MATHEMATICS OF COMPUTATION
VOLUME 56, NUMBER 194
APRIL 1991, PAGES 755-774

A STOCHASTIC ROUNDOFF ERROR ANALYSIS FOR THE FAST FOURIER TRANSFORM

DANIELA CALVETTI

To the memory of Peter Henrici

ABSTRACT. We study the accuracy of the output of the Fast Fourier Transform by estimating the expected value and the variance of the accompanying linear forms in terms of the expected value and variance of the relative roundoff errors for the elementary operations of addition and multiplication. We compare the results with the corresponding ones for the direct algorithm for the Discrete Fourier Transform, and we give indications of the relative performances when different rounding schemes are used. We also present the results of numerical experiments run to test the theoretical bounds and discuss their significance.

1. INTRODUCTION

1.1. **Purpose.** The subject of this paper is the analysis of the sensitivity of the Fast Fourier Transform to numerical errors. The Fast Fourier Transform (FFT) is an algorithm that evaluates the Discrete Fourier Transform of an n-dimensional data vector of complex numbers in a number of operations much smaller than the direct algorithm would require.

The model of error propagation that will be used is based on the usual assumptions of floating-point arithmetic. The model is linear in the sense that the *absolute* errors are approximated by the first-order terms of the Taylor expansion in local relative errors, and is stochastic in the sense that these local errors are regarded as random variables, independently and identically distributed (i.i.d.), for each elementary operation in which they arise. This method of analysis will allow us to measure the error in the final output by its statistical properties, i.e., its expected value and variance, rather than in worst-case terms. The statistical properties of the final output will depend on the properties of the local errors arising in elementary operations. Accordingly, a stochastic model of these local errors is introduced and its validity is tested by numerical experiments.

1.2. **Basic assumptions and methods of analysis.** In the machine M, real numbers $x \in R$ are represented by the elements of a discrete set R_M, which,

Received October 30, 1989.

1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 65G05; Secondary 42A38, 65T05.

Key words and phrases. Accompanying linear forms, floating-point arithmetic, Radix-2 Fast Fourier Transform, rounding errors, random variables.

if $x \neq 0$, are of the form

$$\tilde{x} = \pm m b^{l},$$

where b is the base of the machine, l is an integer such that $-L \leq l \leq U$, and the mantissa m is a T-digit number in base b such that $b^{-1} \leq m < 1$. If x and y are elements of R_M, and f is an elementary operation, the machine will compute $f(x, y)^{\Delta}$, which is in R_M and is such that

$$f(x, y)^{\Delta} = f(x, y)(1 - \varepsilon),$$

where $|\varepsilon| \leq \varepsilon_M$, the bound ε_M depending only on M. The quantity ε is the local *relative error* related to the operation f, and it can be expressed as follows:

$$\varepsilon = \frac{f(x, y) - f(x, y)^{\Delta}}{f(x, y)}.$$

In general, an algorithm is initiated from a set of data $(x_0, \ldots, x_{n-1})$ and proceeds through a set of intermediate results $t_1, \ldots, t_m$, each of which is computed through an elementary operation depending on the previous results and the initial data, i.e.,

$$t_k = f_k(t_1, \ldots, t_{k-1}; x_0, \ldots, x_{n-1}).$$

Hence, f_k represents an elementary operation which depends explicitly on at most two of the values $t_1, \ldots, t_{k-1}; x_0, \ldots, x_{n-1}$. In actual computation, each x_j is replaced by its machine representation, $\tilde{x}_j$, and each t_j is replaced by a computed value t_j^{Δ}. Hence, if ε_k is the local relative error for the kth operation, then

$$t_k^{\Delta} = f_k(t_1^{\Delta}, \ldots, t_{k-1}^{\Delta}; \tilde{x}_0, \ldots, \tilde{x}_{n-1})(1 - \varepsilon_k).$$

If we assume that the initial data are machine numbers, then, using the differentiability of the f_k, it can be shown by induction on k that

$$t_k^{\Delta} = t_k + \lambda_k(\varepsilon) + O(\|\varepsilon\|^2), \tag{1.1}$$

where λ_k is a homogeneous linear function of $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k)$.

