Testing for numerical computations

M. Karpovsky, Mem.I.E.E.E.

Indexing terms: *Digital computers and computation, Errors and error analysis*

Abstract: We consider the problem of error detection in programs or specialised devices computing real functions $f(x)$, where the argument x is represented in binary form. For error detection we use the linear check inequalities

$$\left|\sum_{\tau \in T} f(x \oplus \tau) - C\right| < \epsilon,$$

where $\epsilon > 0$ is some given small constant, $\oplus$ denotes componentwise addition mod 2 of binary vectors, T is some set of binary vectors and C is a constant. A method for the construction of a minimal check set T and constant C for the given $f(x)$ and ϵ is proposed. This method is based on the techniques of Walsh transforms and least-absolute-error polynomial approximation. Several important examples of optimal checks for programs computing exponential, logarithmic and trigonometric functions will be given.

1 Introduction

We shall consider the problem of error detection for programs or devices computing real functions $f(x)$, where x is represented in binary form. By errors we mean errors in the text of programs or the catastrophic structure failures in digital devices.

Let $x = (x_1, \ldots, x_n) \in G, x_i \in \{0, 1\}$ and G be the group of binary n-vectors with respect to the operation $\oplus$ of componentwise addition mod 2. In References 1 to 4 the methods of error detection based on linear checks

$$\sum_{\tau \in T} f(x \oplus \tau) - C = 0 \tag{1}$$

were considered. In eqn. 1, T is a 'check' subgroup of G and C is a constant. (We use the same letter to denote the integer from $\{0, 1, \ldots, 2^n - 1\}$ and its binary representation). The verification of whether eqn. 1 is satisfied constitutes the error-detection method. In Reference 1 the method of constructing optimal checks so as to minimise the cardinality of a check set T was proposed and generalised to the case where G is an arbitrary commutative group. This method was based on the very powerful technique of Fourier transforms over the finite groups and the corresponding fast Fourier transforms.[5, 6, 7, 8] Advantages and limitations of the error-detection method based on linear checks (eqn. 1) and their error-detecting capability were also considered in Reference 1.

In References 2 and 3 this method was generalised to the case where G is an arbitrary finite group, and summation of $f(x \oplus \tau)$ in eqn. 1 is carried out in an arbitrary (possibly finite) field. Very simple checks (such as that of eqn. 1) for such standard computer blocks as counters, adders, subtractors, multipliers etc., are also given in Reference 2.

We note also that a similar technique was used for the problems of logical design[12, 13, 14, 15, 16] for the design of linear systems over the groups.[17, 18]

In a program or a device for computing $f(x)$ the problem of error correction by a system of linear checks was considered in References 2 and 3. Decoding methods for the results of these checks (syndromes), complexity of decoding and error-detecting and error-correcting capabilities were also considered in those References. It was shown[3] that systems of linear checks of the type of eqn. 1 generate good error-correcting codes. Methods of error detection based on these checks for programs computing polynomials of several arguments were considered,[4] and we note that they may be effectively used in the case where $f(x)$ is an integer for every $x \in \{0, 1, \ldots, 2^n - 1\}$, and very few noninteger functions have nontrivial checks.

In this paper we shall generalise linear check methods to the case of noninteger computations. These generalised checks will be constructed for such important noninteger computations as exponential, logarithmic and trigometric computations (see Table 1, Section 4). For error detection in noninteger computations we shall use linear *inequality* checks.

The method described in this paper may be effectively used for the testing of manufacturing acceptance of the program or of the device computing the given numerical function $f(x)$. In the case of hardware implementation, this method may be used for maintenance testing of the corresponding devices.

2 Complexity of check sets

Let $f(x)$ be a real number for every $x \in \{0, 1, \ldots, 2^n - 1\}$, $x = (x_1, \ldots, x_n)$ and $x_i \in \{0, 1\}$ and let $\epsilon \geqslant 0$ be some small constant. For error detection we shall use linear inequality checks

$$\left|\sum_{\tau \in T} f(x \oplus \tau) - C\right| \leqslant \epsilon \tag{2}$$

where C is some constant and T is a subgroup of the group G of binary n-vectors. (Check eqn. 1 is a special case of expr. 2 with $\epsilon = 0$).

We shall discuss in this Section the cardinality $|T(f, \epsilon)|$ of a minimal check set $T = T(f, \epsilon)$ for the given function f and $\epsilon \geqslant 0$. These results will be the generalisation of the corresponding results from Reference 4, where the case $\epsilon = 0$ was considered.

Paper 595E, first received 2nd July 1979 and in revised form 8th January 1980

Dr. Karpovsky is with the Computer Science Department, School of Advanced Technology, State University of New York at Binghamton, Binghamton, New York 13901, USA

2.1 Linear transform of arguments

Let σ be an $(n \times n)$-binary nonsingular over $GF(2)$ matrix, $y = (y_1, \ldots, y_n)$ be some binary vector and

$$\phi(x) = f(\sigma x \oplus y) \text{ for every } x = (x_1, \ldots, x_n) \tag{3}$$

Then, by definition of $T(f, \epsilon)$, there exists C such that

$$\left| \sum_{\tau \in T(f, \epsilon)} f(x \oplus \tau) - C \right| \leqslant \epsilon$$

for every x, and we have from eqn. 3 that

$$\left| \sum_{\tau \in \sigma^{-1} T(f, \epsilon)} \phi(x \oplus \tau) - C \right| = \left| \sum_{\tau \in T(f, \epsilon)} \phi(x \oplus \sigma^{-1}\tau) - C \right|$$

$$= \left| \sum_{\tau \in T(f, \epsilon)} f(\sigma x \oplus y \oplus \tau) - C \right| \leqslant \epsilon$$

