_k, z such that x_1, \dots, x_k form a simplex, and z can be (uniquely) expressed as

$$z = \sum_i \lambda_i x_i, \tag{1}$$

where all $\lambda_i > 0$ and $\sum_i \lambda_i = 1$. The point z is called the *centre* of the simplex, and we say that the simplex is *centred at z* when this condition is satisfied.

In other words, x_1, \dots, x_k, z is a centred simplex if z is inside the convex hull of x_1, \dots, x_k and no x_i can be removed from the simplex without breaking this property. When X is a finite subset of \mathbb{R}^d and $x_1, \dots, x_k, z \in X$, we say that the centred simplex is *of X* .

Definition 2.2. The centred simplex x_1, \dots, x_k, z of X is *minimal* iff z is the only point of X inside the convex hull of the simplex x_1, \dots, x_k except for its vertices.

Lemma 2.2. *Let $f: X \rightarrow \mathbb{R}$ be a convex functions with X a finite subset of \mathbb{R}^d . Then the function can be extended to a convex function on the whole space \mathbb{R}^d . That is, there exists a convex function $g: \mathbb{R}^d \rightarrow \mathbb{R}$ such that $g(x) = f(x)$ for all $x \in X$. Moreover, for a point z in the convex hull of X the function g can be defined as*

$$g(z) = \min_{x_1, \dots, x_k} \sum_i \lambda_i f(x_i), \quad (2)$$

where x_1, \dots, x_k range over all simplices of X centred at z , and λ_i are as in Equation (1).

Combining the above two lemmata, we see that if $f: X \rightarrow \mathbb{R}$ is a convex function with $X \subseteq \mathbb{R}^d$ finite, and $X \subseteq Y \subseteq \mathbb{R}^d$, then the function f can be extended to a convex function on Y . This is how we will usually use the above lemma.

We say that a function $f: X \rightarrow \mathbb{R}$ is convex on a centred simplex x_1, \dots, x_k, z if its restriction to this set of points is convex. This is equivalent to

$$f(z) \leq \sum_i \lambda_i f(x_i),$$

where λ_i are as in Equation (1). This notion provides a characterization of convexity that we will use to test convex functions.

Theorem 2.3. *A function $f: X \rightarrow \mathbb{R}$ is convex if and only if it is convex on every minimal centred simplex of X .*

Let us apply the general Theorem 2.3 to the setting where f is a function over the line. For the rest of this section, let $X = \{x_1, x_2, x_3, \dots\} \subseteq \mathbb{R}$ where $x_1 < x_2 < x_3 < \dots$. A centred simplex in this case is a *triple* $x < y < z$ and y is the centre of the triple. A function f is convex on the triple if and only if

$$\frac{f(y) - f(x)}{y - x} \leq \frac{f(z) - f(y)}{z - y}. \quad (3)$$

A minimal centred simplex is a *minimal triple* of the form $x_i < x_{i+1} < x_{i+2}$. Thus, we get the following corollary.

Corollary 2.4. *The function $f: X \rightarrow \mathbb{R}$ is convex if and only if it is convex on every triple $x_i < x_{i+1} < x_{i+2}$ of consecutive points.*

A nice feature of convex functions on the line is that we can efficiently find their minimum.

Theorem 2.5. *Assume $f: X \rightarrow \mathbb{R}$ is a convex function. It is possible to find the minimum of f on X in time $O(\log |X|)$.*

Proof. Use bisection. Let $n = |X|$. If $n < 6$, query all the values of f and find the minimum. Otherwise, let $a = \lfloor n/2 \rfloor$ and $b = a + 1$. Query $f(x_a)$ and $f(x_b)$. If $f(x_a) < f(x_b)$, execute the minimum search on the set x_1, \dots, x_a . Otherwise, execute the minimum search on the set x_b, \dots, x_n . By each execution, the size of the set decreases roughly by a factor of 2, hence $O(\log |X|)$ iterations suffice. \square

3 Algorithms for testing convexity over the line

In this section, we prove Theorem 1.1 and Theorem 1.2. Both theorems are established using similar ideas, by constructing explicit convexity testing algorithms that are inspired by the monotonicity tester on the line [1].

Definition 3.1. Let $a \in [n]$. A *triple test* rooted at a is a (non-necessarily sorted) triple (a, b, c) such that

- $b \in \{2^k \lfloor \frac{a-1}{2^k} \rfloor, 2^k \lceil \frac{a+1}{2^k} \rceil\}$ for some integer k satisfying $1 \leq 2^k < n$, and
- c is either $a + 1$ or $b + 1$.

The element a is called the *root*, and b is called a *hub* of a . The integer 2^k is called the *height* of the triple. We say that a passes the triple test if the function f is convex on $\{a, b, c\}$. We say that a passes all its triple tests if it passes all the triple tests rooted at it.

Claim 3.1. *If $x < y - 1$, then x and y have a common hub with height not exceeding $2(y - x)$.*

Proof. Let k be such that $\frac{y-x}{2} \leq 2^k \leq y-x$. There can be either one or two multiples of 2^k between x and y . If there is just one then we are done and that is the common hub. If there are two then there will be exactly one multiple of 2^{k+1} between x and y and that is their common hub. \square

Lemma 3.2. *Assume $x < y < z$ is a non-convex triple. Then at least one of x, y or z fails some of its triple tests with height not exceeding $2 \cdot \max\{y - x, z - y\}$.*

Proof. Assume for now that $y - x \geq 2$ and $z - y \geq 2$. Let h be the common hub between x, y with height not exceeding $2(y - x)$ and h' be the common hub between y, z with height not exceeding $2(z - y)$. Consider the function f restricted to the domain $x < h < y < h' < z$. By Lemma 2.1, we know the function is not convex, hence, by Corollary 2.4, it is non-convex on at least one of the triples (x, h, y) , (h, y, h') , or (y, h', z) . Let us consider the three cases separately:

- f is non-convex on the triple x, h, y . Consider the function f on the domain $x < h < h+1 \leq y$. Using Corollary 2.4 if needed, we get that the function f is non-convex on one of the triples $(x, h, h+1)$ or $(h, h+1, y)$. Each of them constitutes a triple test: $a = x, b = h, c = h+1$, or $a = y, b = h, c = h+1$, respectively.
- f is non-convex on the triple h, y, h' . Consider the function f on the domain $h < y < y+1 \leq h'$. The function f is non-convex on one of the triples $(h, y, y+1)$ or $(y, y+1, h')$. Again, each of them constitutes a triple test: $a = y, b = h, c = y+1$, or $a = y, b = h', c = y+1$, respectively.
- f is non-convex on the triple y, h', z . This case is analogous to the first one.

If $y = x + 1$, then the above analysis works with $h = x$ (the first case never holds, and $x = y - 1$ is a hub of y). If $z = y + 1$, the above analysis works with $h' = z$ (the third case never holds, and $z = y + 1$ is a hub of y). Finally, if both $y = x + 1$ and $z = y + 1$, we can use the triple test with $a = x, b = y$, and $c = z$. \square

A simple consequence of this lemma is that the function f is convex on the set of points passing all their triple tests. This allows us to formulate the following notion.

Definition 3.2. A *convex replacement* of a function $f: [n] \rightarrow \mathbb{R}$ is a convex function $\tilde{f}: [n] \rightarrow \mathbb{R}$ such that $f(x) = \tilde{f}(x)$ for all x that pass all their triple tests.

The proof of Theorem 1.2 now follows easily.

Proof of Theorem 1.2. The algorithm is simple: sample a from \mathcal{D} and run all the triple tests rooted at a . It takes 1 sample and $O(\log n)$ queries. The probability that this test fails is at least the distance (with respect to \mathcal{D}) to the convex replacement to f . Repeat the above test $O(1/\varepsilon)$ times to increase the success probability to $\Omega(1)$. \square

We are now also ready to complete the proof of Theorem 1.1.

Proof of Theorem 1.1. The algorithm is a triple tester. It selects a root a of the triple uniformly at random from $[n]$, select a triple rooted at a with height at most $2\varepsilon n$ uniformly at random, and tests it for convexity. For completeness, let us restate the algorithm:

Algorithm 1: CONVEXITYTEST1D

- 1 Draw a uniformly at random from $[n]$;
 - 2 Draw k uniformly at random from $\{0, \dots, \lceil \log_2(2\varepsilon n) \rceil\}$;
 - 3 Let b be either the largest multiple of 2^k strictly smaller than x , or the smallest multiple of 2^k strictly larger than x , each case with probability $1/2$;
 - 4 Let c be either $a + 1$ or $b + 1$, each with probability $1/2$;
 - 5 Query $f(a)$, $f(b)$ and $f(c)$ and test whether a, b, c form a convex triple;
-

We claim that if the function f is ε -far from convex, then this test fails with probability $\Omega(\varepsilon/\log(\varepsilon n))$. Thus, this test has to be repeated $O(\log(\varepsilon n)/\varepsilon)$ times.