The λ_k in (1.1) are called the *accompanying linear forms* of the algorithm. These forms can be used to compute the build-up of the first-order absolute errors for the numbers computed in the algorithm by using the fact that all of the ε_j are bounded in absolute value by ε_M:

$$|t_k^{\Delta} - t_k| \leq \lambda_k(|\varepsilon_M|) + O(\varepsilon_M^2).$$

An alternative approach, and one we shall use here, is to assume that the local relative errors ε_j are random variables with given distributions. In this case the distributional properties of these first-order absolute errors can be estimated by considering the properties of the accompanying linear forms, which are linear combinations of random variables and hence random variables themselves.

Because the λ_k are linear, we can write

$$\lambda_k(\varepsilon) = \sum_{j=1}^{k} g_{kj}\varepsilon_j,$$

where the g_{kj} are combinations of the t_j, and therefore depend on the initial data. If we assume that the ε_j are independent identically distributed (i.i.d.) random variables with means μ and variances σ^2, then the first two moments of the λ_k are of the form

$$E(\lambda_k) = \mu \cdot \sum_{j=1}^{k} g_{kj}$$

and

$$\operatorname{var}(\lambda_k) = \sigma^2 \cdot \left(\sum_{j=1}^{k} |g_{kj}|^2 \right) = \|g_k\|_2^2 \cdot \sigma^2,$$

where $\|\cdot\|_2$ indicates the l_2 norm, and $g_k = (g_{k1}, \ldots, g_{kk})$.

Some discussion of the interpretation of the local relative errors as random variables is called for. The local relative error at the jth step of the algorithm, ε_j, is for any machine a function of the jth operation, f_j, and the two values that this operation processes, say t_l^{Δ} and t_k^{Δ} $(l, k < j)$. Thus, it is a random variable if the values of t_l^{Δ} and t_k^{Δ} are considered to be random variables, and the distribution of ε_j will depend on the distributions of t_l^{Δ} and t_k^{Δ} and on the type of operation f_j. In this work we will consider only the two operations of addition and multiplication, and we will assume that these two operations produce different distributions for the local errors. Thus the local errors occurring in the algorithm under analysis will be separated into two groups, and the contribution of each to the final error will be computed.

Given the group of local relative errors corresponding to the operation of addition, we will assume that local errors, denoted by α_j, are *independently and identically distributed* (i.i.d.) with means μ_a and variances σ_a^2. Similarly, we denote the local errors arising from multiplication by π_j and denote their means by μ_m and variances by σ_m^2. The assumption of independence of the local errors introduced at different steps can be questioned. For example, it can be easily seen by following the algorithm for the FFT that in computing the t_{k+1}^{Δ} a few multiplications are repeated, therefore the same roundoff errors are repeated and the relative roundoff errors for these particular operations are correlated. However, these correlated errors affect different components of the output vector. Less obvious correlation might be observed through numerical experiments. In any case, it is reasonable to expect that, for large sets of data, the effect of correlation will be relatively small.

Since λ_k is linear, $\lambda_k(\varepsilon)$ can now be written

$$\lambda_k(\varepsilon) = (\lambda a)_k(\alpha) + (\lambda m)_k(\pi),$$

where $\alpha = (\alpha_1, \ldots, \alpha_i)$ and $\pi = (\pi_1, \ldots, \pi_l)$ with $i + l = k$. Thus,

$$E(\lambda_k) = E((\lambda a)_k) + E((\lambda m)_k) = \mu_a \cdot \sum_{j=1}^{i} g_{kj}^a + \mu_m \cdot \sum_{j=1}^{l} g_{kj}^m$$

and

$$\text{var}(\lambda_k) = \sigma_a^2 \cdot \sum_{j=1}^{i} |g_{kj}^a|^2 + \sigma_m^2 \cdot \sum_{j=1}^{l} |g_{kj}^m|^2 .$$

The expression for the mean will remain valid even without assuming independence of the relative errors. The expression for the variance, on the other hand, would need to contain all the terms of the correlation coefficients, which in the case of independent variables are zero.