where σ^{-1} is the inverse of σ over $GF(2)$ and $\sigma^{-1}T(f, \epsilon) = \{\sigma^{-1}\tau | \tau \in T(f, \epsilon)\}$ and $\sigma^{-1}T(f, \epsilon)$ is a check set for $\phi(x)$; $\sigma^{-1}T(f, \epsilon)$ is a minimal check set for $\phi(x)$ since $|\sigma^{-1}T(f, \epsilon)| = |T(f, \epsilon)|$. Thus, we have, for any $\phi(x)$ defined by eqn. 3

$$T(\phi, \epsilon) = \sigma^{-1} T(f, \epsilon) \tag{4}$$

2.2 Linear transform of functions

Let $f_1, \ldots, f_r$ be some real functions, $\epsilon_1 \geqslant 0, \ldots, \epsilon_r \geqslant 0$ be some small constants

$$\left| \sum_{\tau \in T(f_i, \epsilon_i)} f_i(x \oplus \tau) - C \right| \leqslant \epsilon_i \quad i = 1, \ldots, r \tag{5}$$

and

$$\phi(x) = \sum_{i=1}^{n} \alpha_i f_i(x) \tag{6}$$

Denote

$$T = \overset{r}{\underset{i=1}{\oplus}} T(f_i, \epsilon_i)$$
$$= \{\tau_1 \oplus \ldots \oplus \tau_r | \tau_1 \in T(f_1, \epsilon_1), \ldots, \tau_r \in T(f_r, \epsilon_r)\} \tag{7}$$

Since $T(f_i, \epsilon_i)$ is, by definition, a subgroup of the group G of binary n-vectors,

$$T = \overset{r}{\underset{i=1}{\oplus}} T(f_i, \epsilon_i)$$

is also a subgroup of G. Denote by T_i a subgroup isomorphic to the factor group $T/T(f_i, \epsilon_i)$. Then, we have from eqns. 5, 6 and 7

$$\left| \sum_{\tau \in T} \phi(x \oplus \tau) - \sum_{i=1}^{r} \alpha_i C_i \right|$$

$$= \left| \sum_{i=1}^{r} \alpha_i \sum_{\tau \in T_i} \sum_{\tau_i \in T(f_i, \epsilon_i)} f_i(x \oplus \tau \oplus \tau_i) - \sum_{i=1}^{r} \alpha_i C_i \right|$$

$$= \sum_{i=1}^{r} \alpha_i \sum_{\tau \in T_i} \left(\sum_{\tau_i \in T(f_i, \epsilon_i)} f(x \oplus \tau \oplus \tau_i) - C_i \right) \Bigg|$$

$$\leqslant \sum_{i=1}^{n} \epsilon_i |\alpha_i| \, |T_i| \tag{8}$$

It follows from expr. 8 that, for the function $\phi(x)$ defined by eqn. 6

$$T = \overset{r}{\underset{i=1}{\oplus}} T(f_i, \epsilon_i)$$

is a check set and

$$\left| T\left(\phi, \sum_{i=1}^{r} \epsilon_i |\alpha_i| \, |T_i|\right) \right| \leqslant \left| \overset{r}{\underset{i=1}{\oplus}} T(f_i, \epsilon_i) \right.$$

$$\leqslant \prod_{i=1}^{r} T(f_i, \epsilon_i) \tag{9}$$

2.3 Convolution of functions

Denote

$$\phi(x) = \sum_{Z \in G} f_1(Z) f_2(x \oplus Z) \tag{10}$$

If

$$\left| \sum_{\tau \in T(f_i, \epsilon_i)} f_i(x \oplus \tau) - C_i \right| \leqslant \epsilon_i \qquad i = 1, 2$$

then we have from eqn. 10

$$\left| \sum_{\tau \in T(f_2, \epsilon_2)} \phi(x \oplus \tau) - C_2 \sum_{Z \in G} f_1(Z) \right|$$

$$= \left| \sum_{\tau \in T(f_2, \epsilon_2)} \sum_{Z \in G} f_1(Z) f_2(x \oplus \tau \oplus Z) - C_2 \sum_{Z \in G} f_1(Z) \right|$$

$$= \left| \sum_{Z \in G} f_1(Z) \left(\sum_{\tau \in T(f_2, \epsilon_2)} f_2(x \oplus Z \oplus \tau) - C_2 \right) \right|$$

$$\leqslant \epsilon_2 \sum_{Z \in G} |f_1(Z)| \tag{11}$$

It follows from expr. 11 that $T(f_2, \epsilon_2)$ is a check set for $\phi(x)$ defined by eqn. 10, and

$$\left| T\left(\phi, \epsilon_2 \sum_{Z \in G} |f_1(Z)|\right) \right| \leqslant |T(f_2, \epsilon_2)| \tag{12}$$

By a similar proof we can show also that

$$\left| T\left(\phi, \epsilon_1 \sum_{Z \in G} |f_2(Z)|\right) \right| \leqslant |T(f_1, \epsilon_1)| \tag{13}$$

2.4 Superposition of functions

Let $x = (x_1, \ldots, x_n)$ and $f(x) = (f_1(x), \ldots, f_n(x))$ be a system of n Boolean functions of n arguments.

We shall say that $\tau = (\tau_1, \ldots, \tau_n)$ is a self duality point for $f_i(x)$ if $f_i(x \oplus \tau) = 1 \oplus f_i(x)$.[5] We shall denote by D_i the set of all self duality points for $f_i (i = 1, \ldots, n)$. (If $(1, \ldots, 1) \in D_i$, then f_i is self dual).