We will construct a subset $A \subseteq [n]$ of size εn such that every $a \in A$ fails one of its triple tests with height at most $2\varepsilon n$. Start with $S \leftarrow [n]$. We treat S as a sorted list. While $|[n] \setminus S| < \varepsilon n$, the function $f|_S$ is non-convex. Choose three neighbouring elements $x < y < z$ in S that violate convexity. Let (a, b, c) be the non-convex triple constructed in Lemma 3.2. The height of this triple is at most $2\varepsilon n$. Remove a from S . When $|[n] \setminus S| \geq \varepsilon n$, let $A \leftarrow [n] \setminus S$. \square

4 Algorithm for testing convexity on the $[3] \times [n]$ stripe

In this section, we prove the upper bound in Theorem 1.5.

4.1 High-level description

Our approach to testing convexity on the stripe $[3] \times [n]$ is as follows. This set is very close to the 1-dimensional line, so we can draw a lot from the tester of Section 3. In this vein, for $i \in [3]$, let $f_i: [n] \rightarrow \mathbb{R}$ be the restrictions of f to the column $\{i\} \times [n]$. We will construct a convex replacement \tilde{f} of f so that every point where f and \tilde{f} disagree fails some test. Sampling $a \in [3] \times [n]$ from \mathcal{D} and executing the test on a will give us a distribution-free tester of convexity.

Any simplex centred at a point in the line $\{0\} \times [n]$ or $\{2\} \times [n]$ is completely contained inside this line. Hence, for f_0 and f_2 we can simply take convex replacements \tilde{f}_0 and \tilde{f}_2 from Definition 3.2, and assume that \tilde{f} restricted to $\{0\} \times [n]$ or $\{2\} \times [n]$ is \tilde{f}_0 or \tilde{f}_2 , respectively.

Let us define a function $h: \{0, 1/2, 1, 3/2, \dots, n-1\} \rightarrow \mathbb{R}$ as

$$h(x) = \min_{\delta} \frac{\tilde{f}_0(x - \delta) + \tilde{f}_2(x + \delta)}{2}. \quad (4)$$

Note that $h(x) = g(1, x)$ where g is the convex extension, as in Equation (2), of the function \tilde{f} restricted to $\{0, 2\} \times [n]$. (We have not defined \tilde{f} on the line $\{1\} \times [n]$ yet.) By Lemma 2.2 and

Lemma 2.1, the function h is convex. Its value can be computed by minimising the convex function $\delta \mapsto (\tilde{f}_0(x - \delta) + \tilde{f}_2(x + \delta))/2$. This is exactly the place where our tester uses adaptivity.

The main part of our algorithm deals with interplay between the functions h and f_1 . Let us give some relations between h and f_1 for the case when f is convex. First, the function f is convex on any simplex of the form $(0, x - \delta), (2, x + \delta)$ centred at $(1, x)$, which implies that $f_1(x) \leq h(x)$ for every $x \in [n]$. Next, for every $\{x, x + 1\} \subseteq [n]$, let $\beta: \mathbb{R} \rightarrow \mathbb{R}$ be the affine function agreeing with f_1 at x and $x + 1$. We have that the function f is convex on any simplex of the form $(0, z - \delta), (1, x), (2, z + \delta)$ centred at $(1, x + 1)$, which implies that $\beta(z) \leq h(z)$ for all $z > x + 1$.³ Similarly, considering simplices $(0, z - \delta), (1, x + 1), (2, z + \delta)$ centred at $(1, x)$, we get that $\beta(z) \leq h(z)$ for all $z < x$. Our tester will check these conditions.

4.2 Subroutines

We are now ready to describe the subroutines used by our tester. The first subroutine is the convexity test for the line from Section 3.

Algorithm 2: 1DTEST(i, x)

- 1 Execute all the triple tests rooted on x for the function f_i as in Definition 3.1;
 - 2 If at least one of the tests fails, output that f is not convex and terminate the algorithm;
-

The complexity of this subroutine is $O(\log n)$. The following claim is a direct consequence of Definition 3.2.

Claim 4.1. *If 1DTEST(i, x) does not fail for $i = 0$ or $i = 2$, then $\tilde{f}_i(x) = f_i(x)$.*

The next subroutine evaluates the function h .

Algorithm 3: EVALUATE(x)

- 1 Find a point δ^* where the function $g(\delta) = \frac{1}{2}(f_0(x - \delta) + f_2(x + \delta))$ attains its minimum, assuming this function is convex;
 - 2 For $y \in \{x - \delta^* - 1, x - \delta^*, x - \delta^* + 1\}$ perform 1DTEST($0, y$);
 - 3 For $y \in \{x + \delta^* - 1, x + \delta^*, x + \delta^* + 1\}$ perform 1DTEST($2, y$);
 - 4 Check that $g(\delta^*) \leq g(\delta^* - 1)$ and $g(\delta^*) \leq g(\delta^* + 1)$;
 - 5 Return $g(\delta^*)$;
-

Claim 4.2. *The subroutine either finds a violation of convexity or returns $h(x)$. The complexity of the subroutine is $O(\log n)$.*

Proof. Define $\tilde{g}(\delta) = \frac{1}{2}(\tilde{f}_0(x - \delta) + \tilde{f}_2(x + \delta))$ so that $h(x) = \min_{\delta} \tilde{g}(\delta)$. Steps 2 and 3 of the subroutine ensure that g and \tilde{g} agree on $\delta^* - 1, \delta^*$ and $\delta^* + 1$. If Step 4 fails, we get that $g \neq \tilde{g}$, meaning that the function f is not convex. Otherwise, we get that $\tilde{g}(\delta^*) \leq \tilde{g}(\delta^* - 1)$ and $\tilde{g}(\delta^*) \leq \tilde{g}(\delta^* + 1)$. As \tilde{g} is convex, this implies that the minimum of \tilde{g} is attained at δ^* . The complexity estimate is obvious. \square

³Note that this observation does not immediately follow from the first observation and convexity of f_1 , because it also incorporates half-integer values of z , where f_1 is not defined.

4.3 The algorithm

Now let us state the test for convexity over the stripe.

Algorithm 4: CONVEXITYTESTSTRIPE

- 1 Sample (i, x) from \mathcal{D} ;
 - 2 Perform $\text{1DTEST}(i, x)$;
 - 3 **if** $i = 1$ **then**
 - 4 EVALUATE $h(x)$ and verify that $f_1(x) \leq h(x)$;
 - 5 Perform $\text{1DTEST}(1, x - 1)$ and $\text{1DTEST}(1, x + 1)$;
 - 6 Let $\beta^- : [n] \rightarrow \mathbb{R}$ be the affine function satisfying $\beta^-(x - 1) = f_1(x - 1)$ and $\beta^-(x) = f_1(x)$;
 - 7 Minimise the convex function $h - \beta^-$ on the interval $\{x + 1, x + \frac{3}{2}, x + 2, \dots, n - 1\}$ and verify that the minimum is non-negative;
 - 8 Let $\beta^+ : [n] \rightarrow \mathbb{R}$ be the affine function satisfying $\beta^+(x) = f_1(x)$ and $\beta^+(x + 1) = f_1(x + 1)$;
 - 9 Minimise the convex function $h - \beta^+$ on the interval $\{0, \frac{1}{2}, 1, \dots, x - 1\}$ and verify that the minimum is non-negative;
 - 10 **end**
-

The tester uses one sample from \mathcal{D} and $O(\log^2 n)$ queries to f , since steps 7 and 9 each require $O(\log n)$ calls to the EVALUATE subroutine, which in turn makes $O(\log n)$ queries to f .

By the discussion at the beginning of the section, any convex function f passes the test with probability 1. Let $f : [3] \times [n] \rightarrow \mathbb{R}$ be any function that is ε -far from convex with respect to \mathcal{D} . Let S be the set of points that pass the test. We claim that f restricted to S is convex. Hence, the error probability of the test is at least ε , and it suffices to repeat the test $O(1/\varepsilon)$ times.