The main result of this paper are the bounds on $E((\lambda a))$ and $E((\lambda m))$ for the final step of the Radix-2 Fast Fourier Transform expressed in terms of the expected values of the relative roundoff errors for the elementary operations of addition and multiplication.

1.3. **Literature.** Since the appearance of the article [7] in which Cooley and Tukey proved the existence and provided an implementation of the Fast Fourier Transform, several papers have appeared on the subject, and numerous versions of the algorithm have been implemented for use in digital computers. Our treatment of the FFT, based on reduction formulas for the case $n = 2^l$, follows Cooley, Lewis, and Welsh [5]. Several authors have dealt with implementations of FFT algorithms; in addition to the ones already mentioned, see Singleton [18], Uhrich [22], Cooley, Lewis, and Welsh [6], Gander and Mazzario [9], de Boor [8], and Temperton [21]. A survey of implementations can be found in Merz [16]. Recently, a highly optimized FFT subroutine has been included in the Engineering and Scientific Subroutine Library (ESSL) for the IBM 3090 Vector Facility (see Agarwal and Cooley [1]).

The question of accuracy when the FFT is implemented on a finite-length-word computer has been previously addressed by Gentleman and Sande [10]. They showed, by comparing upper bounds on roundoff noise, that the accumulated roundoff error, in floating-point arithmetic, is considerably lower than that obtained when the Discrete Fourier Transform is computed directly from the definition. Welsh [23] provides upper and lower bounds on the root mean square (RMS) error in a power-of-two algorithm in fixed-point arithmetic, and Ramos [17] derives upper bounds for the RMS and maximum roundoff errors in floating-point arithmetic for Radix-2 and Radix-4 FFT algorithms.

A statistical model for floating-point roundoff errors is used by Weinstein [24] to predict the output noise variance, and by Kanero and Liu [15], who derived expressions for the mean square error in a power-of-two FFT. A linear model of error propagation is used by Henrici [14] in a worst-case rounding error analysis for a power-of-two FFT in floating-point arithmetic.

Our roundoff error analysis for the Radix-2 FFT is more general than the work of Weinstein [24] and Kanero and Liu [15] because it does not assume

that the expected value of the relative roundoff errors is zero and, by differentiating between the relative errors coming from addition and those coming from multiplication, it can be used to predict the expected global error in the output when different rounding schemes are used. In the special case where the relative errors have zero mean, our upper bound on the variance agrees with theirs. In the present work we apply the same type of roundoff error analysis also to the traditional algorithm to compute the Discrete Fourier Transform (pre-FFT), and we are able to point out how first- and second-order moments of the relative errors play a fundamental role in establishing which algorithm gives a more accurate output. The advantage of our analysis over the worst-case approach of Ramos and Henrici is that the constant ε_M, largest possible relative roundoff error, is replaced by the mean value of the actual relative roundoff errors, which, in general, will be smaller. The results of numerical experiments are in good agreement with the theoretical findings.

The accompanying linear forms that we use to describe the first-order effects of the local relative errors on the intermediate results, and therefore on the output, were introduced by Stummel and Heiner [20] (see also Stummel [19]). For fixed-point arithmetic this technique was used earlier by Henrici [11].

2. Error analysis of Fast Fourier Transform

2.1. **Accompanying linear forms for the Discrete Fourier Transform.** The Discrete Fourier Transform is a method which can be used to analyze an arbitrary set of data by transforming it into periodic components of certain frequencies, whether or not the data appears to be periodic. Let $\omega_j = 2\pi j/n$ be the jth Fourier frequency, $0 \le j \le n/2$, and let $x_0, x_1, \ldots, x_{n-1}$ be any set of n complex numbers. Then it is well known that, for $t = 0, \ldots, n-1$,

$$x_t = \sum_{j=0}^{n-1} y_j \cdot \exp(i\omega_j t),$$

where

$$y_j = \frac{1}{n} \cdot \sum_{t=0}^{n-1} x_t \cdot \exp(-i\omega_j t). \tag{2.1}$$

The vector $(y_0, \ldots, y_{n-1})$ is called the Discrete Fourier Transform of $(x_0, x_1, \ldots, x_{n-1})$.