Suppose

$$\psi(x) = \phi(f(x)) \tag{14}$$

and for $\phi(x)$ we have for every x

$$|\phi(x)+\phi(x\oplus(1,\ldots,1))-C| \leqslant \epsilon \tag{15}$$

where C and ϵ are some constants. Then we have, for every $\tau\in\bigcap_{i=1}^{n} D_i$

$$|\psi(x)+\psi(x\oplus\tau)-C| = |\phi(f(x))+\phi(f(x\oplus\tau))-C|$$

$$= |\phi(f(x))+(\phi(f(x)\oplus(1,\ldots,1)))-C| \leqslant \epsilon \tag{16}$$

It follows now from expr. 16 that, for $\psi(x)$ and every $\tau\in\bigcap_{i=1}^{n} D_i$

$$T(\psi,\epsilon) = \{(0,\ldots,0),\tau\} \tag{17}$$

We note also that, for any function $\phi(x)$, expr. 15 is satisfied for every $\epsilon \geqslant 0{\cdot}5\,(\max_x\,(\phi(x)+\phi(x\oplus(1,\ldots,1))) - \min_x\,(\phi(x)+\phi(x\oplus(1,\ldots 1))))$ if we choose $C=0{\cdot}5\,(\max_x\,(\phi(x)+\phi(x\oplus(1,\ldots,1))) + \min_x\,(\phi(x)+\phi(x\oplus(1,\ldots,1))))$.

2.4.1 Example 1: Let $n=3$, $f_1(x_1, x_2, x_3)=x_1$, $f_2(x_1, x_2, x_3) = \text{Maj}\,(x_1, x_2, x_3) = \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3$, where $\vee$ denotes logical addition, $f_3(x_1,x_2,x_3) = \text{EXOR}\,(x_1,x_2,x_3) = x_1\oplus x_2\oplus x_3$, and

$$\phi(x) = \left(\frac{7-x}{2}\right)^3$$

where

$$x = \sum_{i=1}^{3} x_i 2^{i-1}.$$

Then f_1, f_2, f_3 are self dual ($f_i(x\oplus(1,1,1)) = 1\oplus f_i(x)$), expr. 15 is satisfied for $C=\epsilon=0$ and we have from eqns. 16 and 17 for $\psi(x)=\phi(f(x))=2^{-3}(7-(x_1+2\,\text{Maj}\,(x_1,x_2,x_3)+4\,\text{EXOR}\,(x_1,x_2,x_3)))^3 : \psi(x)+\psi(x\oplus(1,\ldots,1))=0$ for every $x=(x_1,x_2,x_3)$.

2.5 Check complexities for positive (negative) functions

Let $f(x)$ be a positive function ($f(x)\geqslant 0$ for every $x\in\{0,\ldots,2^n-1\}$). By the definition of $T(f,\epsilon)$ there exists a constant C such that

$$\left|\sum_{\tau\in T(f,\epsilon)} f(x\oplus\tau) - C\right| \leqslant \epsilon \tag{18}$$

or

$$C-\epsilon \leqslant \sum_{\tau\in T(f,\epsilon)} f(x\oplus\tau) \leqslant C+\epsilon$$

for every $x\in\{0,1,\ldots,2^n-1\}$

Then we have from expr. 18

$$(C-\epsilon)2^n|T(f,\epsilon)|^{-1} \leqslant \sum_{Y\in G} f(Y)$$

$$= \sum_{x\in G/T(f,\epsilon)}\ \sum_{\tau\in T(f,\epsilon)} f(x\oplus\tau)$$

$$\leqslant (C+\epsilon)\ 2^n|T(f,\epsilon)|^{-1} \tag{19}$$

where $G/T(f,\epsilon)$ is a subgroup isomorphic to a factor group of G with respect to $T(f,\epsilon)$ and $|G/T(f,\epsilon)| = 2^n|T(f,\epsilon)|^{-1}$. Since $\log_2|T(f,\epsilon)|$ is an integer, we have from expr. 19 the following lower and upper bounds for positive functions:

$$n+\,]\log_2(C-\epsilon)-\log_2\sum_{Y\in G} f(Y)[$$

$$\leqslant \log_2|T(f,\epsilon)| \leqslant n+[\log_2(C+\epsilon)-\log_2\sum_{Y\in G} f(Y)] \tag{20}$$

where $]\alpha[$ ($[\alpha]$) is a smallest (greatest) integer $\geqslant \alpha$ ($\geqslant \alpha$).

For positive functions $C\geqslant \min_{[x|f(x)\neq 0]} f(x)$, and we have from expr. 20

$$\log_2|T(f,\epsilon)| \geqslant n$$

$$+\,]\log_2(\min_{[x|f(x)\neq 0]} f(x)-\epsilon)-\log_2\sum_{Y\in G} f(Y)[\tag{21}$$

The bounds similar to exprs. 20 and 21 may be obtained also for negative functions ($f(x)\leqslant 0$ for every x).

We note also that the bounds in exprs. 20 and 21 are exact, and there exist positive functions such that these bounds are reached (see example 2, following).

2.5.1 Example 2: Let

$$x = \sum_{i=1}^{n} x_i 2^{i-1}(x_i\in\{0,1\}),$$

$$N = 2^n,$$

$$\epsilon = 0{\cdot}5(2^{-0{\cdot}5N}-2^{-N+1})$$

and

$$f(x) = x_n(a+2^{-x}) \quad \text{where} \quad a\geqslant 1.$$

Then

$$\min_{[x|f(x)\neq 0]} f(x) = a+2^{-N+1}$$

$$\sum_{Y\in G} f(Y) = \sum_{x=0{\cdot}5N}^{N-1}(a+2^{-x}) = 0{\cdot}5aN+2^{-0{\cdot}5N+1}-2^{-N+1}$$

and by eqn. 21 for every $a\geqslant 1, n>1$

$$\log_2|T(f)| \geqslant n+\,]\log_2(a+2^{-N+1}-0{\cdot}5(2^{-0{\cdot}5N}-2^{-N+1}))$$

$$-\log_2(0{\cdot}5aN+2^{-0{\cdot}5N+1}-2^{-N+1})[\ = 1$$

Choose

$$C = a+0{\cdot}5(2^{-0{\cdot}5N}+2^{-N+1})$$

Then it follows from expr. 20 for every $a\geqslant 1, n>1$

$$n+\,]\log_2(a+2^{-N+1})-\log_2(0{\cdot}5aN+2^{-0{\cdot}5N+1}-2^{-N+1})[$$

$$= n+[\log_2(a+2^{-0{\cdot}5N})$$

$$-\log_2(0{\cdot}5aN+2^{-0{\cdot}5N+1}-2^{-N+1})]$$

$$= \log_2|T(f,\epsilon)| = 1$$

We note that, for $f(x) = x_n(a + 2^{-x}), N = 2^n$

$$|f(x) + f(x \oplus (0, 0, \ldots, 01)) \\ - (a + 0{\cdot}5(2^{-0{\cdot}5N} + 2^{-N+1}))| \\ \leqslant 0{\cdot}5(2^{-0{\cdot}5N} - 2^{-N+1})$$

for every x, and the bounds of exprs. 20 and 21 are reached.