In order to prove that f is convex on S , we extend it to a slightly larger domain. This is done to better handle possible minimal centred simplices. Let as above \tilde{f}_i be convex replacement of f_i . We claim that the function $\tilde{f} : (\{0, 2\} \times [n]) \cup S \rightarrow \mathbb{R}$ defined by

$$\tilde{f}(i, x) = \begin{cases} \tilde{f}_i(x), & \text{if } i = 0 \text{ or } i = 2; \\ f(i, x), & \text{if } (i, x) \in S; \end{cases}$$

is convex (the two values are equal when both conditions apply). As f and \tilde{f} agree on S , this implies that f restricted to S is convex.

By Theorem 2.3 and above discussion, it suffices to consider minimal simplices centred at points of the form $(1, x) \in S$. From Lemma 3.2, we get that the function \tilde{f} is convex on a centred simplex of the form $\{(1, x), (1, y), (1, z)\} \subseteq S$. The function \tilde{f} is also convex on a simplex $(0, x - \delta), (2, x + \delta)$ centred at $(1, x)$ because $h(x) \geq f_1(x)$ by Step 4.

Any other minimal simplex centred at $(1, x)$ is of the form $(0, a), (1, b), (2, c)$. Let $y = (a + c)/2$, and assume $b < x < y$ (the case $y < x < b$ is similar). From Step 7 of the algorithm, we know that the function $g : \{x - 1, x, y\} \rightarrow \mathbb{R}$ defined by

$$\begin{aligned} g(x - 1) &= f_1(x - 1), \\ g(x) &= f_1(x), \\ g(y) &= h(y) \end{aligned}$$

is convex. Both $(1, b)$ and $(1, x)$ are in S and so they pass the test. Thus, from Steps 2 and 5, we have that f_1 and \tilde{f}_1 agree on $b, x - 1$ and x . As \tilde{f}_1 is convex, and using Corollary 2.4, we have that

the function $g' : \{b, x-1, x, y\} \rightarrow \mathbb{R}$ defined by

$$\begin{aligned} g'(b) &= f_1(b), \\ g'(x-1) &= f_1(x-1), \\ g'(x) &= f_1(x), \\ g'(y) &= h(y) \end{aligned}$$

is also convex. Finally, since $h(y) \leq (\tilde{f}_0(a) + \tilde{f}_2(c))/2$, we get that \tilde{f} is convex on the simplex $(0, a), (1, b), (2, c)$ centred at $(1, x)$.

5 Lower bounds for testing convexity in high dimensions

In this section we prove the $\Omega\left(\left(\frac{n}{d}\right)^{\frac{d}{2}}\right)$ lower bound for non-adaptive algorithms that test convexity on the $[n]^d$ grid in Theorem 1.6 and the $\Omega(\sqrt{n})$ lower bound for non-adaptive algorithms that test convexity over the stripe $[3] \times [n]$ in Theorem 1.5.

5.1 Overview of the proof

The lower bounds in Theorems 1.5 and 1.6 are both obtained using the same general construction. We describe it in the setting of functions over $[n]^d$ for simplicity.

The key idea is that we can construct convex functions whose increase in slope (i.e., second derivative) is small in a particular direction and large in the rest of the directions. We can perturb the values of such functions by ± 1 in a way that yields functions which are far from convex but for which the only violation of convexity on the hypergrid will contain at least two points that form a line along the direction where the slope was increasing slowly. So any algorithm that does not query two points which give a line in that direction cannot catch any violations of convexity. To get a strong lower bound from this key idea, we show that it is possible to “hide” the slowly-increasing direction among $\Omega\left(\left(\frac{n}{d}\right)^d\right)$ possible directions. Since a set of q queries contains pairs of points that form at most q^2 different directions, this construction shows that any non-adaptive convexity testing algorithm with one-sided error—i.e., that always accepts convex functions—must have query complexity at least $\Omega\left(\left(\frac{n}{d}\right)^{d/2}\right)$.

To generalize this argument in a way that gives a lower bound for non-adaptive testing algorithms with two-sided error as well, we consider a different perturbation of the convex functions of ± 1 that preserves convexity. We can do this by performing the same perturbation (i.e., either all $+1$ or all -1) for every point along a line in the slowly-increasing direction. The perturbations for each line are chosen independently at random; by ensuring that the slope of the original function is large enough in all other directions, these independent perturbations do not violate convexity. As we show in the rest of this section, non-adaptive algorithms with query complexity $o\left(\left(\frac{n}{d}\right)^{d/2}\right)$ cannot distinguish this type of perturbation from the type that breaks convexity.

5.2 Preliminaries

We write $x_{[a,b]}$ to denote the coordinates x_a, x_{a+1}, \dots, x_b of an input x . We use the following standard results in our proof.

Lemma 5.1 (Hoeffding's inequality). *Let $x_1, \dots, x_n \in \mathbb{R}$ be negatively correlated random variables bounded by $x_i \in [b_i, a_i]$ and define $\bar{x} = \frac{1}{n}(x_1 + \dots + x_n)$. Then*

$$\Pr[|\bar{x} - \mathbb{E}[\bar{x}]| \geq t] \leq 2e^{-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}}.$$

Lemma 5.2 (Yao's minimax). *Fix any disjoint sets \mathcal{P} and \mathcal{N} of functions mapping \mathcal{X} to \mathcal{Y} . Let $\mathcal{D}_{\mathcal{P}}$ and $\mathcal{D}_{\mathcal{N}}$ be probability distributions on functions mapping \mathcal{X} to \mathcal{Y} that satisfy*

$$\Pr_{f \sim \mathcal{D}_{\mathcal{P}}}[f \in \mathcal{P}] = 1 \quad \text{and} \quad \Pr_{g \sim \mathcal{D}_{\mathcal{N}}}[f \in \mathcal{N}] = 1 - o(1).$$

Let \mathcal{D} be the distribution where with probability $\frac{1}{2}$ we sample from $\mathcal{D}_{\mathcal{P}}$ and with probability $\frac{1}{2}$ we sample from $\mathcal{D}_{\mathcal{N}}$. If any non-adaptive deterministic algorithm Π with query complexity q can not answer correctly with probability $\frac{2}{3}$, then any non-adaptive randomized algorithm that decides whether $f \in \mathcal{P}$ or $f \in \mathcal{N}$ with error at most $\frac{1}{4}$ makes $\Omega(q)$ queries.

Proposition 5.3 (Theorem 332 [13]). *Let $a, b \in [n]$ be two numbers picked uniformly at random. The probability that the pair (a, b) is co-prime is > 0.5 .*

5.3 Change of basis and convexity

Definition 5.1. A *lattice basis* is a matrix $B = [b_1, \dots, b_k] \in \mathbb{R}^{d \times k}$ whose columns are linearly independent vectors in \mathbb{R}^d . The *lattice generated by B* is the set

$$\mathcal{L}(B) = \left\{ Bx \mid x \in \mathbb{Z}^k \right\} = \left\{ \sum_{i=1}^k x_i b_i \mid x_1, \dots, x_k \in \mathbb{Z} \right\}.$$

Fact 5.4 (Lemma 1.2 [18]). *$B \in \mathbb{Z}^{[d] \times [d]}$ is a basis of \mathbb{Z}^d if and only if its determinant is ± 1 .*

Definition 5.2. Given any vector $a \in \mathbb{Z}^d$ whose first two coordinates a_1 and a_2 are coprime, the *canonical basis completion* of a is the basis $B(a) = [b_1(a), \dots, b_d(a)] \in \mathbb{Z}^{d \times d}$ whose i th column is

$$b_i(a) = \begin{cases} a & \text{if } i = 1 \\ c_1 e_1 + c_2 e_2 & \text{if } i = 2 \\ e_i & \text{if } 3 \leq i \leq d \end{cases}$$

where c_1 and c_2 are the integers that satisfy $a_1 c_1 - a_2 c_2 = 1$ and $e_i \in \mathbb{Z}^d$ is the vector with value 1 in the i th coordinate and 0 in all other coordinates.

The next proposition shows that the canonical basis completion of any vector $a \in \mathbb{Z}^d$ that satisfies the condition of the above definition generates the lattice \mathbb{Z}^d .

Proposition 5.5. *Given any vector $a \in \mathbb{Z}^d$ whose first two coordinates a_1 and a_2 are coprime, the canonical basis completion $B(a)$ of a generates the lattice $\mathcal{L}(B(a)) = \mathbb{Z}^d$.*

Proof. Follows from Fact 5.4. □

If $x \in \mathbb{Z}^d$ be the representation of a point according to the basis I , then $x^B = B^{-1}x$ is the representation according to the basis B . So $y = x + a$ and $y^B = x^B + e_1$ are equivalent.

