Theory of Computing Systems manuscript No.
(will be inserted by the editor)

Multi-Linearity Self-Testing with Relative Error *

Frédéric Magniez**

CNRS–LRI, Université Paris–Sud, 91405 Orsay, France, e-mail: magniez@lri.fr.

The date of receipt and acceptance will be inserted by the editor

Abstract We investigate self-testing programs with relative error by allowing error terms proportional to the function to be computed. In the self-testing literature for numerical computations, only absolute errors and sublinear (in the input size) errors were previously studied. We construct new self-testers with relative error for real-valued multi-linear functions defined over finite rational domains. The existence of such self-testers positively solves an open question in [KMS03].

Key words Program checking – Self-Testing – Relative Error – Robustness - Stability - Linearity Equation

1 Introduction

1.1 Motivation

By the nature of floating point computation, a program P purposed to implement a real-valued function f can only compute an approximation of it. The accumulation of inaccuracies in numerical operations could be significant. Moreover, once P is implemented, it is difficult to verify its correctness, *i.e.* that $P(x)$ is a good approximation of $f(x)$ for all valid inputs x. In a good approximation one would like the significant digits to be correct. This leads to the notion of relative error. If a is a real number and $\hat{a}$ is its approximation, then the quantity $\theta = |\hat{a} - a|/a$ is called the *relative error* of the approximation. Relative error is one of the most important notions in numerical computation. Proving that a program is relatively close to its correct implementation is the challenge of many numerical analysts.

In recent years, several tools, concepts and theories were developed to address the software correctness problem such as formal methods, model checking, software testing. Here we focus on the following scenario. Firstly, the program to be tested is viewed as a black box, *i.e.* we can only query it on any chosen inputs. Secondly, we want a very efficient testing procedure. In particular, a test should be more efficient than any known correct program (see [BK95] for a formal definition). For exact computation, program checking [Blu88, BK95], self-testing programs [BLR93], and self-correcting programs [BLR93, Lip91] were developed in the early 90's. A *program checker* for f verifies whether the program P computes f on a particular input x; a *self-tester* for f verifies whether the program P is correct on most inputs; and a *self-correcting program* for f uses a program P, which is correct on most inputs, to compute f correctly everywhere with high probability. More formally, self-testing for exact computation consists in the following task.

* A preliminary version of this paper appeared in *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science*, LNCS 1770, pages 302–313, 2000.

** Partially supported by a CNRS–Conicyt'98 Project, ESPRIT Working Group RAND-APX IST-1999-14036, and by CNRS/STIC OBS050 grant.

Problem *Given a class of functions $\mathcal{F}$ defined over a finite domain D, and positive constants δ_1, δ_2, we want an efficient probabilistic algorithm T with oracle such that, for any program P defined over D:*
- *if for some $f \in \mathcal{F}$, $\mathbf{Pr}_{x\in D}[P(x) \neq f(x)] \leq \delta_1$, then T^P (T with oracle P) accepts with high probability;*
- *if for all $f \in \mathcal{F}$, $\mathbf{Pr}_{x\in D}[P(x) \neq f(x)] > \delta_2$, then T^P rejects with high probability.*

In this context, results on testing linear functions and polynomials have theoretical implications for probabilistically checkable proofs [ALM+98, AS98] and in approximation theory. For a survey see [Bab93].

1.2 Previous work on linearity self-testing

Let us recall the problem of linearity testing which has been fundamental in the development of self-testers [BLR93]. Given a program P which computes a function from a finite abelian group G into another group, we want to verify that P computes a homomorphism on most elements in G.

The Blum-Luby-Rubinfeld linearity test is based on the linearity property $f(x + y) = f(x) + f(y)$, for all $x, y \in G$, which is only satisfied when f is a homomorphism. The test consists of verifying this linearity equation on random instances. More precisely, it checks that $P(x + y) = P(x) + P(y)$, for random inputs $x, y \in G$. Note that checking the linearity equation is usually easier than computing a linear function: it uses only two additions whereas computing a linear function requires a multiplication (when G is a cyclic group).

If the probability of failing the linearity test is small, then P computes a homomorphism except on a small fraction of the inputs. This property of the linearity equation is usually called its *robustness*. This term was defined in [RS96] and studied in [Rub99]. The analysis of the above test is due to Coppersmith [Cop89]. It consists of correcting P by querying it on a few inputs. Let g be the function which takes at x the majority of the votes $(P(x + y) - P(y))$, for all $y \in G$. When the failure probability in the linearity test is small, majority turns to quasi-unanimity, g equals P on a large fraction of inputs, and g is linear. This idea of *property testing* has been formalized and extended to testing graph properties in [GGR98, GR99].

These notions of testing were extended to approximate computation with absolute error for self-testers/correctors [GLR+91] and for checkers [ABCG93]. For a survey see [KMS00]. In [GLR+91], Gemmel, Lipton and Rubinfeld studied functions defined over algebraically closed domains. Ergün, Kumar, and Rubinfeld [EKR01] initiated and solved the problem of self-testing with absolute error for linear functions, polynomials, and additive functions defined over rational domains. Rational domains, *i.e.* sets $D_{n,s} = \{i/s : |i| \leq n, i \in \mathbb{Z}\}$, for some integer $n \geq 1$ and real $s > 0$, were first considered by Lipton [Lip91].

In the case of approximate self-testing for rational domains, when an error is allowed in the linearity test, both the closeness of g to P and the linearity of g are approximate. This is usually called the *approximate robustness* of the linearity equation. Since we want to prove that P is close to a perfectly linear function, a second stage is needed. It consists of proving the *stability* of the linearity equation. This means that every function that approximately satisfies the linearity equation is close everywhere to a linear function. The approximate robustness together with the stability implies the robustness of the linearity equation for absolute error. The stability part is a well-studied problem in mathematics for several kinds of error terms when x and y cover a semi-group like $\mathbb{N}$ or $\mathbb{Z}$. It corresponds to the study of Hyers-Ulam stability.

The stability problem is due to Ulam and was first solved for the absolute error case in 1941 by Hyers [Hye41]. For a survey of Hyers-Ulam stability see [For95, HR92]. Since $D_{n,s}$ is not a semi-group, these results cannot be applied directly. But using an argument due to Skof [Sko83], the local stability of the linearity equation over $D_{n,s}$ can be derived from its stability over the whole domain $\mathbb{Z}$.

Using these elegant techniques of Hyers-Ulam stability theory, Kiwi, Magniez and Santha [KMS03] extended a part of [EKR01]'s work for non-constant error terms. They considered error terms proportional in every input x to $|x|^p$, for some $0 \leq p < 1$, that is, they studied computations where inaccuracies were sublinear in the size of the values involved in the calculations. Among other things, they showed how to self-test whether a program approximately computes a linear function for these error terms. To this end they proved the local stability of the linearity equation using its stability on the whole domain $\mathbb{Z}$. Since the linearity equation is unstable for the case $p = 1$ [HS92], their work did not lead to self-testers either for the case $p = 1$, which corresponds to linear error terms, or for relative error terms (*i.e.* proportional to the function to be computed) [KMS03, Section 5].