3 Optimal inequality checks and error-correcting codes

Our problem is to construct, for the given function $f(x)$ and given $\epsilon \geqslant 0$, nontrivial inequality checks expr. 2. (For every function $f(x)$ there exists the trivial check with $T = G$, $C = \sum_{x \in G} f(x)$ and $\epsilon = 0$).

Let $P_s(x) = \sum_{i=0} a_i x^i$ be a polynomial of degree s which is the least-absolute-error approximation for $f(x)$ over the set $\{0, 1, \ldots, 2^n - 1\}$, with maximum absolute error:

$$\Delta_s = \max_{x \in [0, \ldots, 2^n - 1]} |f(x) - P_s(x)| \tag{22}$$

Methods for the construction of the polynomial approximation $P_s(x)$ and estimation on Δ_s for the given $f(x)$ are well known (e.g. see References 9 and 10).

Suppose that we have already found a check set T and constant C such that $P_s(x)$ satisfies eqn. 1. Then we have, for $f(x) = P_s(x) + \Delta_s(x)(|\Delta_s(x)| \leqslant \Delta_s)$ for every $x \in \{0, \ldots, 2^n - 1\}$

$$\left|\sum_{\tau \in T} f(x \oplus \tau) - C\right|$$

$$= \sum_{\tau \in T} P_s(x \oplus \tau) - C + \sum_{\tau \in T} \Delta_s(x \oplus \tau)$$

$$= \sum_{\tau \in T} \Delta_s(x \oplus \tau) \leqslant \Delta_s |T| \tag{23}$$

where $|T|$ is the cardinality of T.

Thus, it follows from expr. 23 that the check set T and constant C satisfy expr. 2, if T and C satisfy eqn. 1 for the polynomial approximation $P_s(x)$, and

$$\Delta_s |T| \leqslant \epsilon \tag{24}$$

For the construction of the check set T and constant C satisfying eqn. 1 for $P_s(x)$ we may use the results from References 2 and 4. Let $V(n, d)$ be a maximal binary linear error-correcting code with code words of length n and distance d,[11] and let $V^{\perp}(n, d)$ be a dual code to $V(n, d)$. Then

$$V^{\perp}(n, d) = \{\tau = (\tau_1, \ldots, \tau_n) \in G \bigoplus_{i=1}^{n} \tau_i x_i = 0$$

for every

$$x = (x_1, \ldots, x_n) \in V(n, d)\}$$

Methods for constructing $V(n, d)$, $V^{\perp}(n, d)$ and estimating their cardinalities may be found, e.g. in Reference 11. It was shown in References 2 and 4 that, if

$$P_s(x) = \sum_{i=0}^{s} a_i x^i, \quad a_s \neq 0$$

and

$$x \in \{0, 1, \ldots, 2^n - 1\},$$

then

$$\sum_{\tau \in V^{\perp}(n, s+1)} P_s(x \oplus \tau) - C = 0 \tag{25}$$

where

$$C = |V(n, s+1)|^{-1} \sum_{x \in G} P_s(x)$$

$$= |V(n, s+1)|^{-1} \sum_{i=0}^{s} a_i (i+1)^{-1} \\ \times \sum_{\nu=0}^{i} \binom{i+1}{\nu} 2^{(i+1-\nu)n} B_\nu$$

where B_ν stands for Bernoulli numbers.

Thus, we have from exprs. 23 and 25

$$\left|\sum_{\tau \in V^{\perp}(n, s+1)} f(x \oplus \tau) - C\right| \leqslant \Delta_s |V^{\perp}(n, s+1)| \tag{26}$$

and the dual code $V^{\perp}(n, s+1)$ is the check set for $f(x)$ if $\Delta_s |V^{\perp}(n, s+1)| \leqslant \epsilon$.

3.1 Example 3

Let

$$n = 24, \quad x \in \{0, 1, \ldots, 2^{24} - 1\},$$

$$x = (x_1, \ldots, x_{24}),$$

then

$$(x_i \in (0, 1)), \quad f(x) = \exp(-(\log_2 e) 2^{-24} x)$$

where e is the base of the natural logarithms, and

$$(0{\cdot}5 \leqslant f(x) < 1 \quad \text{for every} \quad x \in \{0, 1, \ldots, 2^{24} - 1\})$$

and $\epsilon = 10^{-6}$

The function $f(x)$ may be approximated by the polynomial $P_7(x)$ of degree 7 with maximum absolute error $\Delta_7 = 2 \times 10^{-10}$ (Reference 9). Choose the (24, 12)-Golay code $V_G(24, 8)$ with distance 8 as $V(n, s+1) = V(24, 8)$.[11] Then

$$|V(24, 8)| = |V_G^{\perp}(24, 8)| = 2^{12},$$

$$\Delta_7 |V^{\perp}(24, 8)| \leqslant 2 \times 10^{-10} \times 2^{12} < 10^{-6}$$

and

$$T(f, 10^{-6}) = V_G(24, 8).$$

We note also that, as shown in Reference 4, the complexity of the network implementation of the check of eqn. 25 for $P_s(x)$ (minimal number of two-input gates in the corresponding logical network) for $n \to \infty$ is, at most, $[s/2]n$, where $[a]$ is the greatest integer $\leqslant a$.

4 Error detection in computation of analytical functions

We note that, for the great variety of analytical functions,

$\Delta_s|V^{\perp}(n, s+1)|$ decreases very rapidly with the increase of the degree s $(s < n)$ of an approximating polynomial.