5.4 Constructions

In this subsection we show how to construct the distributions $\mathcal{D}_Y, \mathcal{D}_N$. We also prove that every function in \mathcal{D}_Y is convex and every function in \mathcal{D}_N is $\frac{1}{20}$ -far from convex.

Let \mathcal{B} be the distribution over bases obtained by drawing a vector $a \in \mathbb{Z}^d$ uniformly at random among all vectors whose coordinates are in the range $0 \leq a_1, a_2, \dots, a_d \leq \frac{n}{4d}$ and whose first two coordinates a_1 and a_2 are coprime and returning the canonical basis $B(a)$ for a .

The distributions \mathcal{D}_Y and \mathcal{D}_N are both obtained by drawing a basis from \mathcal{B} and starting with a convex function g_B associated with that basis that we will call the *canonical* convex function for B .

Definition 5.3. The *canonical convex function* for a basis B of \mathbb{Z}^d is the function $g_B: \mathbb{Z}^d \rightarrow \mathbb{Z}$ defined by

$$g_B(x) = (x_1^B)^2 + 2 \sum_{i=2}^d (x_i^B)^2.$$

Our distribution on convex functions is obtained by shifting the values of the canonical convex function g_B in a way that preserves convexity.

Definition 5.4 (\mathcal{D}_Y). Let \mathcal{S}^B to be the distribution on functions $h: [n]^d \rightarrow \mathbb{Z}$ obtained by drawing values $\sigma(z) \in \{\pm 1\}$ independently and uniformly at random for each $z \in \mathbb{Z}^{d-1}$ and defining

$$h(x) = g_B(x) + \sigma(x_2^B, \dots, x_d^B)$$

for each $x \in [n]^d$. Let \mathcal{D}_Y be the distribution obtained by drawing $B \sim \mathcal{B}$ and then drawing a function $h \sim \mathcal{S}^B$.

Our distribution on functions that are far from convex is similar, except that the shifts of the canonical convex function g_B are now constructed in a way that will create many disjoint violations of convexity.

Definition 5.5 (\mathcal{D}_N). Let \mathcal{A}^B be the distribution on functions $h: [n]^d \rightarrow \mathbb{Z}$ obtained by drawing values $\sigma(z) \in \{\pm 1\}$ independently and uniformly at random for each $z \in \mathbb{Z}^{d-1}$ and defining

$$h(x) = g_B(x) + \sigma(x_2^B, \dots, x_d^B) \cdot (-1)^{x_1^B}$$

for each $x \in [n]^d$. Let \mathcal{D}_N be the distribution obtained by drawing $B \sim \mathcal{B}$ and then drawing a function $h \sim \mathcal{A}^B$.

We complete this section by showing that the functions in the support of \mathcal{D}_Y are indeed convex and that the functions in the support of \mathcal{D}_N are far from convex.

Claim 5.6. *Every function in the support of \mathcal{D}_Y is convex.*

Proof. Fix any B in the support of \mathcal{B} , any h in the support of \mathcal{S}^B , and any points $z, x_1, \dots, x_k \in \mathbb{Z}^d$ such that $z = \sum_{i=1}^k \lambda_i x_i$ is a convex combination of the points x_1, \dots, x_k , $\lambda_1, \dots, \lambda_k \geq 0$ and $\sum_{i=1}^k \lambda_i = 1$. We will show that $\sum_{i=1}^k \lambda_i h(x_i) \geq h(z)$.

Let us define $\delta_1, \dots, \delta_k \in \mathbb{Z}^d$ to be the vectors for which $x_i^B = z^B + \delta_i$ for each $i \in [k]$. Then the identity $\sum_{i=1}^k \lambda_i (x_i^B - z^B) = 0$ implies that $\sum_{i=1}^k \lambda_i \delta_{ij} = 0$ for every $j \in [d]$ and that

$$\begin{aligned} \sum_{i=1}^k \lambda_i g_B(x_i) &= \sum_{i=1}^k \lambda_i \left((x_{i1}^B)^2 + 2 \sum_{j=2}^d (x_{ij}^B)^2 \right) \\ &= \sum_{i=1}^k \lambda_i \left((z_1^B + \delta_{i1})^2 + 2 \sum_{j=2}^d (z_j^B + \delta_{ij})^2 \right) \\ &= g_B(z) + \sum_{i=1}^k \lambda_i \left(\delta_{i1}^2 + 2 \sum_{j=2}^d \delta_{ij}^2 \right). \end{aligned}$$

Define $I = \{i \in [k] \mid z_{[2,d]}^B \neq x_{i[2,d]}^B\}$. For each $i \in I$, the vector δ_i satisfies $\sum_{j=2}^d \delta_{ij}^2 \geq 1$ so we have that

$$\sum_{i=1}^k \lambda_i g_B(x_i) - g_B(z) \geq 2 \sum_{i=1}^k \lambda_i \sum_{j=2}^d \delta_{ij}^2 \geq 2 \sum_{i \in I} \lambda_i.$$

Furthermore, since $\sigma(x_{i[2,d]}^B) - \sigma(z_{[2,d]}^B)$ is always bounded below by -2 and the difference is zero whenever $i \notin I$, we obtain

$$\sum_{i=1}^k \lambda_i h(x_i) - h(z) \geq \sum_{i=1}^k \lambda_i g_B(x_i) - g_B(z) - 2 \sum_{i \in I} \lambda_i \geq 0. \quad \square$$

Claim 5.7. *Every function in the support of \mathcal{D}_N is $\frac{1}{20}$ -far from convex.*

Proof. Fix any B in the support of \mathcal{B} and any h in the support of \mathcal{A}^B . For any points $x, y, z \in [n]^d$ that satisfy $y^B = x^B + e_1$ and $z^B = y^B + e_1$, if we have

$$h(x) = g_B(x) - 1, \quad h(y) = g_B(y) + 1, \quad \text{and} \quad h(z) = g_B(z) - 1$$

Then the triple (x, y, z) is a witness of non-convexity of h since

$$\frac{1}{2}h(x) + \frac{1}{2}h(z) = \frac{1}{2}g_B(x) + \frac{1}{2}g_B(z) - 1 = g_B(y) < h(y) = h\left(\frac{1}{2}x + \frac{1}{2}z\right).$$

Hence from how we defined h , any four points $w, x, y, z \in [n]^d$ that satisfy $x^B = w^B + e_1$, $y^B = x^B + e_1$ and $z^B = y^B + e_1$ one of (w, x, y) , (x, y, z) is a witness on non-convexity. Let $L = [\frac{n}{2d}, n - \frac{n}{2d}]^d$. For $s \in \mathbb{Z}^{d-1}$, let $L_s = \{x \mid x \in [n]^d, \exists y \in L \text{ s.t. } y_{[2..d]}^B = x_{[2..d]}^B = s\}$. Since $a_1, a_2, \dots, a_d < \frac{n}{4d}$ we have that $|L_s| \geq 4$. And since any 4 consecutive points with the same $[2, d]$ coordinates, in basis B , have a witness of non-convexity, the number of witnesses in L_s is $\geq \frac{|L_s|}{7}$. Also $L \subseteq \cup_{s \in \mathbb{Z}^{d-1}} L_s$, hence the number of disjoint witnesses of non-convexity is greater than $\frac{|L|}{7} = \frac{1}{7}(1 - \frac{1}{d})^d n^d \geq \frac{1}{20}n^d$. In every disjoint non-convexity witness we have to change the value of at least one point to make the function convex. Therefore h is $\frac{1}{20}$ -far from convex. \square

5.5 Proof of Theorem 1.6

Let \mathcal{D} be the distribution where with probability $\frac{1}{2}$ we pick something from \mathcal{D}_Y and with probability $\frac{1}{2}$ we pick something from \mathcal{D}_N . In this section we prove that there does not exist a non-adaptive deterministic algorithm with query complexity $q < 0.01(\frac{n}{4d})^{\frac{d}{2}}$ that answers correctly with probability $\frac{2}{3}$ on the distribution \mathcal{D} . From Lemma 5.2 this would prove Theorem 1.6 as from Claim 5.6

and Claim 5.7 we know that every function in the support of \mathcal{D}_Y is convex and every function in the support of \mathcal{D}_N is $\frac{1}{10}$ -far from convex.