1.3 Main result of the paper

In this paper, we investigate the study of approximate self-testing with relative error. Computing with relative error $\theta > 0$ means that the accuracy we allow for computing f on x is $\theta|f(x)|$, and hence depends on the generally unknown value of $f(x)$. In this setting, self-testing consists of the following task.

Problem *Given a class of real-valued functions $\mathcal{F}$ defined over a finite domain D, and positive constants $\theta_1, \theta_2, \delta_1, \delta_2$, we want an efficient probabilistic algorithm T with oracle such that, for any program $P : D \to \mathbb{R}$:*
- *if for some $f \in \mathcal{F}$, $\mathbf{Pr}_{x\in D}\left[|P(x) - f(x)| > \theta_1|f(x)|\right] \leq \delta_1$, then T^P accepts with high probability;*
- *if for all $f \in \mathcal{F}$, $\mathbf{Pr}_{x\in D}\left[|P(x) - f(x)| > \theta_2|f(x)|\right] > \delta_2$, then T^P rejects with high probability.*

But even if f is known, the self-tester cannot compute $f(x)$ since it has to be more efficient than any program computing f. Thus, it is not *a priori* clear that one can self-test in the context of relative error.

Nevertheless, we give a positive answer to this problem for the set of real-valued d-linear functions, for any integer $d \geq 1$. This is the first positive answer to this problem in the literature, and hence it solves some open problems in [KMS03]. We will only deal with self-testing, but similar results can be easily derived for self-correcting (and therefore for program checking) using standard techniques [EKR01].

1.4 Structure of the paper

The rest of the paper is organized as follows. In Section 2, we describe self-testing concepts in the most general notion of approximate computing [KMS00]. In this context, we review basic methods for constructing a self-tester from a functional equation which are based on the well known notions of robustness, approximate robustness, and stability.

A new test for linear functions is defined and studied in Section 3. It is constructed from a new functional equation for linearity which is robust for linear error (Theorem 2). The robustness is proved in two parts : the approximate robustness (Theorem 3) and the stability (Theorem 4). Then, an approximate self-tester for linear functions with linear error is deduced (Corollary 1).

In the final Section 4, two approximate self-testers with relative error are constructed using the preceding self-tester. First, a self-tester for linear functions is presented (Corollary 2). Then, it is generalized to multi-linear functions (Corollary 3) using an argument similar to that in [FHS94]. All of our self-testers only use *simple operations*, that is comparisons, additions, and multiplications by powers of 2. Moreover, the number of queries and simple operations does not depend on the size of the domain where the program is tested.

2 Approximate self-testing

The forthcoming presentation is based on [KMS00]. In this paper we only consider real-valued functions. Throughout this section, let D be a fixed finite set. We will always think on a program P which on any input from D, outputs a real number, as a function from D to $\mathbb{R}$. We will use the word 'program' to denote the function we are testing. We are interested in determining, maybe probabilistically, how "close" a program $P : D \to \mathbb{R}$ is to an underlying family of functions of interest, $\mathcal{F}$. In Sections 3 and 4, D will denote an integer domain $D_n = \{i \in \mathbb{Z} : |i| \leq n\}$ or a power of it $(D_n)^d$, and $\mathcal{F}$ will be the set of linear functions or d-linear functions.

The image elements might be hard to represent (for example, when $\mathcal{F}$ is a family of trigonometric functions). Thus, any reasonable program P for computing $f \in \mathcal{F}$ will necessarily have to compute an approximation. In fact, P might never equal f over D but still be, for all practical purposes, a good computational realization of a program that computes f. Thus we first need to define what we mean by allowing an error in the computation process. For that we define the notion of computational error term.

Definition 1 (Computational error term) *A* computational error term *is a function $\varepsilon : D \times \mathbb{R} \to \mathbb{R}_+$. If $P, f : D \to \mathbb{R}$ are two functions, then P* ε-computes f on $x \in D$ *if $|P(x) - f(x)| \leq \varepsilon(x, f(x))$.*

This definition encompasses several models of approximate computing that depend on the restriction placed on the computational error term ε. Indeed, it encompasses the

– exact computation case, where $\varepsilon(x,v) = 0$ (for every $x \in D$ and $v \in \mathbb{R}$);
– approximate computation with absolute error, where $\varepsilon(x,v) = \varepsilon_0$ for some constant $\varepsilon_0 \in \mathbb{R}_+$;
– approximate computation with error relative to input size, where $\varepsilon(x,v) = \varepsilon_1(x)$ for some function $\varepsilon_1 : D \to \mathbb{R}_+$;
– approximate computation with relative error, where $\varepsilon(x,v) = \theta|v|$ for some constant $\theta \in \mathbb{R}_+$.

Based on the definition of computational error term we can give a notion of distance (even if it is not a distance in general), which measures the proportion of inputs where P ε-computes f.

Definition 2 (ε-Distance) *Let $P, f : D \to \mathbb{R}$, and let ε be a computational error term. The* ε-distance of P from f on D *is*

$$\varepsilon\text{-}\mathsf{Dist}_D(P,f) = \mathop{\mathbf{Pr}}_{x\in D}\left[P \text{ does not } \varepsilon\text{-compute } f \text{ on } x\right].$$

If $\mathcal{F}$ is a family of real functions over D, then the ε-distance of P from $\mathcal{F}$ on D *is*

$$\varepsilon\text{-}\mathsf{Dist}_D(P,\mathcal{F}) = \mathop{\mathrm{Inf}}_{f\in\mathcal{F}} \varepsilon\text{-}\mathsf{Dist}_D(P,f).$$

Now we define the approximate self-tester for general computational error terms which generalizes the previous definitions of [EKR01, GLR+91, KMS03].

Definition 3 (Approximate self-tester) *Let $\mathcal{F}$ be a family of real functions over D, let $D' \subseteq D$, let ε and ε' be computational error terms and let $0 \le \eta \le \eta' < 1$ be constants. A $(D,\varepsilon,\eta;D',\varepsilon',\eta')$*-approximate self-tester *for $\mathcal{F}$ is a randomized oracle algorithm T such that for every $P : D \to \mathbb{R}$:*

– *if $\varepsilon\text{-}\mathsf{Dist}_D(P,\mathcal{F}) \le \eta$, then $\mathbf{Pr}\left[T^P \text{ accepts}\right] \ge 2/3$;*
– *if $\varepsilon'\text{-}\mathsf{Dist}_{D'}(P,\mathcal{F}) > \eta'$, then $\mathbf{Pr}\left[T^P \text{ rejects}\right] \ge 2/3$;*

where the probabilities are taken over the coin tosses of T.

In this definition, note that the success probability 2/3 can be replaced by $1-\gamma$, for any confidence parameter $\gamma > 0$, using a majority argument based on $\log(1/\gamma)$ iterations of T.

Usually the construction of a self-tester is based on the existence of a *test*, that is a randomized oracle algorithm whose acceptance probability is directly connected to the distance of interest. Computing the distance of a function P from a family $\mathcal{F}$ is usually a hard task. On the other hand, the rejection probability of a function P by a test T can be easily approximated by standard sampling techniques.

Therefore, if a test T is such that for every function P, the rejection probability $\mathsf{Rej}(P,T)$ of P by T is closely related to its distance from the function class $\mathcal{F}$ of interest, then by approximating the rejection probability one can estimate the distance. In other words, one obtains a self-tester for $\mathcal{F}$. The two important properties of a test which ensure that this approach succeeds are the soundness and the robustness. The soundness was first defined in [KMS00] as continuity, and the robustness in [RS96, Rub99].