Denoting $y = 2^{-n}x (0 \leqslant y < 1)$, an example of the behaviour of $\Delta_s|V^{\perp}(n, s+1)|$ is given by Fig. 1 for $f(y) = 10^{0\cdot 25y}$. The maximum absolute errors Δ_s for this example are taken from Reference 9.

Using the Varshamov bound[11] for $|T| = |V^{\perp}(n, s+1)|$ we have, from expr. 26, sufficient condition for the minimal degree s of an approximating polynomial $P_s(x)$:

$$\Delta_s \sum_{j=0}^{s-1} \binom{n-1}{j} \leqslant \epsilon \tag{27}$$

For estimating Δ_s we may use Taylor's expansion for $f(y)$ $(y = 2^{-n}x)$:

$$f(y) = \sum_{i=0}^{s} (i!)^{-1} f^{(s)}(1/2)(y-1/2)^i$$
$$+ ((s+1)!)^{-1} f^{(s+1)}(\theta(y))(y-\tfrac{1}{2})^{s+1} \tag{28}$$

where $f^{(s+1)}(y)$ is the $(s+1)$th derivative of $f(y)$ and $0 \leqslant \theta(y) \leqslant 1$. Then we have, from eqn. 28

$$\Delta_s = \max_{y \in [0,1]} |\Delta(y)|$$
$$\leqslant \max_{y \in [0,1]} |((s+1)!)^{-1} f^{(s+1)}(\theta(y))(y-1/2)^{s+1}|$$
$$\leqslant ((s+1)!)^{-1} 2^{-(s+1)} \max_{y \in [0,1]} |f^{(s+1)}(y)| \tag{29}$$

Thus, from exprs. 27 and 29 if

$$((s+1)!)^{-1} 2^{-(s+1)} \Big(\max_{y \in [0,1]} |f^{(s+1)}(y)| \Big) \sum_{j=0}^{s-1} \binom{n-1}{j} \leqslant \epsilon \tag{30}$$

then there exists C such that

$$\left| \sum_{\tau \in V^{\perp}(n, s+1)} f(y \oplus \tau) - C \right| \leqslant \epsilon$$

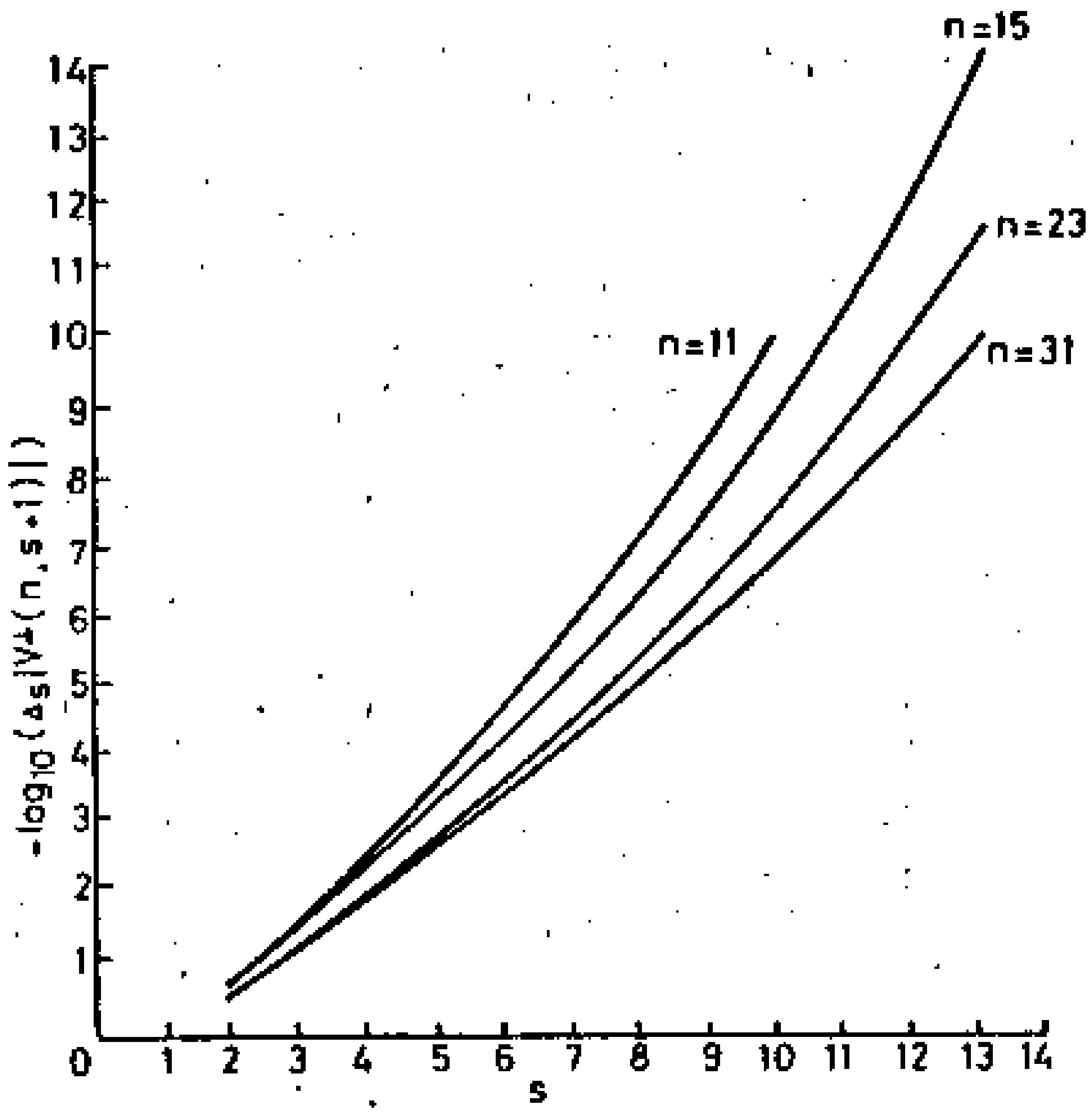

Fig. 1 $f(y) = 10^{0\cdot 25y}$

Expr. 30 provides us with a good upper bound for the cardinality $|T(f, \epsilon)|$ of the minimal check set for the given f, ϵ. Let $s(f, \epsilon)$ be the minimal s satisfying expr. 30, then, using the Varshamov bound, we have

$$|T(f, \epsilon)| \leqslant \sum_{j=0}^{s(f,\epsilon)-1} \binom{n-1}{j} \tag{31}$$

It follows, also, from expr. 30, that simple inequality checks (expr. 2) may be constructed only for 'smooth' functions $f(y)$, such that $\max_{y \in [0,1]} |f^{(s+1)}(y)|$ increases very slowly (or not at all) with increase of s.