Let us assume there exists such a deterministic algorithm Π that answers correctly on a distribution $\mathcal{D} = \frac{1}{2}\mathcal{D}_Y + \frac{1}{2}\mathcal{D}_N$ with probability greater than $\frac{2}{3}$. We can think of the distribution \mathcal{D} as pick a $B \sim \mathcal{B}$ and pick a $\sigma: \mathbb{Z}^{d-1} \rightarrow \pm 1$ uniformly at random. And at the end with probability $\frac{1}{2}$ we choose whether we want a function in the support of \mathcal{D}_Y or \mathcal{D}_N . Let the points the algorithm Π queries be $Q = x_1, x_2, \dots, x_q \in \mathbb{Z}^d$.

We refer to a B in the support of \mathcal{B} to be *exposed* if there exists $i, j < q$ such that $x_{i[2,d]}^B = x_{j[2,d]}^B$, otherwise we refer to it as *hidden*.

Claim 5.8. *On the distribution \mathcal{D} the probability that Π answers correctly is less than 0.6.*

Proof. When B is hidden then there is no way the algorithm Π can answer correctly with probability greater than $\frac{1}{2}$. This is because $\Pr_{f \sim \mathcal{D}_Y | B \text{ is hidden}} [f|_Q = \alpha] = \Pr_{g \sim \mathcal{D}_N | B \text{ is hidden}} [g|_Q = \alpha]$. In fact it is even stronger, along with function values at the queried points even if we give what the hidden basis B is, the algorithm can not answer correctly with probability greater than $\frac{1}{2}$. This is because, as for any $i, j < q$, $x_{i[2,d]}^B \neq x_{j[2,d]}^B$, we have $f|_Q - g_B|_Q = s$, for each $s \in \{-1, +1\}^q$, with probability $\frac{1}{2^q}$ irrespective of f being in \mathcal{S}^B or \mathcal{A}^B . We can assume that the algorithm always answers correctly when B is exposed. The probability that the algorithm Π answers correctly is $\leq \Pr[B \text{ is exposed}] \cdot 1 + \Pr[B \text{ is hidden}] \cdot \frac{1}{2}$.

Since there are only $\binom{q}{2} < q^2$, $i, j < q$ pairs, there are at most q^2 exposed B . From Proposition 5.3 and the construction of \mathcal{B} we know that $|\mathcal{B}| \geq 0.5(\frac{n}{4d})^d$ and if $q < 0.01(\frac{n}{4d})^{\frac{d}{2}}$ the probability that a $B \sim \mathcal{B}$ is exposed is $\leq \frac{1}{100}$.

Hence the success probability of the algorithm is $\leq \frac{1}{100} \cdot 1 + \frac{99}{100} \cdot \frac{1}{2} \leq \frac{101}{200}$. \square

This a contradiction on the assumption that the algorithm answers correctly with probability $\frac{2}{3}$. Hence there can not exist such a non-adaptive deterministic algorithm Π .

5.6 Non-adaptive lower bound for $[3] \times [n]$

In this section we prove a $\Omega(\sqrt{n})$ lower bound for non-adaptively testing convexity on the $[3] \times [n]$ grid. The proof is almost the same as the the higher dimensional setting with slight changes.

Let \mathcal{B} be the distribution over bases obtained by drawing a vector $a \in \mathbb{Z}^2$ uniformly at random among all vectors whose first coordinate is 1 and the second coordinate is in the range $0 \leq a_2 \leq \frac{n}{100}$ and returning the canonical basis $B(a)$ for a .

Define the distributions \mathcal{D}_Y and \mathcal{D}_N as above with the one modification that the domain of h is set to be $[3] \times [n]$ instead of $[n]^d$. In this setting, we again have that every function in the support of \mathcal{D}_Y is convex, using the same argument as in Claim 5.6. But now it is no longer true that every function in the support of \mathcal{D}_N is $\frac{1}{10}$ -far from convex. Instead, we have that a function $f \sim \mathcal{D}_N$ is $\frac{1}{10}$ -far from convex with probability $1 - o(1)$.

Claim 5.9. *A function $f \sim \mathcal{D}_N$ is $\frac{1}{10}$ -far from convex with probability $1 - o(1)$.*

Proof. For any B in the support of \mathcal{B} , a function $h \sim \mathcal{A}^B$ is $\frac{1}{10}$ -far from convex with probability $1 - o(1)$. Let $X = \{x \mid x_1 = 0, 0 \leq x_2 \leq \frac{9n}{10}\}$. For any points $x \in X$ and $y, z \in \mathbb{Z}^2$ that satisfy $y^B = x^B + e_1$ and $z^B = y^B + e_1$ we have that $y, z \in [3] \times [n]$ and

$$h(x) = g_B(x) - 1, \quad h(y) = g_B(y) + 1, \quad \text{and} \quad h(z) = g_B(z) - 1$$

with probability $\frac{1}{2}$. Therefore, x, y, z form a witness for non-convexity with probability $\frac{1}{2}$. This is true for all $x \in X$. Using Hoeffding's inequality the probability that the number of witnesses for non-convexity is less than $\frac{n}{3}$ is $\leq e^{-cn}$. Hence with probability $1 - e^{-cn}$ the distance to convexity is at least $\frac{\frac{n}{3}}{3n} \geq \frac{1}{10}$. \square

Any non-adaptive deterministic algorithm which performs $q < \frac{\sqrt{n}}{100}$ can not answer correctly with probability greater than 0.6. The proof is similar to that of Claim 5.8. From Lemma 5.2 this completes the proof of the lower bound in Theorem 1.5.

6 Lower bound for testing convexity on the line

The lower bound in Theorem 1.3 is obtained by using similar ideas to the ones in [1] used to prove the analogous lower bound for testing monotonicity. The key idea is to introduce violations of convexity that are only visible at a given scale.

We first show a lower bound of $\Omega(\log n)$ for $\frac{1}{5}$ -testing convexity and then extend it to general ϵ .

6.1 General principle

In this section, we formulate the general principle our proof is based on in an abstract form to give the overall structure of our proof. In the next sections, we show how to apply it to convexity testing.

We deal with randomised query algorithms whose inputs are functions $f: [n] \rightarrow [r]$, and which want to distinguish the set of positive inputs \mathcal{P} from the set of negative inputs \mathcal{N} , that is, accept all $f \in \mathcal{P}$ and reject all $g \in \mathcal{N}$. If \mathcal{T} is a deterministic decision tree, then $\mathcal{T}(f)$ denotes the terminal leaf of the decision tree \mathcal{T} on input f .

Lemma 6.1. *Let \mathcal{P} and \mathcal{N} be two disjoint sets of functions mapping $[n]$ to $[r]$. Let A and B be sets of labels, and assume there are mappings $A \ni a \mapsto f_a \in \mathcal{P}$ and $B \ni b \mapsto g_b \in \mathcal{N}$. Let μ and ν be two probability measures supported on A and B , respectively.*

Assume that for every deterministic decision tree \mathcal{T} of depth q , one can find a partial mapping $\eta: B \rightarrow A$ such that

- $\mathcal{T}(f_{\eta(b)}) = \mathcal{T}(g_b)$ for every b in the domain of η ;
- $\mu(\eta(B)) = \Omega(1)$;
- $\nu(\eta^{-1}(a)) = \Omega(\mu(a))$ for every $a \in \eta(B)$.

Then, every randomised query algorithm distinguishing \mathcal{P} from \mathcal{N} makes $\Omega(q)$ queries.

Proof. Assume $\mu(\eta(B)) \geq C_1$ and $\nu(\eta^{-1}(a)) \geq C_2\mu(a)$ for every $a \in \eta(B)$, where $C_1, C_2 > 0$ are constants. Performing standard error reduction, we may assume that the error probability of the algorithm is a constant $\epsilon > 0$, which depends on C_1 and C_2 in a way to be determined later.

By standard Yao's principle, there exists a deterministic decision tree \mathcal{T} that accepts with probability $\geq 1 - 2\epsilon$ on f_a where $a \sim \mu$, and rejects with probability $\geq 1 - 2\epsilon$ on g_b where $b \sim \nu$.

Let P be the set of $a \in A$ such that \mathcal{T} accepts f_a . By the first property of η , \mathcal{T} accepts all g_b with $b \in \eta^{-1}(P)$. We have

$$\nu(\eta^{-1}(P)) \geq C_2\mu(P \cap \eta(B)) \geq C_2(C_1 - 2\epsilon).$$

As this quantity is supposed to be less than 2ε , we get a contradiction when $2\varepsilon < C_2(C_1 - 2\varepsilon)$, or

$$\varepsilon < \frac{C_1 C_2}{2(1 + C_2)},$$

which is a positive constant. \square

6.2 The case of $\varepsilon = \Omega(1)$

In this subsection we prove the following theorem, which covers the $\varepsilon = \Omega(1)$ case of Theorem 1.3.