Definition 4 (Soundness & Robustness) *Let $\mathcal{F}$ be a family of real functions over D, let ε be a computational error term, and let $0 \le \eta, \delta < 1$ be constants. Let T be a randomized oracle algorithm. Then, T is* (η,δ)-sound for $\mathcal{F}$ on D with respect to ε *if for all $P : D \to \mathbb{R}$,*

$$\varepsilon\text{-}\mathsf{Dist}_D(P,\mathcal{F}) \le \eta \implies \mathsf{Rej}(P,T) \le \delta.$$

Moreover, T is (η,δ)-robust for $\mathcal{F}$ on D with respect to ε *if for all $P : D \to \mathbb{R}$,*

$$\mathsf{Rej}(P,T) \le \delta \implies \varepsilon\text{-}\mathsf{Dist}_D(P,\mathcal{F}) \le \eta.$$

Thus, proving soundness of a test implies upper bounding the rejection probability of the test in terms of the relevant distance. On the other hand, to prove robustness one needs to upper bound the relevant distance in terms of the rejection probability of the test.

Typically, tests that are both sound and robust give rise to self-testers. We now precisely state this claim which is often used. For a proof see for instance [KMS00].

Theorem 1 (Approximate generic self-tester) *Let $\mathcal{F}$ be a family of real functions over D, let $D' \subseteq D$, let ε and ε' be computational error terms, and let $0 \le \delta < \delta' < 1$ and $0 \le \eta \le \eta' < 1$ be constants. Also, let T be a randomized oracle algorithm such that*

– *T is (η, δ)-sound for $\mathcal{F}$ on D with respect to ε,*
– *T is (η', δ')-robust for $\mathcal{F}$ on D' with respect to ε',*

then the following algorithm is a $(D, \varepsilon, \eta; D', \varepsilon', \eta')$-approximate self-tester for $\mathcal{F}$:

***Generic Self-Tester**($T, \delta, \delta'; P$)*
1. Let $N = \theta(\frac{\delta+\delta'}{(\delta-\delta')^2})$ and $E = 0$.
2. Do N times: Run T and let $E = E+1$ if T rejects.
3. Reject if $\frac{E}{N} > \frac{\delta+\delta'}{2}$ times.

Realizable approximate tests are often constructed through a functional equation whose set of solutions is the function class $\mathcal{F}$. Specifically, let $\mathbb{R}^D$ denote the set of function from D to $\mathbb{R}$, and let $\Phi : \mathbb{R}^D \times \mathcal{N} \to \mathbb{R}$ be a functional over a *neighborhood set* $\mathcal{N} \subseteq D^k$, where $k \geq 1$ is an integer. The functional Φ and a *test error term* $\beta : \mathcal{N} \times \mathcal{F} \to \mathbb{R}_+$ induce the following test:

Functional Equation Test($\Phi, \beta; P$)
1. Randomly choose $(x_1, \ldots, x_k) \in \mathcal{N}$.
2. Reject if $\lvert\Phi(P, x_1, \ldots, x_k)\rvert > \beta(x_1, \ldots, x_k, P)$.

If Φ and β are efficiently computable, then this test is realizable. When the test is sound and robust with respect to some computational error term, Theorem 1 can be applied to derive the corresponding approximate self-tester. The complexity of the self-tester will ultimately depend on the complexity of computing Φ and β.

It is worth pointing out one common aspect of all known analyses of approximate tests based on functional equations, specifically in their proofs of robustness. There are two clearly identifiable stages in such proofs: approximate robustness and stability. The *approximate robustness* consists of proving robustness when the family $\mathcal{F}$ is replaced by the set of approximate solutions of the functional equation, *i.e.* a function having a small rejection probability by the approximate test, is close to a function that approximately everywhere satisfies the functional equation. The *stability* consists of proving that such an approximate solution is everywhere close to a function of $\mathcal{F}$, *i.e.* a function satisfying the functional equation exactly and everywhere.

3 A new test for linearity

For the sake of brevity and clarity we will only consider functions defined over integer domains $D_n = \{i \in \mathbb{Z} : |i| \leq n\}$, for some even integer $n \geq 2$. Nonetheless, all our results remain valid for the more general rational domains $D_{n,s}$, for any positive integer s, by observing that $D_{n,s} = \frac{1}{s} \times D_n$.

The linearity test of [KMS03] is based on the linearity equation $f(x+y) = f(x) + f(y)$ which is robust for test error terms proportional to $|x|^p$, where $0 \leq p < 1$. Note that in the context of linearity testing over integer domains, computational error terms for relative error are functions that map x to $\theta'|x|$ where θ' is some unknown positive constant. More precisely, if f is the closest linear function to P that P is supposed to compute with some relative error θ (where θ is a known parameter), then $\theta' = \theta|f(1)|$. Even if θ' were known, the previous test could not be used since the linearity equation is unstable (and therefore non robust) when $p = 1$, that is when the test error term is linear.

Proposition 1 ([RS92]) *Let $\theta > 0$ and $f(x) = \theta x \log_2(|x|+1)$, for all $x \in \mathbb{Z}$. Then f satisfies $|f(x+y) - f(x) - f(y)| \leq 2\theta \operatorname{Max}\{|x|, |y|\}$, for all x, y, and $\lim_{x\to\infty} \frac{f(x)}{x} = \infty$.*

It follows that, for any real $A > 0$, there exists an integer N such that, for every integer $n \geq N$ and every linear function l, the set $\{x \in D_n : |f(x) - l(x)| > A\theta|l(x)|\}$ is non empty, where f is the function defined above.

Hence for proving the stability, either the equation or the test error term has to be modified. In [KMS03] the linearity test was unchanged and test error terms proportional to $|x|^p$ were considered, for some $0 \leq p < 1$. Since this approach leads to a sublinear computational error term, in this paper we keep the linear error term while we change the test.

When x is *large*, say $|x| \geq n/2$, linear and absolute error terms are essentially the same. When x is *small*, say $0 < |x| < n/2$, we would like to amplify the linear error term to an absolute one. This can be done by multiplying x by the smallest power of 2 such that the absolute value of the result is at least $n/2$. This procedure can be efficiently implemented (for example, by means of binary shifts). Formally, each x is multiplied by 2^{k_x} where

$$k_x = \begin{cases} 0, & \text{if } x = 0, \\ \text{Min}\left\{k \in \mathbb{N} : 2^k|x| \geq n/2\right\}, & \text{otherwise.} \end{cases}$$

(See Figure 1 for an example where $n/8 < x < n/4$.)

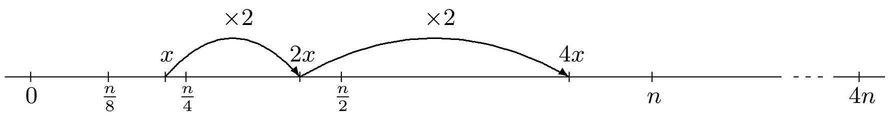

Fig. 1 Amplification procedure.