The computation of the Discrete Fourier Transform directly from the definition is quite inefficient. The computation requires n^2 complex multiplications and a similar number of complex additions, hence is very costly to perform on large data sets. It is well known that the algorithm known as the Radix-2 Fast Fourier Transform computes the transformation much more efficiently, on the order of $n \log_2 n$ operations. In this paper we compare the accuracy of the two methods of computation. Since the particular Fast Fourier Transform algorithm that we are interested in requires n to be a power of 2, we assume that $n = 2^l$.

Furthermore, the components of the data vector $x_0, \dots, x_{n-1}$ are assumed to be machine numbers, thus eliminating the error inherent in rounding the original data to machine numbers. In most applications the lack of accuracy in the initial data makes this initial rounding insignificant.

It should be noticed that, so far as the accuracy of the output is concerned, dividing by n in a base-2 machine does not introduce any roundoff error because, since n is a power of 2, it merely amounts to a shift in the exponent of the number.

We obtain expressions for the accompanying linear forms for the Direct Fourier Transform and, under reasonable assumptions, bounds on the mean and variance of these forms. In order to compute recursively the components

$$y_k = \frac{1}{n} \cdot \sum_{t=0}^{n-1} x_t w^{kt}, \qquad k = 0, \dots, n-1,$$

where $w = \exp(-2\pi i/n)$, we define the jth intermediate components

$$z_{k,j} = \sum_{t=0}^{j} x_t w^{kt}, \qquad k = 0, \dots, n-1,$$

for $j = 0, \dots, n-1$. Hence, for $j = 0, \dots, n-2$ and $k = 0, \dots, n-1$,

$$z_{k,j+1} = z_{k,j} + x_{j+1} w^{k\cdot(j+1)}. \tag{2.2}$$

Computationally,

$$\tilde{z}_{k,0} = x_0, \qquad \tilde{z}_{k,j+1} = (\tilde{z}_{k,j} + x_{j+1} \cdot \tilde{w}^{k\cdot(j+1)})^{\Delta},$$

where $\tilde{z}$ is the computed value of z. Since for machine numbers a, b, and c we have

$$(a + b \cdot c)^{\Delta} = (a + b \cdot c(1+\pi)) \cdot (1+\alpha),$$

where π is the relative error in multiplication and α is the relative error in addition, we get for each k and j,

$$\begin{aligned} \tilde{z}_{k,j+1} &= \{\tilde{z}_{k,j} + x_{j+1} \cdot \tilde{w}^{k\cdot(j+1)}(1+\pi_{k,j+1})\} \cdot (1+\alpha_{k,j+1}) \\ &\doteq (\tilde{z}_{k,j} + x_{j+1} \cdot \tilde{w}^{k\cdot(j+1)}) \cdot (1+\alpha_{k,j+1}) \\ &\quad + x_{j+1} \cdot \tilde{w}^{k\cdot(j+1)} \cdot \pi_{k,j+1}, \end{aligned} \tag{2.3}$$

where $\doteq$ indicates that only linear error terms have been retained. Defining the accompanying linear forms by

$$\lambda_{k,j+1} \doteq \tilde{z}_{k,j+1} - z_{k,j+1},$$

we obtain from (2.2) and (2.3)

$$\begin{aligned} z_{k,j+1} + \lambda_{k,j+1} \doteq\ & z_{k,j} + \lambda_{k,j} + x_{j+1} \cdot \tilde{w}^{k\cdot(j+1)} \\ & + z_{k,j+1}\alpha_{k,j+1} + x_{j+1} \cdot \tilde{w}^{k\cdot(j+1)} \cdot \pi_{k,j+1}. \end{aligned}$$

We now make the assumption that $\tilde{w} = w$. This is justified by the fact that most programs which compute the Discrete Fourier Transform perform the operations necessary to compute the w's in double precision. Now using (2.2) and assuming that $\tilde{w}^{k\cdot(j+1)} = w^{k\cdot(j+1)}$, we obtain, from (2.3), the recursion equation for the linear forms

$$\lambda_{k,j+1} = \lambda_{k,j} + z_{k,j+1}\alpha_{k,j+1} + x_{j+1}w^{(j+1)k}\cdot\pi_{k,j+1}. \tag{2.4}$$