In Table 1, the minimal s satisfying $\Delta_s|V^{\perp}(n, s+1)| \leqslant \epsilon$ is given for several important analytical functions for $n = 23$ and $\epsilon = 5 \times 10^{-3}$. The Table also gives corresponding approximation errors Δ_s taken from Reference 10, the parameters (n, K, d) of the codes $V(n, s+1)$ and check complexities $|T|$. As regards the parameters (n, K, d) of the code $V(n, s+1)$, n is the number of binary components in the code words, K the number of information bits and d the distance of the code; check complexity means the cardinality of the check set $T = V^{\perp}(n, s+1)$.

Thus, we may see from Table 1 that many important analytical functions have simple inequality checks of the type of expr. 2.

4.1 Example 4

Let us construct an optimal inequality check for the function

$$f(y) = y^{-0\cdot 5} \sin \frac{\pi}{2} y^{0\cdot 5}$$

with

$$\epsilon = 5 \times 10^{-3}$$

where

$$y = 2^{-23}x, \quad x \in \{0, 1, \ldots, 2^{23}-1\}$$

(see no. 7 in Table 1). This function can be approximated by the polynomial $P_2(y)$ of degree two:[10]

$$y^{-0\cdot 5} \sin \frac{\pi}{2} y^{0\cdot 5} = P_2(y) + \Delta_2(y)$$

where

$$P_2(y) = 0{\cdot}07287y^2 - 0{\cdot}64338y + 1{\cdot}57064$$

and

$$\max_y \Delta_2(y) = \Delta_2 \leqslant 14 \times 10^{-5}$$

Choose the (23, 18) code with the distance 3 and the check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Table 1: Linear inequality checks for some analytical functions ($n = 23, \epsilon = 5 \times 10^{-3}$)

No.	Function $f(y)$	Minimal degrees of approximating polynomial	Approximation error Δ_s	Parameters (n, K, d) of the chosen code $V(n, s+1)$	Check complexity $\lvert T\rvert$
1	e^y	7	2×10^{-7}	(23, 11, 8)	2^{12}
2	$10^{0.25y}$	5	18×10^{-7}	(23, 13, 6)	2^{10}
3	$\ln(1+y)$	6	1.5×10^{-6}	(23, 12, 7)	2^{11}
4	$\ln(1-ay)$ $a = 1-0.5\sqrt{2}$	5	4.1×10^{-7}	(23, 13, 6)	2^{10}
5	$\sin \frac{\pi}{4} y$	5	5×10^{-7}	(23, 13, 6)	2^{10}
6	$(\sin \frac{\pi}{4} y) y^{-2}$	4	12×10^{-7}	(23, 14, 5)	2^9
7	$\dfrac{\sin \frac{\pi}{2}\sqrt{y}}{\sqrt{y}}$	2	14×10^{-5}	(23, 18, 3)	2^5
8	$\cos \frac{\pi}{4} y$	4	99×10^{-7}	(23, 14, 5)	2^9
9	2^y	4	38×10^{-7}	(23, 14, 5)	2^9
10	$\sin y$	7	10^{-6}	(23, 11, 8)	2^{12}
11	$\Gamma(1+y)$	7	1.2×10^{-6}	(23, 11, 8)	2^{12}
12	$\dfrac{\frac{\pi}{2} - \sin^{-1} y}{\sqrt{1-y}}$	3	7×10^{-5}	(23, 17, 4)	2^6

as $V(n, s+1) = V(23, 3)$.[11] Then $|V^{\perp}(23, 3)| = 2^5$ and $\Delta_2 |V^{\perp}(23, 3)| \leqslant \epsilon = 5 \times 10^{-3}$. For the constant C, we have

$$C = 2^{-18} \sum_y P_2(y) = 40.74372.$$

Thus, we finally have the following optimal inequality check for our function:

$$\left| \sum_{\tau \in T} (2^{-23}(x \oplus \tau))^{-0.5} \sin \frac{\pi}{2}(2^{-23}(x \oplus \tau))^{0.5} - 40.7437 \right| \leqslant 5 \times 10^{-3}$$

for every $x \in \{0, \ldots, 2^{23}-1\}$ where $T = V^{\perp}(23, 3)$ is the set of all 32 linear mod 2 combinations of the rows of H.

We note also that all the results given above may be generalised to the case when x is represented in nonbinary form. If x is represented as a q-ary n-vector $x = (x_1, \ldots x_n)$, $(x_i \in \{0, \ldots, q-1\}$, $q \geqslant 2)$, then all the previous results remain valid, but the check set $V^{\perp}(n, s+1)$ must be replaced by the set $V_q^{\perp}(n, s+1) = \{(\tau_1, \ldots, \tau_n) | \overset{n}{\underset{i=1}{\oplus}} \tau_i x_i = 0$ for every $(x_1, \ldots, x_n) \in V_q(n, s+1)\}$ where $x_i, \tau_i \in \{0, \ldots, q-1\}$, the symbol '$\oplus$' stands for mod q addition and $V_q(n, s+1)$ is the maximal linear code in n-dimensional space of q-ary vectors with Hamming distance $s+1$.[11]