The amplification procedure described above leads to the following new functional equation characterization of the class of linear functions (whose domain is D_{8n}):

$$\forall x, y \in D_{4n}, \qquad f(2^{k_x}x + y) - 2^{k_x}f(x) - f(y) = 0.$$

Note how this new characterization of linear functions relies not only on their additive, but also on their homothetic properties (namely, $f(\lambda x) = \lambda f(x)$). In the standard way, it leads to a functional equation test. Specifically, for $\theta \geq 0$ it yields the following:

Linear error Linearity Test(P, θ)
1. Randomly choose $x, y \in D_{4n}$.
2. Reject if $|P(2^{k_x}x + y) - 2^{k_x}P(x) - P(y)| > \theta 2^{k_x}|x|$.

Note that the test error $\theta 2^{k_x}|x|$ is actually pretty close to an absolute error. When $x \neq 0$, it is in $[\theta n/2, 4\theta n]$, and when $x = 0$ it equals 0. In a first approach, the reader might thinkof it as the absolute error θn. For technical reasons, we want the program P to pass the test for $x = 0$ only if $P(0) = 0$ (exactly). Therefore the test error term has to be 0 in 0.

The rest of the paper is based on the robustness of this test for linear computational error terms. We henceforth denote by $\mathsf{Rej}(P, \theta)$ the rejection probability of the **Linear error Linearity test**(P, θ), and by $\theta|x|$ the computational error term $\varepsilon(x, v) = \theta|x|$. Let $\text{Med}_{x \in X}(f(x))$ denote the median value of $f : X \to R$ when x ranges over X:

$$\underset{x \in X}{\text{Med}}(f(x)) = \text{Inf}\left\{a \in \mathbb{R} \,:\, \underset{x \in X}{\mathbf{Pr}}[f(x) \geq a] \leq 1/2\right\}.$$

Then the robustness can be stated as:

Theorem 2 *Let $\theta \geq 0$, $0 \leq \delta < 1/512$, $P : D_{8n} \to \mathbb{R}$, and let $l : D_n \to \mathbb{R}$ be the linear function such that*

$$l(n) = \underset{y \in D_{2n}: y \geq 0}{\text{Med}}(P(n + y) - P(y)).$$

Then,

$$\mathsf{Rej}(P, \theta) \leq \delta \implies (137\theta|x|)\text{-}\mathsf{Dist}_{D_n}(P, l) \leq 32\delta.$$

The proof of this theorem proceeds in two parts: the approximate robustness (Theorem 3), and the stability (Theorem 4). Together with the soundness (Lemma 4), the robustness of the test implies by Theorem 1 that repeating **Linear error Linearity test** $O(\frac{1}{\delta})$ times gives a self-tester for linear functions with linear computational error term (Corollary 1).

3.1 Basic tools

Here we state two simple lemmas which will be repeatedly applied in the forthcoming sections. For their proofs see [KMS00].

The first lemma is an extension of some standard majority principles to the median function. The importance of the median function in the context of approximate self-testing was recognized by Ergün, Kumar and Rubinfeld in [EKR01], where the median principle was also introduced.

Lemma 1 (Median principle) *Let D, D' be two finite sets. Let $\varepsilon \geq 0$ and $F : D \times D' \to \mathbb{R}$. Then,*

$$\mathop{\mathbf{Pr}}_{x\in D}\left[\left|\mathop{\mathrm{Med}}_{y\in D'}(F(x,y))\right| > \varepsilon\right] \leq 2 \mathop{\mathbf{Pr}}_{(x,y)\in D\times D'}\left[|F(x,y)| > \varepsilon\right].$$

The second lemma quantifies how much an event might increase when the probability space is shrunk. The fact that the halving principle can substantially simplify the standard proof arguments one encounters in the approximate testing scenario was observed in [KMS03].

Lemma 2 (Halving principle) *Let Ω and S denote finite sets such that $S \subseteq \Omega$, and let ψ be a boolean function defined over Ω. Then,*

$$\mathop{\mathbf{Pr}}_{x\in S}[\psi(x)] \leq \frac{|\Omega|}{|S|}\mathop{\mathbf{Pr}}_{x\in\Omega}[\psi(x)].$$

3.2 Approximate robustness

Assuming that the rejection probability of the test is small, we will construct from P a function g which is not linear, but approximately linear for large inputs, and perfectly homothetic for small inputs. In a sense g approximately corrects the program P.

The following theorem states the existence of such a function g. The definition of g is based on the test and it consists of computing, for some $x \in D_{2n}$, the median of the votes $(P(2^{k_x}x + y) - P(y))/2^{k_x}$ for all $y \in D_{2n}$ such that $xy \geq 0$. We will explain this restriction on the sign of y during the proof.

Theorem 3 *Let $0 \leq \delta < 1/512$ and $\theta \geq 0$. Let $P : D_{8n} \to \mathbb{R}$ be such that*

$$\mathop{\mathbf{Pr}}_{x,y\in D_{4n}}\left[|P(2^{k_x}x + y) - 2^{k_x}P(x) - P(y)| > \theta 2^{k_x}|x|\right] \leq \delta.$$

Then, the function $g : D_{2n} \to \mathbb{R}$ defined by

$$g(x) = \frac{1}{2^{k_x}}\mathop{\mathrm{Med}}_{y\in D_{2n}:xy\geq 0}\left(P(2^{k_x}x + y) - P(y)\right),$$

is such that

$$\mathop{\mathbf{Pr}}_{x\in D_n}\left[|P(x) - g(x)| > \theta|x|\right] \leq 32\delta.$$

Moreover, $g(x) = g(2^{k_x}x)/2^{k_x}$ for all $x \in D_{2n}$, $g(0) = 0$, $|g(n) + g(-n)| \leq 16\theta n$, and for all $x, y \in \{n/2, \ldots, n\}$ (respectively $x, y \in \{-n/2, \ldots, -n\}$)

$$|g(x + y) - g(x) - g(y)| \leq 24\theta n.$$

Proof The proof uses standard techniques developed in [BLR93,EKR01,KMS03]. First, observe that the median function g satisfies $g(0) = 0$ and

$$g(x) = \begin{cases} \mathop{\mathrm{Med}}_{y\in D_{2n}:xy\geq 0}\left(P(x + y) - P(y)\right), & \text{if } |x| \geq n/2, \\ g(2^{k_x}x)/2^{k_x}, & \text{for every } x \in D_{2n}. \end{cases}$$

We show now that g is close to P. To simplify the notation, let $P_{x,y} = P(2^{k_x}x+y) - P(y) - 2^{k_x}P(x)$. By the median principle and the definition of g we get

$$\begin{aligned}\Pr_{x\in D_n}\left[|g(x)-P(x)| > \theta|x|\right] &= \Pr_{x\in D_n}\left[|\operatorname*{Med}_{y\in D_{2n}:xy\geq 0}(P_{x,y})| > \theta 2^{k_x}|x|\right]\\ &\leq 2\Pr_{x\in D_n;y\in D_{2n}:xy\geq 0}\left[|P_{x,y}| > \theta 2^{k_x}|x|\right].\end{aligned}$$

Observe that the rejection probability is exactly the right hand side when the random variables are both in D_{4n}. To get the required closeness of g to P, one application of the halving principle, where the small domain is $\{(x,y) : x \in D_n; y \in D_{2n}\}$ and the big domain is $(D_{4n})^2$, upper bounds the right hand side by $2 \times 16 \times \delta$.