Decomposing the linear forms into the components generated by addition and multiplication, we have

$$\lambda_{k,j} = (\lambda a)_{k,j} + (\lambda m)_{k,j}$$

where the $(\lambda a)_{k,j}$ are only functions of the addition errors and the $(\lambda m)_{k,j}$ are only functions of the multiplication errors. Thus, from (2.4) we obtain the difference equations

$$(\lambda a)_{k,j+1} = (\lambda a)_{k,j} + z_{k,j+1}\cdot\alpha_{k,j+1}$$

and

$$(\lambda m)_{k,j+1} = (\lambda m)_{k,j} + x_{j+1}w^{(j+1)k}\cdot\pi_{k,j+1}.$$

The initial conditions for the above difference equations are

$$(\lambda a)_{k,0} = 0, \qquad (\lambda m)_{k,0} = 0.$$

Using our assumption that division by n introduces no additional error, we see that

$$(\lambda a)_{k,n-1} = \frac{1}{n}\cdot\sum_{i=1}^{n-1} z_{k,i}\alpha_{k,i} = \frac{1}{n}\cdot\sum_{i=1}^{n-1}\left(\sum_{t=0}^{i} x_t w^{kt}\right)\alpha_{k,i}$$

and similarly

$$(\lambda m)_{k,n-1} = \frac{1}{n}\cdot\sum_{t=1}^{n-1} x_t w^{tk}\cdot\pi_{k,t}$$

for each $k = 0, \ldots, n-1$.

If we assume that the $\alpha_{k,i}$'s are i.i.d. with mean μ_a and variance σ_a^2 and that the $\pi_{k,t}$'s are i.i.d. with mean μ_m and variance σ_m^2, then we can obtain the following bounds on the expected value of the linear form for addition and multiplication errors:

$$\begin{aligned} |E((\lambda a)_{k,n-1})| &\le \frac{1}{n}\cdot\mu_a\cdot\sum_{i=1}^{n-1}\sum_{t=0}^{i}|x_t| = \frac{1}{n}\cdot\mu_a\cdot\sum_{i=1}^{n-1}(i+1)\|\mathbf{x}\|_\infty \\ &= \mu_a\cdot\frac{1}{2}\left(n+1-\frac{2}{n}\right)\|\mathbf{x}\|_\infty \end{aligned} \tag{2.5}$$

and

$$|E((\lambda m)_{k,n-1})| \le \mu_m \cdot \left(1 - \frac{1}{n}\right) \|\mathbf{x}\|_\infty \tag{2.6}$$

for $k = 0, 1, \ldots, n-1$.

Under the i.i.d. assumption on the errors we can also obtain the following bounds for the variances of $(\lambda a)_{k,n-1}$ and $(\lambda m)_{k,n-1}$:

$$\begin{aligned} |\operatorname{var}((\lambda a)_{k,n-1})| &\le \frac{1}{n^2} \cdot \sigma_a^2 \cdot \sum_{i=1}^{n-1} \sum_{t=0}^{i} |x_t|^2 \le \frac{1}{n^2} \cdot \sigma_a^2 \cdot \sum_{i=1}^{n-1} \|\mathbf{x}\|_2^2 \\ &= \sigma_a^2 \cdot \left(\frac{n-1}{n^2}\right) \cdot \|\mathbf{x}\|_2^2 \end{aligned} \tag{2.7}$$

and

$$|\operatorname{var}((\lambda m)_{k,n-1})| \le \frac{1}{n^2} \cdot \sigma_m^2 \cdot \sum_{s=1}^{n-1} |x_s|^2 \le \frac{1}{n^2} \cdot \sigma_m^2 \cdot \|\mathbf{x}\|_2^2. \tag{2.8}$$

2.2. **Accompanying linear forms for the Radix-2 Fast Fourier Transform.** The FFT, which was discovered by Cooley and Tukey in 1965 [7], reduces the number of arithmetic operations required to compute the Fourier transform. If n is a power of 2, $n = 2^l$, the "Duplication Theorem" can be applied recursively until one arrives at the Fourier Transform of a vector of size 1, which is just the identity. This algorithm is often referred to as the Radix-2 FFT; the implementation used here organizes the intermediate results in a pyramidal structure and can be found in Henrici [13, pp. 367–371].