All checks considered above may be represented as a convolution over $GF(2)$:

$$\left| \sum_{\tau} a(\tau) f(x \oplus \tau) \right| \leqslant \epsilon \qquad (32)$$

where $a(\tau) \in \{0, 1\}$ for every τ. We note that the check complexity (number of nonzero values of $a(\tau)$) may sometimes be essentially decreased if we use checks with $a(\tau) \in \{0, \pm 1\}$ for every τ. For example, if

$$f(x) = \frac{x^t + b_1}{x^t + b_2} \; (x \in \{0, 1, \ldots, 2^n - 1\}, b_1 \geqslant b_2 > 0)$$

then we may construct the following check:

$$|f(x) - f(x \oplus (00 \ldots 01))| \leqslant \frac{b_1 - b_2}{b_2(b_2+1)} \qquad (33)$$

The problem with constructing optimal checks (expr. 32) with $a(\tau) \in \{0, \pm 1\}$ for the given $f(x)$ seems to be very difficult.

5 Error-detecting capability of linear inequality checks

As in References 1, 2, 3 and 4, we shall use the additive way of describing the influence of errors, namely, by the error e in a program or a device computing $f(x)$ $(x \in \{0, 1, \ldots, 2^n - 1\})$, we mean the function $e(x)$ $(x \in \{0, 1, \ldots, 2^n - 1\})$ such that, as a result of the error, our program or device computes $f(x) + e(x)$. We suppose also that for every $T \subset \{0, 1, \ldots, 2^n - 1\}$ either $\sum_{x \in T} e(x) = 0$ or $|\sum_{x \in T} e(x)| > 2\epsilon$. (The last condition may be used for the choice of ϵ for practical applications).

The error-detecting capability of proposed linear inequality checks depends on a specific implementation of a computational process for $f(x)$. We shall consider three widely used types of computational processes: polynomial approximation $f(x) \simeq P_1(x)$, rational approximation

$$f(x) \simeq \frac{P_1(x)}{Q(x)}$$

and continued fraction approximation

$$f(x) \simeq \frac{P_1(x)}{Q_1(x)} + \frac{P_2(x)}{Q_2(x)} + \ldots + \frac{P_t(x)}{Q_t(x)}$$

where $P_1, \ldots, P_t, Q_1, \ldots, Q_t$ represent polynomials (e.g. see Reference 10). By an error of multiplicity $\ell \geqslant 1$ we mean any error resulting in the replacement in a program computing $f(x)$ of ℓ coefficients in some of these polynomials by constants $C_1, \ldots, C_\ell$. We assume that every coefficient of these polynomials is stored in a corresponding m-bit memory cell; thus, the binary representation of constants $C_r (r = 1, \ldots, \ell)$ each contain m bits.

Suppose that expr. 2 is satisfied for $f(x)$. Then, for the error e, such that $|\sum_{\tau \in T} e(x)| > 2\epsilon$, we have $|\sum_{\tau \in T} (f(x \oplus \tau) + e(x \oplus \tau)) - C| > \epsilon$, and this error will be detected by the inequality check, expr. 2. Thus, if an error e cannot be detected by expr. 2, then $\sum_{\tau \in T} e(x \oplus \tau) = 0$ for every x. The last condition may be used for estimating the error-detecting capability of an inequality check expr. 2 For the practical implementation of linear inequality checks, we may verify expr. 2 for any given test pattern x (say, $x = 0$). Let us now describe the error-detecting capability in this case.

We denote the relative frequency of errors of multiplicity ℓ which cannot be detected by $\eta(\ell)$. (If the number of all possible errors of multiplicty ℓ tends to infinity, then $1 - \eta(\ell)$ tends to the probability of the detection of errors with multiplicity ℓ.)

If the error e is an asymmetric error (i.e. $e(x) \geqslant 0$ for every x or $e(x) \leqslant 0$ for every x) and for the given test pattern x there exists $\tau \in T$ such that $e(x \oplus \tau) \neq 0$, then $\eta(\ell) = 0$ for every ℓ, since for asymmetric errors $\sum_{\tau \in T} e(x \oplus \tau) \neq 0$.

Since for polynomial, rational or continued-fraction approximations any single error is an asymmetric error, all single errors are detected. For any error e of a multiplicity $\ell > 1$ and for any type of approximation for every $C_1, \ldots, C_{r-1}, C_{r+1}, \ldots, C_\ell$, there exists at most one C_r, such that $\sum_{\tau \in T} e(x \oplus \tau) = 0$ for the given test pattern x. Since the binary representation of C_r contains m bits, we have for $\eta(\ell)$

$$\eta(\ell) \leqslant (1 - \delta_{\ell, 1})(2^m - 1)^{-1} \qquad (34)$$

where $\delta_{\ell, 1}$ is the symbol of Kronecker.

Expr. 34 illustrates the good error-detecting capability of inequality checks for errors in coefficients in the case of polynomial, rational or continued-fraction approximations. As a disadvantage of these checks, we note that, if we use for the computation of $f(x)$ some expansion in orthogonal polynomials $P_i(x)$ (e.g. Chebyshev, Legendre or Hermite polynomials), that is,

$$f(x) \simeq \sum_{i=1}^{s} a_i P_i(x)$$

(where the degree of $P_i(x)$ is i), then for an error of multiplicity ℓ in coefficients $a_{i_1}, \ldots, a_{i_\ell}$ we have

$$e(x) = \sum_{r=1}^{\ell} (a_{i_r} - C_r) P_{i_r}(x)$$

$$(a_{i_r} \neq C_r, \quad r = 1, \ldots, \ell)$$

If $i_1 < i_2 < \ldots < i_\ell \leqslant s$ we have from eqn. 25 for every given test x

$$\sum_{\tau \in T} e(x \oplus \tau) = \sum_{\tau \in V^{\perp}(n, s+1)} e(x \oplus \tau)$$

$$= \sum_{r=1}^{\ell} (a_{i_r} - C_r) \sum_{\tau \in V^{\perp}(n, s+1)} P_{i_r}(x \oplus \tau) = 0$$

and this error cannot be detected by expr. 2. Thus, inequality checks are inefficient for computations by expansions in orthogonal polynomials.

