A Note on Self-Testing/Correcting Methods for Trigonometric Functions

Richard Cleve Michael Luby

TR-90-032

July 10, 1990

Abstract

Blum, Luby and Rubinfeld (1990) introduced the notion of self-testing/correcting for various numerical problems. We show how to apply some of their techniques to construct a self-testing/correcting pair for the problem of computing the sin and cos functions.

1 Introduction

Throughout this note, we use conventions in (Blum, Luby, and Rubinfeld, 1990) and assume the reader is familiar with them.

We consider the functions sin and cos in the following context. The domain is a discrete set of 2^n equally spaced angles (i.e. they are the angles expressed in radians as $\{i\frac{2\pi}{2^n} : 0 \leq i < 2^n\}$). Thus, each element in the domain has a natural representation as an n-bit string. The co-domain is assumed to arise from some countable subring of the real numbers. We assume that there is an implementation of an exact representation of the numbers in the co-domain in which addition, negation, and multiplication are inexpensive to perform relative to the cost of computing the sin and cos functions.

We show how to obtain efficient self-testing/correcting pairs exist for the sin and cos functions. This is done by showing that, with very slight modifications, the trigonometric functions can be viewed as a homomorphism between two abelian groups. Then some of the results of (Blum, Luby, and Rubinfeld, 1990) apply directly.

2 Results

Let $\mathbf{Z}_{2^n}$ denote the additive group of integers modulo 2^n. Let $\mathcal{R}$ be any countable subring of the real numbers. Let $ROT(\mathcal{R})$ denote the set of all matrices of the form

$$\begin{pmatrix} r & -s \\ s & r \end{pmatrix} ,$$

where $r^2 + s^2 = 1$. Note that the matrices in $ROT(\mathcal{R})$ form an abelian multiplicative group.

Consider the function $f : \mathbf{Z}_{2^n} \rightarrow ROT(\mathcal{R})$ defined as

$$f(x) = \begin{pmatrix} \cos(x\frac{2\pi}{2^n}) & -\sin(x\frac{2\pi}{2^n}) \\ \sin(x\frac{2\pi}{2^n}) & \cos(x\frac{2\pi}{2^n}) \end{pmatrix} ,$$

for all $x \in \mathbf{Z}_{2^n}$. Note that, from the identity $\sin(x\frac{2\pi}{2^n}) = \cos((x + 2^{n-1})\frac{2\pi}{2^n})$, the complexity of computing f is within a factor of two (plus the cost of one addition in $\mathbf{Z}_{2^n}$ and one negation in $\mathcal{R}$) of the cost of computing either sin or cos. Also, using a constant number of $\mathcal{R}$-arithmetic operations, one can determine whether any 2×2 matrix over $\mathcal{R}$ is in $ROT(\mathcal{R})$. Thus, disregarding a multiplicative constant of two, it is sufficient for us to consider the problem of designing self-testing/correcting programs for f.

The key idea here is that, for all $x, y \in \mathbf{Z}_{2^n}$,

$$f(x + y) = f(x) \cdot f(y) .$$

This follows from the elementary trigonometric identities

$$\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b)$$

and

$$\sin(a + b) = \sin(a)\cos(b) + \cos(a)\sin(b) ,$$

for all a and b. A more intuitive explanation is obtained by observing that the matrix $f(x)$, viewed as a linear transformation, corresponds to a "rotation" by $x\frac{2\pi}{2^n}$ radians.

Since f is a group homomorphism, and $\mathbf{Z}_{2^n}$ is a finite abelian group generated by 1, and $ROT(\mathcal{R})$ is a countable abelian group, we can apply the results of Blum, Luby and Rubinfeld (1990) directly. In particular, the self-correcting method is trivial: on input $x \in \mathbf{Z}_{2^n}$, one simply forms random splits of the form $x = x_1 + x_2$ and evaluates $f(x)$ as $f(x_1) \cdot f(x_2)$. For the self-testing part, one performs a linear test, checking whether $f(x_1+x_2) = f(x_1) \cdot f(x_2)$ on several independent, uniformly random pairs (x_1, x_2), and a neighbor test, checking whether $f(x+1) = f(x) \cdot f(1)$, on several uniformly random x. Formally, one applies the program "Generic Self-Testing Program 2", described on page 78 of (Blum, Luby, and Rubinfeld, 1990). Then the following theorem applies directly.

Theorem 2 (Blum, Luby, and Rubinfeld, 1990): *Generic Self-Testing Program 2 is $(\epsilon/36, \epsilon)$-self-testing for any $0 \leq \epsilon \leq 1$.*

3 Acknowledgment

Thanks to Manuel Blum for interesting discussions.

References

Blum, M., M. Luby and R. Rubinfeld (1990), "Self-Testing/Correcting with Applications to Numerical Problems," *Proc. 22nd Ann. ACM Symp. on Theory of Computing*, pp. 73–83.