Theorem 6.2. *For an integer k , it takes $\Omega(k)$ queries to $\frac{1}{9}$ -test a function $f: [3^k] \rightarrow [(9k)^{3k}]$ for convexity.*

We will define the required objects from Lemma 6.1. Clearly, $n = 3^k$ and $r = (9k)^{3k}$. The sets \mathcal{P} and \mathcal{N} consist of convex and $1/9$ -far-from-convex functions, respectively.

Let $m = 3k^3$. Denote by $[3]^{<k}$ the set of ternary strings of length strictly less than k , including the empty string. The set A consists of all the functions from $[3]^{<k}$ into $[k^3 - 1]$. For $a \in A$, the value of a on $s \in [3]^{<k}$ is denoted by a_s . We define the function $f_a \in \mathcal{P}$ corresponding to $a \in A$ by giving its discrete derivative, which is a monotone function $\partial f_a: [3^k] \rightarrow [m^k]$. That is,

$$f_a(x) = \sum_{z < x} \partial f_a(z). \quad (5)$$

It is clear that if the function ∂f_a is monotone, the function f_a is convex. Also, the maximal value of f_a is at most $3^k \cdot m^k < (9k)^{3k}$.

The function ∂f_a is defined as follows. Assume that the argument $x \in [3^k]$ is written in ternary and the value $\partial f_a(x) \in [m^k]$ in m -ary. We prepend leading zeroes if necessary so that each number has exactly k digits. We enumerate the digits from left to right with the elements of $[k]$, so that the 0-th digit is the most significant one, and the $(k-1)$ -st digit is the least significant one. We use x_i to denote the i th digit of x . For an interval $[a..b]$, we define $x_{[a..b]}$ as the substring of x formed by the digits x_i as i ranges over $[a..b]$.

Let

$$\phi_a(x, i) = \begin{cases} a_{x_{[i]}} & \text{if } x_i = 0; \\ a_{x_{[i]}} + 1 & \text{if } x_i = 1; \\ m - 2a_{x_{[i]}} - 1 & \text{if } x_i = 2; \end{cases} \quad (6)$$

for $x \in [3^k]$ and $i \in [k]$. The i -th digit of $\partial f_a(x)$ is equal to $\phi(x, i)$. That is,

$$\partial f_a(x) = \sum_{i=0}^{k-1} m^{k-1-i} \phi_a(x, i). \quad (7)$$

Let us make some clarifying comments here. The main case of interest in Equation (6) is $x_i = 0$ and $x_i = 1$. In the far-from-convex case, the first two cases will be essentially switched. This makes the function far from convex, but it is hard to see that just by observing the $x_i = 0$ or $x_i = 1$ case independently. The $x_i = 2$ case is necessary to ensure that the sum of the elements on the right-hand side of Equation (6) is independent of a_s , see Claim 6.4.

Claim 6.3. *Every function f_a is convex.*

Proof. Every function ∂f_a is monotone because $a_s < a_s + 1 < m - 2a_s - 1$ for every $s \in [3]^{<k}$. \square

Claim 6.4. *The value $f_a(x)$ only depends on the values of a_s as s ranges over the prefixes of x , and is independent from the remaining values of a_s .*

Proof. From Equation (5) and Equation (7), we can write

$$f_a(x) = \sum_{i=0}^{k-1} m^{k-1-i} \sum_{z < x} \phi_a(z, i). \quad (8)$$

Note that the sum of the elements on the right-hand-side of Equation (6) is m for every value of $a_{x[i]}$. This means that for every $s \in [3]^{i-1}$ such that $s < x_{[i]}$, we have

$$\sum_{z < x: z_{[i]} = s} \phi_a(z, i) = 3^{k-1-i} \cdot m.$$

The number of such s is exactly $x_{[i]}$. Using this, and summing explicitly over $z < x : z_{[i]} = x_{[i]}$, we get that

$$f(x) = \sum_{i=0}^{k-1} (3m)^{k-1-i} m x_{[i]} + \sum_{i=0}^{k-1} m^{k-i-1} \cdot \begin{cases} x_{[i+1..k]} \cdot a_{x_{[i]}}, & \text{if } x_i = 0; \\ 3^{k-1-i} \cdot a_{x_{[i]}} + x_{[i+1..k]}(a_{x_{[i]}} + 1), & \text{if } x_i = 1; \\ 3^{k-1-i}(2a_{x_{[i]}} + 1) + x_{[i+1..k]}(m - 2a_{x_{[i]}} - 1), & \text{if } x_i = 2. \end{cases}$$

□

The set B is defined as $A \times [k]$. For $a \in A$, $j \in [k]$, and $\delta = \pm 1$, let $a[j, \delta]$ denote the function $b: [3]^{<k} \rightarrow [-1..k^3]$ defined by

$$b_s = \begin{cases} a_s + \delta, & \text{if } |s| = j; \\ a_s, & \text{otherwise.} \end{cases}$$

Note that the value of b_s may lie outside of $[k^3 - 1]$, but the definition f_b still makes sense, and Claim 6.4 still holds.

For $(a, j) \in B$, the corresponding function $g_{a,j}$ is defined by

$$g_{a,j}(x) = \begin{cases} f_{a[j,+1]}(x), & \text{if } x_j = 0; \\ f_{a[j,-1]}(x), & \text{if } x_j = 1; \\ f_a(x), & \text{if } x_j = 2. \end{cases}$$

Claim 6.5. *Every function $g_{a,j}$ is $\frac{1}{9}$ -far from convex.*

Proof. First consider the case $j < k - 1$. Partition the domain of $g_{a,j}$ into 9-tuples which differ only in the j th and the last $(k - 1)$ th ternary digits. In a given 9-tuple, let x and y be the inputs that satisfy $x_j = 0, x_{k-1} = 0$ and $y_j = 1, y_{k-1} = 0$. The definition of $g_{a,j}$ implies that

$$\partial g_{a,j}(x) = m^{k-1-j}(a_{x_{[j]}} + 1) + \sum_{i=0}^{j-1} m^{k-1-i} \phi_a(x, i) + \sum_{i=j+1}^{k-1} m^{k-1-i} \phi_a(x, i)$$

and

$$\partial g_{a,j}(y) = m^{k-1-j} a_{y_{[j]}} + \sum_{i=0}^{j-1} m^{k-1-i} \phi_a(y, i) + \sum_{i=j+1}^{k-1} m^{k-1-i} \phi_a(y, i).$$

Since $y_{[j]} = x_{[j]}$, we have $a_{y_{[j]}} = a_{x_{[j]}}$ and $\phi_a(y, i) = \phi_a(x, i)$ for each $i < j$. Therefore,

$$\begin{aligned} \partial g_{a,j}(y) - \partial g_{a,j}(x) &= -m^{k-1-j} + \sum_{i=j+1}^{k-1} m^{k-1-i} (\phi_a(y, i) - \phi_a(x, i)) \\ &\leq -m^{k-1-j} + \sum_{i=j+1}^{k-1} m^{k-1-i} (m-1) \\ &= -m^{k-1-j} + (m^{k-1-j} - 1) < 0 \end{aligned}$$

and so $\partial g_{a,j}(y) < \partial g_{a,j}(x)$. Any convex function must disagree with $g_{a,j}$ on at least one of the four points $x, x+1, y, y+1$.

The case $j = k-1$ is similar, but only considering the triples which differ in the last, $(k-1)$ st, digit. \square

The probability distributions μ and ν are uniform on A and B , respectively.

Let \mathcal{T} be a deterministic decision tree of depth $q \leq k/2$. Now we define the mapping $\eta: B \rightarrow A$ which depends on \mathcal{T} .

We will define η in the inverse direction, starting from a potential image $a \in A$. Let $Q = \{x_1, \dots, x_q\} \subseteq [3^k]$ be the values which the decision tree \mathcal{T} queries on input of f_a . Denote $S = \{x_{[j]} \mid x \in Q, j \in [k]\}$. We will proceed only if

$$a_s \in [1..k^3 - 2] \quad \text{for all } s \in S. \quad (9)$$

Take $j \in [k]$, and define b as

$$b_s = \begin{cases} a_s - 1 & \text{if } s = x_{[j]} \text{ for some } x \in Q \text{ with } x_j = 0; \\ a_s + 1 & \text{if } s = x_{[j]} \text{ for some } x \in Q \text{ with } x_j = 1; \\ a_s & \text{if } s = x_{[j]} \text{ for some } x \in Q \text{ with } x_j = 2; \\ a_s & \text{otherwise.} \end{cases} \quad (10)$$

if there are no conflicts among the first three cases in this definition. Note that Equation (9) implies that $b \in A$. If b is well-defined, we let $\eta(b, j) = a$.