If we denote by $\mathbf{q}(j, m)$ the mth vector of the jth step of the iteration, then for $k = 0, 1, \ldots, 2^j - 1$

$$q_k(j, m) = x_\nu, \quad \text{with } \nu = p_j(k) + 2^j m,$$

where p_j is the bit-reversing function that maps the set of integers $\{0, 1, \ldots, 2^j - 1\}$ onto itself by sending

$$m := m_0 + 2m_1 + \cdots + 2^{j-1} m_{j-1}, \qquad m_i \in \{0, 1\},$$

to

$$p_j(m) := m_{j-1} + 2m_{j-2} + \cdots + 2^{j-1} m_0$$

if $j \neq 0$, while $p_0(0) := 0$. Now let $z_k(j, m)$, for $k = 0, \ldots, 2^j - 1$, denote the components of the Discrete Fourier Transform of the vector $\mathbf{q}(j, m)$; from the duplication formulas it follows that

$$z_0(0, m) = x_\nu, \quad \text{where } \nu = p_l(m), \tag{2.9}$$

and

$$z_k(j, m) = \tfrac{1}{2}(z_k(j-1, 2m) + w^{k2^{l-j}} z_k(j-1, 2m+1)), \tag{2.10}$$

$$z_{k+2^{j-1}}(j, m) = \tfrac{1}{2}(z_k(j-1, 2m) - w^{k2^{l-j}} z_k(j-1, 2m+1)), \tag{2.11}$$

where $k = 0, 1, \ldots, 2^{j-1} - 1$; $m = 0, 1, \ldots, 2^{l-j} - 1$; and $j = 1, 2, \ldots, l$. Thus starting at $j = 0$, we can compute the Fourier coefficients recursively to obtain the coefficients for $j = l$ which are the Fourier coefficients of $\mathbf{x}$.

Since each row of the pyramid contains the same number of entries, 2^l, each row can be thought of as a vector $\mathbf{z}_j$ with components $\mathbf{z}_j(s)$, $s = 0, \ldots, 2^l - 1$, where

$$\mathbf{z}_j(2^j m + k) := z_k(j, m)$$

for $j = 0, 1, \ldots, l$, $m = 0, 1, \ldots, 2^{l-j} - 1$, and $k = 0, 1, \ldots, 2^j - 1$. With this notation, the formulas (2.9)–(2.11) become

$$\begin{aligned}\mathbf{z}_j(2^j m + k + b2^{j-1}) = \tfrac{1}{2} \cdot \{&\mathbf{z}_{j-1}(2^{j-1} \cdot 2m + k) \\ &+ (-1)^b w^{k2^{l-1}} \mathbf{z}_{j-1}(2^{j-1}(2m+1) + k)\},\end{aligned} \tag{2.12}$$

where $k = 0, 1, \ldots, 2^{j-1} - 1$, $m = 0, 1, \ldots, 2^{l-j} - 1$, $j = 1, 2, \ldots, l$, and $b = 0, 1$, and

$$\mathbf{z}_0(m) = x(p_l(m)) \tag{2.13}$$

for $m = 0, 1, \ldots, 2^l - 1$.

We can rewrite (2.12) in matrix notation in the following way:

$$\mathbf{z}_j = W_j \mathbf{z}_{j-1}, \tag{2.14}$$

where W_j is a block diagonal matrix of the form

$$W_j = \frac{1}{2} \begin{bmatrix} I_j & D_j & & & \\ I_j & -D_j & & & \\ & & \ddots & & \\ & & & I_j & D_j \\ & & & I_j & -D_j \end{bmatrix},$$

where I_j is the $2^{j-1} \times 2^{j-1}$ identity matrix and D_j is the $2^{j-1} \times 2^{j-1}$ diagonal matrix with entries

$$D_j(p, p) = w^{(p-1)2^{l-j}}.$$

In particular, by induction on j, we have that $\mathbf{z}_{n-1} = F_{n-1}\mathbf{z}_0$, where $F_{n-1} = \prod_{r=n-1}^{1} W_r$.