Now we prove the approximate additivity of g in x and y, when x, y and $x+y$ have no amplification factors associated to them, that is, when both x and y belong to either $\{n/2,\ldots,n\}$ or $\{-n/2,\ldots,-n\}$, or when $\{x,y\} = \{-n,n\}$. This partly justifies the restriction on the set of elements of D_{2n} where the median is computed: When $xy \geq 0$ one knows that the absolute value of $2^{k_x}x+y$ is at least $n/2$. This fact will be used later.

First, we prove that for all $c \in D_{2n}$, such that $|c| \geq n/2$, and all non empty sets $I \subseteq \{t \in D_{2n} : |c+t| \geq n/2\}$, we have

$$\Pr_{t\in I}\left[|g(c)-(P(c+t)-P(t))| > 8\theta n\right] \leq 16\frac{|D_{4n}|}{|I|}\delta. \tag{1}$$

Observe that the absolute values of c, $c+y$ and $c+z$ are all at least $n/2$ when $cy \geq 0$ and $z \in I$. Then

$$\begin{aligned}&\Pr_{t\in I}\left[|g(c)-(P(c+t)-P(t))| > 8\theta n\right]\\ &\leq \Pr_{t\in I}\left[|\operatorname*{Med}_{y\in D_{2n}:cy\geq 0}(P(c+y)-P(y))-(P(c+t)-P(t))| > 8\theta n\right]\\ &\qquad\text{(by definition of } g)\\ &\leq 2\Pr_{\substack{t\in I;y\in D_{2n}\\ cy\geq 0}}\left[|P(c+y)+P(t)-(P(c+t)+P(y))| > 8\theta n\right]\\ &\qquad\text{(from the median principle)}\\ &= 2\Pr_{\substack{t\in I;y\in D_{2n}\\ cy\geq 0}}\left[|P_{c+y,t}-P_{c+t,y}| > 8\theta n\right]\\ &\leq 2\Pr_{\substack{t\in I;y\in D_{2n}\\ cy\geq 0}}\left[|P_{c+y,t}| > \theta|c+y|\right] + 2\Pr_{\substack{t\in I;y\in D_{2n}\\ cy\geq 0}}\left[|P_{c+t,y}| > \theta|c+t|\right],\end{aligned}$$

where the last inequality comes from the union bound and the fact that $|c+y|, |c+t| \leq 4n$. Then the halving principle gives (1).

Now, let us show that $g(-n)$ is close to $-g(n)$. From inequality (1), there exists an integer $t \in \{0,1,\ldots,n/2\}$, with probability $1-512\delta > 0$, such that

$$\begin{aligned}|g(-n)-(P(-n+t)-P(t))| &\leq 8\theta n,\\ |g(n)-(P(n+(-n+t))-P(-n+t))| &\leq 8\theta n.\end{aligned}$$

Thus we get the upper bound on $|g(n)+g(-n)|$. Finally, we show the approximate additivity of g in $x,y \in \{n/2,\ldots,n\}$ (respectively $x,y \in \{-n/2,\ldots,-n\}$). Again from inequality (1), there exists an integer $t \in \{0,1,\ldots,n\}$ (respectively $t \in \{0,-1,\ldots,-n\}$), with probability $1-384\delta > 0$, such that

$$\begin{aligned}|g(x+y)-(P(x+y+t)-P(t))| &\leq 8\theta n,\\ |g(x)-(P(x+t)-P(t))| &\leq 8\theta n,\\ |g(y)-(P(y+(x+t))-P(x+t))| &\leq 8\theta n.\end{aligned}$$

This concludes the proof. $\square$

Note that the approximate additivity of g over $\{n/2, \ldots, n\}$ and $\{-n/2, \ldots, -n\}$ established by the previous result guarantees, due to g's homothetic property, its approximate additivity over small elements of the same amplification factor.

3.3 Stability

We now prove that any function g which approximately satisfies the linearity equation on large inputs and which is perfectly homothetic on small inputs, is close to a perfectly linear one. This is the second part of the proof of the robustness for **Linear error Linearity Test**.

Theorem 4 *Let $\theta_1, \theta_2 \geq 0$. Let $g : D_{2n} \to \mathbb{R}$ be such that $g(x) = g(2^{k_x}x)/2^{k_x}$ for all $x \in D_{2n}$, $g(0) = 0$, $|g(n) + g(-n)| \leq \theta_1 n$, and for all $x, y \in \{n/2, \ldots, n\}$ (respectively $x, y \in \{-n/2, \ldots, -n\}$),*

$$|g(x + y) - g(x) - g(y)| \leq \theta_2 n.$$

Then, the linear function $l : D_n \to \mathbb{R}$ defined by $l(n) = g(n)$ satisfies, for all $x \in D_n$,

$$|g(x) - l(x)| \leq (\theta_1 + 5\theta_2)|x|.$$

Before proving the theorem, we state the following lemma that we will use. The result is due to [KMS03, KMS00] which is based on some techniques developed in [Hye41]. Note that this lemma only states the closeness between a function and its 'limit function'. In general, the limit function is not linear, but in our context, it will be clear that it is linear.

Lemma 3 ([KMS00, Lemma 4]) *Let E_1 be a semi-group and E_2 a Banach space. Let $\varepsilon \geq 0$ and $h : E_1 \to E_2$ be a mapping such that for all $x \in E_1$*

$$\|h(2x) - 2h(x)\| \leq \varepsilon.$$

Then the function $f : E_1 \to E_2$ defined by $f(x) = \lim_{m\to\infty} h(2^m x)/2^m$ is a well defined mapping such that for all $x \in E_1$,

$$\|h(x) - f(x)\| \leq \varepsilon.$$

Proof (of Theorem 4) First we will show that g is close to the linear function l (respectively l') defined by $l(n) = g(n)$ (respectively $l'(-n) = g(-n)$) on the positive (respectively negative) part of the domain. Then the proof is completed by observing that l and l' are necessarily close to each other.

We borrow a technique developed in [KMS03] that we apply to the function g. First we extend the restriction of g to $\{n/2, \ldots, n\}$ into a function h defined over the whole semi-group $\{x \in \mathbb{N} : x \geq n/2\}$ (see Figure 2). The extension h is defined for all $x \geq n/2$ by

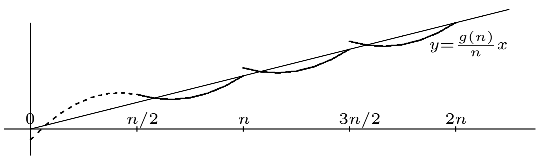

Fig. 2 The extension h on positive integers.

$$h(x) = \begin{cases} g(x) & \text{if } n/2 \leq x \leq n, \\ h(x - n/2) + g(n)/2 & \text{otherwise.} \end{cases}$$

One can verify that h satisfies the following doubling property, for all $x \geq \frac{n}{2}$,

$$|h(2x) - 2h(x)| \leq 5\theta_2 n/2.$$

Therefore Lemma 3 implies that the limit function $f(x) = \lim_{m\to\infty} h(2^m x)/2^m$ is close to h:

$$\forall x \geq n/2, \quad |h(x) - f(x)| \leq 5\theta_2 n/2.$$

Using the definitions of f and h, we obtain that f is the linear function l. Moreover, since $h(x) = g(x)$ and $n/2 \leq |x|$ when $x \in \{n/2, \dots, n\}$, we have

$$\forall x \in \{n/2, \dots, n\}, \quad |g(x) - l(x)| \leq 5\theta_2 |x|.$$

Using that g is perfectly homothetic and $g(0) = 0$, the previous inequality is still valid when $x \in \{0, \dots, n\}$.