Claim 6.6. *The mapping η is well-defined and $\mathcal{T}(f_a) = \mathcal{T}(g_{b,j})$ in the above notation.*

Proof. By definition of $g_{b,j}$ and b , and using Claim 6.4, we have that $f_a(x) = g_{b,j}(x)$ for all $x \in Q$. This proves that $\mathcal{T}(f_a) = \mathcal{T}(g_{b,j})$.

Now consider (b, j) in the domain of η . By the previous paragraph it can only come from $a \in A$ such that $\mathcal{T}(f_a) = \mathcal{T}(g_{b,j})$. Then, the set Q is known, and the mapping in Equation (10) can be inverted, proving that η is well-defined. \square

Claim 6.7. *We have $\mu(\eta(B)) = \Omega(1)$ and $|\eta^{-1}(a)| \geq k/2$ for every $a \in \eta(B)$.*

Proof. We will prove first that $a \in \eta(B)$ if condition Equation (9) is satisfied. Indeed, in this case, we do not set $\eta(b, j) = a$ only if there are two inputs $x, y \in Q$ such that $x_{[j]} = y_{[j]}$ and $x_j \neq y_j$. By a simple modification of [1, Lemma 6], there are at most $|Q| - 1$ values of j for which this happens. As $|Q| \leq k/2$, this proves the second part of the claim.

For the first part of the claim, the probability that Equation (9) does not hold is upper bounded by the union bound over at most $k/2$ elements of Q and k prefixes of each $x \in Q$ as

$$\Pr_{a \sim \mu}[\text{Equation (9) does not hold}] \leq \frac{k}{2} \cdot k \cdot \frac{2}{k^3 - 1} = o(1). \quad \square$$

Now we can apply Lemma 6.1 and get that complexity of 1/9-testing functions for convexity is $\Omega(k) = \Omega(\log n)$ as required.

6.3 General lower bound for the line

The lower bound can be strengthened for general values of ϵ as follows.

Theorem 6.8. *Fix any $\frac{1}{n} \leq \epsilon \leq \frac{1}{9}$. Any ϵ -tester for convexity of functions $[n] \rightarrow \mathbb{Z}$ has query complexity*

$$\Omega\left(\frac{1}{\epsilon} \log(\epsilon n)\right)$$

The proof of Theorem 6.8 is a slight extension of the proof of Theorem 6.2.

Define $\ell = \lceil \frac{1}{9\epsilon} \rceil$, $k = \lfloor \log_3 \frac{n}{\ell} \rfloor$, and $m = 3k^3$. We will show that ϵ -testing the convexity of a function mapping $[\ell 3^k] \rightarrow \mathbb{Z}$ requires $\Omega(\ell k)$ queries.

We will use notations with tilde for objects referring to the proof of Theorem 6.8, and non-tilde notation for the objects from Section 6.2.

Let A be as in Section 6.2, and define $\tilde{A} = A^\ell$. For $a \in \tilde{A}$, we have $a = (a^0, \dots, a^{\ell-1})$ with each $a^t \in A$. The partial derivative is given by

$$\tilde{\partial} f_a(t \cdot 3^k + x) = t \cdot m^k + \partial f_{a^t}(x),$$

for $t \in [\ell]$, $x \in [3^k]$, and ∂f_a as in Section 6.2. The function \tilde{f}_a is given by

$$\tilde{f}_a(t \cdot 3^k + x) = \sum_{z < t \cdot 3^k + x} \tilde{\partial} f_a(t \cdot 3^k + x).$$

Similarly to Claims Claim 6.3 and Claim 6.4, we have the following result

Claim 6.9. *Every function \tilde{f}_a is convex. The value of $\tilde{f}_a(t \cdot 3^k + x)$ only depends on the values of a_s^t as s runs through the prefixes of x .*

The set \tilde{B} is defined as $\tilde{A} \times [\ell] \times [k]$. For $a \in \tilde{A}$, define $a[t, j, \delta]$ as $b = (b^0, \dots, b^{\ell-1})$ with $b^t = a^t[j, \delta]$ and $b^u = a^u$ for $u \neq t$. Then,

$$\tilde{g}_{a,t,j}(t \cdot 3^k + x) = \begin{cases} \tilde{f}_{a[t,j,+1]}(t \cdot 3^k + x), & \text{if } x_j = 0; \\ \tilde{f}_{a[t,j,-1]}(t \cdot 3^k + x), & \text{if } x_j = 1; \\ \tilde{f}_a(t \cdot 3^k + x), & \text{if } x_j = 2. \end{cases}$$

Claim 6.10. *Every function \tilde{g}_a is ϵ -far from convex.*

Proof. This is due to the fact that there are $3^{k-2} \geq \epsilon \cdot \ell 3^k$ disjoint pairs of values $x < y$ for which $\partial g(y) > \partial g(x)$, as in the proof of Claim 6.5. \square

The probability distributions μ and ν are defined as uniform on \tilde{A} and \tilde{B} , respectively.

The mapping $\eta: \tilde{B} \rightarrow \tilde{A}$ is also defined similarly to Section 6.2. Let \mathcal{T} be a deterministic decision tree of depth $q \leq \frac{\ell k}{4}$. Take $a \in \tilde{A}$. Let Q be the set of variables queried by \mathcal{T} on \tilde{f}_a , and let $Q^t = \{x \in [3^k] \mid t \cdot 3^k + x \in Q\}$. For $(t, j) \in [\ell] \times [k]$, let

$$b_s^t = \begin{cases} a_s^t - 1 & \text{if } s = x_{[j]} \text{ for some } x \in Q^t \text{ with } x_j = 0; \\ a_s^t + 1 & \text{if } s = x_{[j]} \text{ for some } x \in Q^t \text{ with } x_j = 1; \\ a_s^t & \text{if } s = x_{[j]} \text{ for some } x \in Q^t \text{ with } x_j = 2; \\ a_s^t & \text{otherwise.} \end{cases} \quad (11)$$

and $b = (a^0, \dots, a^{t-1}, b^t, a^{t+1}, \dots, a^{\ell-1})$. We call the pair (t, j) good if there are no conflicts in the first three cases of Equation (11) (that is, b_s^t is well-defined) and $b \in \tilde{B}$. If there are at least $\ell k/2$ good pairs, we define $\eta(b, t, j) = a$ for each good pair (t, j) , where b , of course, depends on t and j .

Similarly to Claim 6.6, η is well-defined and $\mathcal{T}(\hat{f}_{\eta(b)}) = \mathcal{T}(\tilde{g}_b)$ for every b in the domain of η . Also, by definition, $\nu(\eta^{-1}(a)) = \Omega(\mu(a))$ for every $a \in \eta(\tilde{B})$. In order to apply Lemma 6.1, it remains to show the following.

Claim 6.11. *We have $\mu(\eta(B)) = \Omega(1)$.*

Proof. Fix $a \in \tilde{A}$. Similarly to Claim 6.7, there can be at most $q - 1 < \frac{\ell k}{4}$ pairs such that there is a contradiction in the first three cases of Equation (11). A pair (t, j) can be bad also because a_s^t equals 0 or $k^3 - 2$. The expected number of such pairs as $a \sim \tilde{A}$ is $q \cdot k \cdot \frac{2}{k^3 - 1} = O(\frac{\ell}{k})$. By Markov's inequality, probability that the number of such pairs is $\geq \frac{\ell k}{4}$ is $o(1)$. And if this does not happen, the number of good pairs is at least $\ell k/2$. \square

Acknowledgements

Aleksandrs Belovs is supported by the ERDF grant number 1.1.1.2/VIAA/1/16/113. Eric Blais and Abhinav Bommireddi are funded by an NSERC Discovery grant.