For further improvement of the error-detecting capability of linear inequality checks, we may verify expr. 2 for several test patterns x, which will result in increasing the testing time. Since $T = V^{\perp}(n, s + 1)$, these test patterns have to be chosen as elements of $V(n, s + 1)$. If n is not too big (say $n < 20$), then we may use all elements of $V(n, s + 1)$ as test patterns. We shall describe the error-detecting capability for this case with respect to output errors.

By an output error e of multiplicity ℓ we mean any function $e(x)$ which is not equal to 0 at ℓ points (i.e. the multiplicity of output error in computing the function f is the number of distorted values $f(x)$). This definition is natural if errors in computing $f(x)$ are independent for different xs, as for example in the case where $f(x)$ is information stored in a memory cell whose address is x. An output error $e(x)$ cannot be detected by expr. 2 if, for every x,

$$\sum_{\tau \in V^{\perp}(n, s+1)} e(x \oplus \tau) = \sum_{\tau \in G} e(\tau) = 0$$

Thus, if the computed values of $f(x) + e(x)$ are stored in m-bit memory cells, then we have

$$\eta(2) \leqslant (|V(n, s + 1)| - 1)^{-1} (2^m - 1)^{-1}$$

$$\eta(3) \leqslant (|V(n, s + 1)| - 1)^{-1} (|V(n, s + 1)| - 2)^{-1} (2^m - 1)^{-1} \qquad (35)$$

and for every $\ell > 3$ we have

$$\eta(\ell) \leqslant (|V(n, s + 1)| - 1)^{-1} (2^m - 1)^{-1} \qquad (36)$$

For the estimation on $|V(n, s + 1)|$ in exprs. 35 and 36, we may use Hamming-Rao or Plotkin bounds.[11] Exprs. 35 and 36 illustrate the good error-detecting capability of linear inequality checks with respect to output errors, when the set of test patterns is $V(n, s + 1)$.

We note also that all results in this Section were obtained for the case where fault-free programs or devices compute the exact values of the function $f(x)$, but for many practical cases our program or device computes $f(x)$ only with some finite accuracy δ. Hence, all the previous results remain valid only if $\delta \ll \epsilon$.

6 Conclusions

We have described a method of error detection for numerical computations based on linear inequality checks. For the construction of these checks we use the techniques of least-absolute-error polynomial approximations and of linear error-correcting codes.

We have seen that a great variety of smooth analytical functions have simple inequality checks with good error-detecting capabilities.

The proposed error-detection method may be effectively used for functions with a good least-absolute-error polynomial approximation.

7 References

1 KARPOVSKY, M.G.: 'Error detection in digital devices and computer programs with the aid of linear recurrent equations over finite commutative groups,' *IEEE Trans.*, 1977, C-26, pp. 208–218

2 KARPOVSKY, M.G., and TRACHTENBERG, E.A.: 'Linear checking equations and error correcting capability for computation channels.' Proceedings of the IFIP Congress, 1977 (North Holland Publ. Co.)

3 KARPOVSKY, M.G., and TRACHTENBERG, E.A.: 'Fourier transform over finite groups for error detection and error correction in computation channels,' *Inf. Control*, 1979, 40, pp. 335–358

4 KARPOVSKY, M.G.: 'Error detection for polynomial computations,' *IEE. J. Comput. & Digital Tech.*, 1979, 2, (1), pp. 49–56

✓ 5 KARPOVSKY, M.G.: 'Finite orthogonal series in the design of digital devices' (John Wiley, 1976)

6 ANDREWS, K.C., and CASPARI, D.L.: 'A generalised technique for spectral analysis,' *IEEE Trans.*, 1970, C-19, pp. 16–25

7 APPLE, G., and WINTZ, P.: 'Calculation of Fourier transforms on finite Abelian groups,' *ibid.*, 1970, IT-16, pp. 233–236

8 KARPOVSKY, M.G.: 'Fast Fourier transforms over a finite non-Abelian group,' *ibid.*, 1977, C-26, pp. 1028–1031

9 HASTINGS, C.: 'Approximations for digital computers' (Princeton, New Jersey, 1955)

10 LYUSTERNIK, L.A., CHEZVONENKIS, D.A., and YANPOLSKII, A.R.: 'Handbook for computing elementary functions' (Pergamon Press, 1965)

11 PETERSON, W.W., and WELDON, E.J.: 'Error correcting codes', (MIT Press, Cambridge, Mass., 1972, 2nd edn.)

✓12 LECHNER, R.Y.: 'Harmonic analysis of switching functions,' *in* MAKHOPADHYAY, A. (Ed.): 'Recent development in switching theory' (Academic Press, New York, 1971)

✓13 KARPOVSKY, M.G.: 'Harmonic analysis over finite commutative groups in linearization problems for systems of logical functions,' *Inf. & Control*, 1977, 33, pp. 142–165

✓14 EDWARDS, C.R.: 'The application of the Rademacher-Walsh transform to Boolean function classification and threshold logic synthesis,' *IEEE Trans.*, 1975, C-24, pp. 48–62

✓ 15 EDWARDS, C.R., and HURST, S.L.: 'A digital synthesis procedure under functions symmetries and mapping methods,' *ibid.*, 1978, C-27, pp. 985–997

✓ 16 BESSLICH, P.W.: 'Determination of the irredundant forms of a Boolean function using Walsh-Hadamard analysis and dyadic groups,' *IEE J. Comput. & Digital Tech.*, 1978, 1, (4), pp. 143–151

17 PEARL, J.: 'Optimal dyadic models of time-invariant systems,' *IEEE Trans.*, 1975, C-24, pp. 598–603

18 KARPOVSKY, M.G., and TRACHTENBERG, E.A.: 'Some optimisation problems for convolution systems over finite groups,' *Inf. & Control*, 1977, 34, pp. 227–247