One can similarly prove that l' satisfies the same property over $\{-n, \dots, 0\}$. Then the closeness of $g(n)$ and $g(-n)$ concludes the proof. □

3.4 Testing linearity with linear error

Theorem 2 states the robustness of **Linear error Linearity Test**. In order to use the generic approximate self-tester of Theorem 1, one has to prove also the soundness of **Linear error Linearity Test**. In the following $\mathcal{L}$ denotes the class of real-valued linear functions over D_{8n}.

Lemma 4 *Let $\theta \geq 0$. Then every $P : D_{8n} \to \mathbb{R}$ satisfies*

$$\mathsf{Rej}(P, \theta) \leq 6 \times (\theta|x|/18)\text{-}\,\mathsf{Dist}_{D_{8n}}(P, \mathcal{L}).$$

Proof Let $\eta \geq 0$, $P : D_{8n} \to \mathbb{R}$, and $l \in \mathcal{L}$ be such that $(\theta|x|/18)\text{-}\,\mathsf{Dist}_{D_{8n}}(P, l) = \eta$. By the halving principle we have

$$\begin{aligned} \mathop{\mathbf{Pr}}_{x \in D_{4n}} \left[|P(x) - l(x)| > \theta|x|/18 \right] &\leq 2\eta, \\ \mathop{\mathbf{Pr}}_{y \in D_{4n}} \left[|P(y) - l(y)| > \theta|y|/18 \right] &\leq 2\eta, \\ \mathop{\mathbf{Pr}}_{x,y \in D_{4n}} \left[|P(2^{k_x}x + y) - l(2^{k_x}x + y)| > \theta|2^{k_x}x + y|/18 \right] &\leq 2\eta. \end{aligned}$$

Observing that $2^{k_x}|x| + |y| + |2^{k_x}x + y| \leq 18 \times 2^{k_x}|x|$, for all $x, y \in D_{4n}$, one can conclude that $\mathsf{Rej}(P, \theta) \leq 6\eta$. □

Now, the existence of an approximate self-tester for linear functions with linear error directly follows.

Usually one would like a self-tester to be different and simpler than any correct program. For example one can ask the self-tester to satisfy the *little-oh property* [BK95], *i.e.* having a running time asymptotically smaller than any known correct program. This property could be too restrictive for family testing. Here we simplify this condition. If T is a self-tester for linearity (or multi-linearity) over D_n then T is required to use only *simple operations*, that is comparisons, additions, and multiplications by powers of 2. Moreover the number of queries and simple operations of T has to be independent of the size of the domain, that is n. For the sake of simplicity, we will suppose that both n and θ are powers of 2. If they are not, one can replace them by the closest powers of 2, and our results remain valid up to some multiplicative constants.

Corollary 1 *Let $\theta \geq 0$. Then for all $0 < \delta < 1/1024$, there is a $(D_{8n}, \theta|x|/18, \delta/12; D_n, 528\theta|x|, 64\delta)$-approximate self-tester for $\mathcal{L}$ which performs $O(1/\delta)$ calls to the oracle program and uses only $O(1/\delta)$ simple operations.*

4 Testing with relative error

In this section we show how our results lead to approximate self-testers with relative error for both linear functions and multi-linear functions.

4.1 From linear error to relative error

We now undertake the second stage of the construction of the self-tester for the class of linear functions in the case of relative error. Specifically, we modify the **Linear error Linearity Test** to handle relative test errors.

In order to explain this modification, consider a program P that approximately computes a linear function l with relative error θ (that is, with an error in x upper bounded by $\theta \times |l(x)|$). Thus we would like a test error term of order $\theta \times l(n)$ in the **Linear error Linearity Test**. But note that only θ is known. Nevertheless, we will be able to estimate $l(n)$.

One has to take care of the fact that even if P is close to l, the value $P(n)$ might be very far from $l(n)$. Thus, $P(n)$ is not necessarily a good estimation of $l(n)$. We get around this problem by self-correcting P in n. This leads to the **Relative error Linearity Test** described below.

In the same way, we will overcome the shrunk phenomenon that appears in the robustness and the soundness of the **Linear error Linearity Test** by introducing an extension operator. From $P : D_n \to \mathbb{R}$ and $G \in \mathbb{R}$, we define (over $\mathbb{Z}$) the real-valued function $ext(P, G)$ by:

$$ext(P,G)(x) = \begin{cases} P(x), & \text{if } x \in D_n, \\ ext(P,G)(x-n) + G, & \text{if } x > n, \\ ext(P,G)(x+n) - G, & \text{if } x < -n. \end{cases}$$

We will use this extension of P by letting G be the self-corrected value of P in n. Then, the modified **Linear error Linearity Test** becomes:

Relative error Linearity Test(P, θ)
1. Randomly choose $y \in \{0, \ldots, n\}$.
2. Compute $G_y = P(n-y) + P(y)$.
3. Compute $\tilde{\theta}_y = \theta |G_y|/n$.
4. Call **Linear error Linearity Test($ext(P, G_y), \tilde{\theta}_y$)**.

We henceforth denote by $\mathsf{Rej}^r(P, \theta)$ the rejection probability of P by the **Relative error Linearity Test**, and let $\theta\text{-}\,\mathsf{Dist}^r_D$ denote $(\theta\varepsilon)\text{-}\,\mathsf{Dist}_D$ where $\varepsilon(x, v) = |v|$. The following results establish both the soundness (Lemma 5) and the robustness (Theorem 5) of the **Relative error Linearity Test**.

Lemma 5 *Let $0 \le \theta \le 18$, $\mathcal{L}$ be the set of linear functions over $\mathbb{Z}$, and $P : D_n \to \mathbb{R}$. Then,*

$$\mathsf{Rej}^r(P,\theta) \le 10 \times (\theta/72)\text{-}\,\mathsf{Dist}^r_{D_n}(P, \mathcal{L}).$$

Proof Let $l : D_n \to \mathbb{R}$ be a linear function such that $(\theta/72)\text{-}\,\mathsf{Dist}^r_{D_n}(P, l) = \eta$. For $y \in \{0, \ldots, n\}$, let $G_y = P(n-y) + P(y)$, $\tilde{\theta}_y = \theta|G_y|/n$, and $\tilde{P}_y = ext(P, G_y)$. By the halving principle, $|G_y - l(n)| \le \theta|l(n)|/36$ with probability greater than $1 - 4\eta$, when y is randomly chosen in $\{0, \ldots, n\}$. If this latter inequality is satisfied, then $(\theta/36)\text{-}\,\mathsf{Dist}^r_{D_{8n}}(\tilde{P}_y, l) \le \eta$. Since $\theta/36 \le 1/2$, the assumed inequality also implies that $|l(n)| \le 2|G_y|$. Therefore, it follows that $(\theta|x|/18)\text{-}\,\mathsf{Dist}_{D_{8n}}(\tilde{P}_y, l) \le \eta$. Lemma 4 implies that the rejection probability of the **Linear error Linearity Test($ext(P, G_y), \tilde{\theta}_y$)** is at most 6η. It immediately follows that $\mathsf{Rej}^r(P, \theta) \le (6 + 4)\eta = 10\eta$. $\square$