References

- [1] Aleksandrs Belovs. Adaptive lower bound for testing monotonicity on the line. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018*, pages 31:1–31:10, 2018. 6, 16, 19
- [2] Omri Ben-Eliezer. Testing local properties of arrays. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 11:1–11:20, 2019. 2, 3, 4
- [3] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. l_p testing. In *STOC*, pages 164–173, 2014. 2
- [4] Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *Proceedings of the 29th Conference on Computational Complexity (CCC)*, pages 309–320, 2014. 2, 3, 4, 22
- [5] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74, 2002. 3
- [6] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and lipschitz testing over hypercubes and hypergrids. In *Symposium on Theory of Computing Conference (STOC '13)*, pages 419–428, 2013. 4
- [7] Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014. 4
- [8] Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000. 2

- [9] Shahar Fattal and Dana Ron. Approximating the distance to convexity. 2010. [3](#)
- [10] Eldar Fischer. On the strength of comparisons in property testing. *Inf. Comput.*, 189(1):107–116, 2004. [3](#)
- [11] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. [2](#)
- [12] Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM J. Comput.*, 37(4):1107–1138, 2007. [2](#)
- [13] Godfrey Harold Hardy and Edward Maitland Wright. *An introduction to the theory of numbers*. Oxford University Press, 1979. [12](#)
- [14] Ashley Montanaro and Ronald de Wolf. A survey of quantum property testing. *Theory of Computing, Graduate Surveys*, 7:1–81, 2016. [3](#)
- [15] Kazuo Murota. Discrete convex analysis. *Math. Program.*, 83:313–371, 1998. [23](#)
- [16] Ramesh Krishnan S Pallavoor, Sofya Raskhodnikova, and Nithin Varma. Parameterized property testing of functions. *ACM Transactions on Computation Theory (TOCT)*, 9(4):17, 2018. [2](#)
- [17] M. Parnas, D. Ron, and R. Rubinfeld. On testing convexity and submodularity. *SIAM Journal on Computing*, 32(5):1158–1184, 2003. [2](#), [3](#), [4](#)
- [18] Thomas Rothvoss. Integer optimization and lattices. 2015. [12](#)
- [19] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. [2](#)
- [20] C. Seshadhri and Jan Vondrák. Is submodularity testable? *Algorithmica*, 69(1):1–25, 2014. [5](#)
- [21] Min Yan. Extension of convex function. *arXiv preprint arXiv:1207.0944*, 2012. [24](#)

A On convexity and line convexity

In the introduction, we mentioned that the notion of *linear convexity* studied in [4] is not equivalent to the notion of convexity we study in this current work. In this section, we provide a proof of this statement.

Definition A.1. Fix a set $X \subseteq \mathbb{R}^d$. The function $f: X \rightarrow \mathbb{R}$ is *linearly convex* if for every $x, y \in X$ and every $0 \leq \lambda \leq 1$ for which $\lambda x + (1 - \lambda)y \in X$, we have $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$.

When $X = \mathbb{R}^d$ or, more generally, when X is a convex set, then the notion of linear convexity is equivalent to convexity. When X is a discrete set with dimension $d \geq 2$, however, the two definitions are not equivalent.

Proposition A.1. *For any $d \geq 2$ and any discrete set $X \subseteq \mathbb{R}^d$, every convex function $f: X \rightarrow \mathbb{R}$ is also linearly convex. However, for every $d \geq 2$ there are discrete sets $X \subseteq \mathbb{R}^d$ for which there exist linearly convex functions $g: X \rightarrow \mathbb{R}$ that are not convex.*

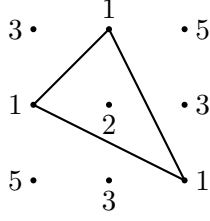


Figure 1: Illustration of the function f constructed in the proof of Proposition A.1.

Proof. That every convex function f is also linearly convex follows directly from the definitions. For the second statement, consider the function $f: [3] \times [3] \rightarrow \mathbb{R}$ defined by

$$\begin{array}{lll} f(0, 2) = 3 & f(1, 2) = 1 & f(2, 2) = 5 \\ f(0, 1) = 1 & f(1, 1) = 2 & f(2, 1) = 3 \\ f(0, 0) = 5 & f(1, 0) = 5 & f(2, 0) = 1 \end{array}$$

The function f is linearly convex, but it has a violation of convexity on the point $(1, 1)$ with respect to the points $(2, 0)$, $(0, 1)$, and $(1, 2)$. \square

We note that many other notions of convexity of functions over discrete domains have also been considered in the context of discrete convex analysis. See [15] and the references therein for more details on those notions.

B Missing proofs from Section 2

For completeness, we include proofs of Lemma 2.2 and Theorem 2.3 in this section.

B.1 Proof of Lemma 2.2

By convexity of f , we have that $g(x) = f(x)$ for all $x \in X$. Hence, g indeed extends f . It remains to prove that g is convex.

Claim B.1. *The definition of g in Equation (2) does not change if we minimise over all possible convex combinations $z = \lambda_1 x_1 + \dots + \lambda_k x_k$, where x_1, \dots, x_k need not form a simplex.*

Proof. Let $g(z)$ be defined as in the statement of this claim. Take a linear combination $z = \lambda_1 x_1 + \dots + \lambda_k x_k$ which minimises $\sum_i \lambda_i f(x_i)$ and such that k is as small as possible. We claim that then x_1, \dots, x_k form a simplex.

Indeed, assume x_1, \dots, x_k are not affinely independent. Then, there exists a non-trivial linear combination $\beta_1 x_1 + \dots + \beta_k x_k = 0$ such that $\beta_1 + \dots + \beta_k = 0$. Changing the sign of each β_i if necessary, we may assume that $\beta_1 f(x_1) + \dots + \beta_k f(x_k) \geq 0$. Let $t \geq 0$ be the maximal real number such that $\lambda_i - t\beta_i \geq 0$ for all i .

Let $\lambda'_i = \lambda_i - t\beta_i$. We have that $\lambda'_i \geq 0$, $\sum_i \lambda'_i = 1$, $\sum_i \lambda'_i x_i = z$, and $\sum_i \lambda'_i f(x_i) \leq \sum_i \lambda_i f(x_i)$. Moreover, at least one of λ'_i is equal to 0, which contradicts minimality of k . \square

Claim B.2. *The function g is convex on the convex hull of X .*

Proof. Consider a convex combination $z = \mu_1 z_1 + \dots + \mu_k z_k$, where all z_i lie in the convex hull of X . For each z_i choose a convex combination $z_i = \sum_{x \in X} \lambda_{i,x} x$ such that $g(z_i) = \sum_{x \in X} \lambda_{i,x} f(x)$. Then, $\lambda_x = \sum_i \mu_i \lambda_{i,x}$ give a convex combination over the elements of X such that $z = \sum_{x \in X} \lambda_x x$. By Claim B.1,

$$g(z) \leq \sum_{x \in X} \lambda_x f(x) = \mu_1 g(z_1) + \dots + \mu_k g(z_k),$$

proving that the function g is convex. \square

By [21], the function g can be extended from the convex hull of X to the whole \mathbb{R}^d . This completes the proof of Lemma 2.2.

B.2 Proof of Theorem 2.3

Assume that f is not convex. Then there exists a convex combination $z = \lambda_1 x_1 + \dots + \lambda_k x_k$ such that $f(z) > \lambda_1 f(x_1) + \dots + \lambda_k f(x_k)$. Choose such a convex combination that k is as small as possible and the convex hull of x_1, \dots, x_k is inclusion-wise minimal. We claim that then x_1, \dots, x_k, z form a minimal centred simplex.

Using the same argument as in Claim B.1, we get that x_1, \dots, x_k is a simplex. Assume it contains more than two points in its convex hull minus the vertices. Let z be such that the violation $f(z) - \lambda_1 f(x_1) - \dots - \lambda_k f(x_k) > 0$ is as large as possible. Let y be any other point in the convex hull of x_1, \dots, x_k except for its vertices. Then,

$$f(z) - \lambda_1 f(x_1) - \dots - \lambda_k f(x_k) \geq f(y) - \mu_1 f(x_1) - \dots - \mu_k f(x_k), \quad (12)$$

where $y = \mu_1 x_1 + \dots + \mu_k x_k$. Let $t \geq 0$ be the largest real number such that $\lambda_i - t\mu_i \geq 0$ for all i . Let $\lambda'_i = \lambda_i - t\mu_i$. We have the following convex combination:

$$z = ty + \lambda'_1 x_1 + \dots + \lambda'_k x_k.$$

Moreover, one of λ'_i is equal to 0. As $z \neq y$, we have that $t < 1$. This together with Equation (12) yields

$$f(z) > tf(y) + \lambda'_1 f(x_1) + \dots + \lambda'_k f(x_k).$$

This contradicts inclusion-wise minimality of x_1, \dots, x_k .