Theorem 5 *Let $\theta \ge 0$, $0 \le \delta < 1/512$, and $P : D_n \to \mathbb{R}$. Then,*

$$\mathsf{Rej}^r(P,\theta) \le \delta \implies (137\theta)\text{-}\,\mathsf{Dist}^r_{D_n}(P, \mathcal{L}) \le 32\delta.$$

Proof Assume $\mathsf{Rej}^r(P, \theta) \le \delta$. Then, there exists $y \in D_n$ such that for $G_y = P(n-y)+P(y)$ and $\tilde{\theta}_y = \theta|G_y|/n$, the rejection probability of **Linear error Linearity Test($ext(P, G_y), \tilde{\theta}_y$)** is at most δ. Thus, by Theorem 2, the linear function $l : D_n \to \mathbb{R}$ defined by

$$l(n) = \operatorname*{Med}_{y \in D_{2n}: y \ge 0} \left(P(n+y) - P(y)\right),$$

is such that $(137\tilde{\theta}_y|x|)\text{-}\,\mathsf{Dist}_{D_n}(P,l) \leq 32\delta$. The equality $l(n) = G_y$ must hold, and therefore

$$(137\theta)\text{-}\,\mathsf{Dist}^r_{D_n}(P,l) = (137\tilde{\theta}_y|x|)\text{-}\,\mathsf{Dist}_{D_n}(P,l) \leq 32\delta.$$

□

The existence of an approximate self-tester for linear functions with relative error now follows. Note that we do not need anymore to test the program on a larger domain since the soundness and the robustness of the test were stated on the same domain.

Corollary 2 *Let $0 \leq \theta \leq 18$, and let ε be the computational error term such that $\varepsilon(x,v) = |v|$. Then for all $0 < \delta < 1/1024$, there is a $(D_n, \theta\varepsilon/72, \delta/20; D_n, 137\theta\varepsilon, 64\delta)$-approximate self-tester for $\mathcal{L}$ which performs $O(1/\delta)$ calls to the oracle program and uses only $O(1/\delta)$ simple operations.*

4.2 Testing multi-linearity

Using our results, we construct an approximate self-tester for multi-linear functions with relative error. To state the result we define for every $\vec{z} \in (D_n)^d$, and $i = 1, \ldots, d$, the function $\tilde{P}^i_{\vec{z}} : D_n \to \mathbb{R}$ which at t takes the value $P(z_1, \ldots, z_{i-1}, t, z_{i+1}, \ldots, z_d)$. Let d be a positive integer, then the test for d-linearity is:

Relative error d-Linearity Test(P, θ)
1. Randomly choose $\vec{z} \in (D_n)^d$.
2. Randomly choose $i \in \{1, \ldots, d\}$.
3. Do **Relative error Linearity Test$(\tilde{P}^i_{\vec{z}}, \theta)$**.

The soundness and the robustness of the test are directly derived from those of **Relative error Linearity Test** using some techniques from [FHS94], where a similar result for multi-variate polynomials in the context of exact computation is proven. Fact 1 and Lemma 6 lower and upper bound the distance between a d-variate function and d-linear functions by its successive distances from functions which are linear in only one of their variables.

Let $\mathcal{L}^d$ denote the set of d-linear functions defined over $(D_n)^d$, and $\mathcal{L}^d_i$ the set of functions defined over $(D_n)^d$ which are linear in their ith variable. First let us state the easy bound that will be used for the soundness.

Fact 1 *Let $\theta \geq 0$. Then for all $f : (D_n)^d \to \mathbb{R}$,*

$$\frac{1}{d}\sum_{i=1}^{d} \theta\text{-}\,\mathsf{Dist}^r_{(D_n)^d}(f, \mathcal{L}^d_i) \leq \theta\text{-}\,\mathsf{Dist}^r_{(D_n)^d}(f, \mathcal{L}^d).$$

The following more difficult bound will be used for the robustness.

Lemma 6 *Let $0 \leq \theta \leq 1/(16d^2)$. Then for all $f : (D_n)^d \to \mathbb{R}$,*

$$(4d\theta)\text{-}\,\mathsf{Dist}^r_{(D_n)^d}(f, \mathcal{L}^d) \leq 2\sum_{i=1}^{d} \theta\text{-}\,\mathsf{Dist}^r_{(D_n)^d}(f, \mathcal{L}^d_i).$$

Proof We proceed by induction on d. The case $d = 1$ is clear. Let $d \geq 2$, and suppose that the property holds for $d - 1$. By a slight abuse of notation, the set $(D_n)^d$ will now be omitted in distance notations. Let L^{d-1} denote the set of functions defined over $(D_n)^d$ which are linear in their first $(d-1)$ variables. We will prove the following inequality which implies the induction step:

$$(4d\theta)\text{-}\,\mathsf{Dist}^r(f, \mathcal{L}^d) \leq (4(d-1)\theta)\text{-}\,\mathsf{Dist}^r(f, L^{d-1}) + 2 \times (\theta\text{-}\,\mathsf{Dist}^r(f, \mathcal{L}^d_d)). \tag{2}$$

Denote by η_1 (respectively η_2) the first (respectively second) distance on the right hand side of (2). Let $l_1 \in L^{d-1}$ and $l_2 \in \mathcal{L}_d^d$ be such that $(4(d-1)\theta)$- $\mathsf{Dist}^r(f, l_1) = \eta_1$, and θ- $\mathsf{Dist}^r(f, l_2) = \eta_2$. Then there exists $b \in D_n$, such that

$$\mathop{\mathbf{Pr}}_{\vec{z}\in(D_n)^d : z_d = b}\left[\begin{array}{c} |l_2(\vec{z}) - f(\vec{z})| > \theta|l_2(\vec{z})| \\ \text{or } |f(\vec{z}) - l_1(\vec{z})| > (4d-4)\theta|l_1(\vec{z})| \end{array}\right] \leq \eta_1 + \eta_2. \quad (3)$$

Let $l \in \mathcal{L}^d$ denote the d-linear function which satisfies $l(\cdot,\ldots,\cdot,b) = l_1(\cdot,\ldots,\cdot,b)$. We will prove that f is close to l in the sense of (2).

First, since $0 \leq \theta \leq 1/(16d^2)$, we have

$$(4d\theta)\text{- }\mathsf{Dist}^r(f,l) \leq \mathop{\mathbf{Pr}}_{\vec{z}\in(D_n)^d}\left[\begin{array}{c} |f(\vec{z}) - l_2(\vec{z})| > \theta|l_2(\vec{z})| \\ \text{or } |l_2(\vec{z}) - l(\vec{z})| > (4d-2)\theta|l_2(\vec{z})| \\ \text{or } |l_2(\vec{z})| > (1+4d\theta)|l(\vec{z})| \end{array}\right]. \quad (4)$$

Since l and l_2 are linear in their dth variable, we can fix $z_d = b$ in the last two inequalities without changing the probability. Then using the closeness of l_2 to f, we rewrite (4) using the union bound:

$$(4d\theta)\text{- }\mathsf{Dist}^r(f,l) \leq \eta_2 + \mathop{\mathbf{Pr}}_{\vec{z}\in(D_n)^d : z_d = b}\left[\begin{array}{c} |l_2(\vec{z}) - l(\vec{z})| > (4d-2)\theta|l_2(\vec{z})| \\ \text{or } |l_2(\vec{z})| > (1+4d\theta)|l(\vec{z})| \end{array}\right].$$

We now rewrite the first inequality inside the probability term using the argument that gave (4). Since $0 \leq \theta \leq 1/(16d^2)$ and $l(\cdot,\ldots,\cdot,b) = l_1(\cdot,\ldots,\cdot,b)$, we get that

$$\begin{aligned} &(4d\theta)\text{- }\mathsf{Dist}^r(f,l) \\ &\leq \eta_2 + \mathop{\mathbf{Pr}}_{\vec{z}\in(D_n)^d : z_d = b}\left[\begin{array}{ll} \quad |l_2(\vec{z}) - f(\vec{z})| > \theta|l_2(\vec{z})| & \text{(a)} \\ \text{or } |f(\vec{z}) - l_1(\vec{z})| > (4d-4)\theta|l_1(\vec{z})| & \text{(b)} \\ \text{or } |l_2(\vec{z})| > (1+4d\theta)|l_1(\vec{z})| & \text{(c)} \\ \text{or } |l_1(\vec{z})| > (1+4d\theta)|l_2(\vec{z})| & \text{(d)} \end{array}\right]. \end{aligned}$$

Using again that $0 \leq \theta \leq 1/(16d^2)$, it appears that conditions (c) and (d) can not be satisfied unless conditions (a) and (b) are also true. Thus the probability term does not depend on conditions (c) and (d). Therefore inequality (3) concludes the proof:

$$\begin{aligned} (4d\theta)\text{- }\mathsf{Dist}^r(f,l) &\leq \eta_2 + \mathop{\mathbf{Pr}}_{\vec{z}\in(D_n)^d : z_d = b}\left[\begin{array}{c} |l_2(\vec{z}) - f(\vec{z})| > \theta|l_2(\vec{z})| \\ \text{or } |f(\vec{z}) - l_1(\vec{z})| > (4d-4)\theta|l_1(\vec{z})| \end{array}\right] \\ &\leq 2\eta_2 + \eta_1. \end{aligned}$$

□

We now state our final result which extends Corollary 2 to multi-linear functions. Quite surprisingly it does not use any multiplication but only few simple operations.

Corollary 3 *Let $d \geq 1$ be an integer, and let $0 \leq \theta \leq O(1/d^2)$. Let ε be the computational error term such that $\varepsilon(x,v) = |v|$. Then for all $0 < \delta < O(1)$, there is a $((D_n)^d, \theta\varepsilon, \delta; (D_n)^d, O(d)\theta\varepsilon, O(d)\delta)$-approximate self-tester for $\mathcal{L}^d$ which performs $O(1/\delta)$ calls to the oracle program and uses only $O(1/\delta)$ simple operations.*

Open questions

In this paper we have achieved the goal of approximate self-testing with relative error for multi-linear functions. It would be interesting to extend this work to polynomials. More generally, when one does not have *a priori* information on the size of the function to be computed, constructing approximate self-testers with relative error is a challenging problem.

Acknowledgments

We would like to thank Stéphane Boucheron, Julia Kempe, Marcos Kiwi, Sophie Laplante, and Miklos Santha for useful discussions and assistance while writing this paper. We are also grateful to the anonymous referees for their remarks that greatly improved the presentation of the paper.

References

[ABCG93] S. Ar, M. Blum, B. Codenotti, and P. Gemmell. Checking approximate computations over the reals. In *Proceedings of 25th ACM Symposium on Theory of Computing*, pages 786–795, 1993.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractibility of approximation problems. *Journal of the ACM*, 45(3):505–555, 1998.

[AS98] S. Arora and S. Safra. Probabilistic checkable proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[Bab93] L. Babai. Transparent (holographic) proofs. In *Proceedings of 10th Symposium on Theoretical Aspects of Computer Science*, volume 665, pages 525–534. LNCS, 1993.

[BK95] M. Blum and S. Kannan. Designing programs that check their work. *Journal of the ACM*, 42(1):269–291, 1995.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[Blu88] M. Blum. Designing programs to check their work. Technical Report 88-009, ICSI, 1988.

[Cop89] D. Coppersmith, December 1989. see discussion in [BLR93].

[EKR01] F. Ergün, S. Ravi Kumar, and R. Rubinfeld. Approximate checking of polynomials and functional equations. *SIAM Journal on Computing*, 31(2):550–576, 2001.

[FHS94] K. Friedl, Z. Hátsági, and A. Shen. Low-degree tests. *Proceedings of 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 57–64, 1994.

[For95] G. L. Forti. Hyers-Ulam stability of functional equations in several variables. *Aequationes Mathematicae*, 50:143–190, 1995.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[GLR+91] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of 23rd ACM Symposium on Theory of Computing*, pages 32–42, 1991.

[GR99] O. Goldreich and D. Ron. A sublinear bipartite tester for bounded degree graphs. *Combinatorica*, 19:335–373, 1999.

[HR92] D. H. Hyers and T. M. Rassias. Approximate homomorphisms. *Aequationes Mathematicae*, 44:125–153, 1992.

[HS92] D. H. Hyers and P. Semrl. On the behaviour of mappings which do not satisfy Hyers-Ulam stability. *Proceedings of the American Mathematical Society*, 144(4):989–993, April 1992.

[Hye41] D. H. Hyers. On the stability of the linear functional equation. *Proceedings of the National Academy of Sciences of the USA*, 27:222–224, 1941.

[KMS00] M. Kiwi, F. Magniez, and M. Santha. Exact and approximate testing/correcting of algebraic functions: A survey. In *Proceedings of 1st Summer School on Theoretical Aspects of Computer Science*, volume 2292, pages 30–83. LNCS, 2000. Also ECCC Report TR01-014.

[KMS03] M. Kiwi, F. Magniez, and M. Santha. Approximate testing with error relative to input size. *Journal of Computer and System Sciences*, 66(2):371–392, 2003.

[Lip91] R. Lipton. New directions in testing. *Series in Discrete Mathematics and Theoretical Computer Science*, 2:191–202, 1991.

[RS92] T. M. Rassias and P. Semrl. On the behavior of mappings that do not satisfy Hyers-Ulam stability. *Proceedings of the AMS*, 114(4):989–993, 1992.

[RS96] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):23–32, April 1996.

[Rub99] R. Rubinfeld. On the robustness of functional equations. *SIAM Journal on Computing*, 28(6):1972–1997, 1999.

[Sko83] F. Skof. Sull'approssimazione delle applicazioni localmente δ-additive. *Atti della Accademia delle Scienze di Torino*, 117:377–389, 1983. in Italian.