In the computation of the Discrete Fourier Transform via the Radix-2 Fast Fourier Transform of a data vector $\mathbf{x}$ with $n = 2^l$ entries we will need to compute the following quantities for $b = 0$ and $b = 1$:

$$\begin{aligned}\mathbf{z}_j(2^j m + k + b2^{j-1}) = \tfrac{1}{2} \cdot [&\mathbf{z}_{j-1}(2^{j-1} \cdot 2m + k) + (-1)^b w^{k2^{l-j}} \\ &\cdot \mathbf{z}_{j-1}(2^{j-1}(2m+1) + k)], \\ \mathbf{z}_0(m) = \mathbf{x}[&p_l(m)],\end{aligned} \tag{2.15}$$

for $m = 0, 1, \ldots, 2^l - 1$, where $w := \exp(-2\pi i / 2^l)$.

Let α_j be the local error due to addition and let π_j be the local error due to multiplication in the computation of the components of $\mathbf{z}_j$. Then in computation:

$$\tilde{z}_j(2^j m + k + b2^{j-1}) = \tfrac{1}{2} \cdot [\tilde{z}_{j-1}(2^{j-1} \cdot 2m + k) + (-1)^b \tilde{w}^{k2^{l-j}} \tilde{z}_{j-1}(2^{j-1}(2m+1) + k)]^{\Delta}.$$

Since $\tilde{z} = z + \lambda$, multiplication by $\frac{1}{2}$ produces no error, and we assume that $w^{k2^{l-j}}$ does not contain any error, it follows that

$$\begin{aligned}
&\tilde{z}_j(2^j m + k + b2^{j-1}) \\
&\quad = z_j(2^j m + k + b2^{j-1}) + \lambda_j(2^j m + k + b2^{j-1}) \\
&\quad \doteq z_j(2^j m + k + b2^{j-1}) \\
&\qquad + \tfrac{1}{2} \cdot [\lambda_{j-1}(2^j m + k) + (-1)^b w^{k2^{l-j}} \cdot \lambda_{j-1}(2^j m + 2^{j-1} + k) \\
&\qquad\qquad + (-1)^b w^{k2^{l-j}} z_{j-1}(2^j m + 2^{j-1} + k)\pi(j, m, k, b)] \\
&\qquad + z_j(2^j m + k + b2^{j-1})\alpha(j, m, k, b),
\end{aligned}$$

where $\alpha(j, m, k, b)$ and $\pi(j, m, k, b)$ are the relative errors for addition and multiplication introduced in the computation of the $(2^j m + k + b2^{j-1})$th entry of the vector $\mathbf{z}_j$. Therefore,

$$\begin{aligned}
&\lambda_j(2^j m + k + b2^{j-1}) \\
&\quad = \tfrac{1}{2} \cdot [\lambda_{j-1}(2^j m + k) + (-1)^b w^{k2^{l-j}} \cdot \lambda_{j-1}(2^j m + 2^{j-1} + k) \\
&\qquad\qquad + (-1)^b w^{k2^{l-j}} z_{j-1}(2^j m + 2^{j-1} + k)\pi(j, m, k, b)] \\
&\qquad + z_j(2^j m + k + b2^{j-1})\alpha(j, m, k, b).
\end{aligned}$$

Again decomposing λ as $\lambda = \lambda\alpha + \lambda\pi$ to account separately for errors due to additions and errors due to multiplications, we have that

$$\begin{aligned}
\lambda\alpha_j(2^j m + k + b2^{j-1}) &= \tfrac{1}{2} \cdot [\lambda\alpha_{j-1}(2^j m + k) \\
&\qquad + (-1)^b w^{k2^{l-j}} \lambda\alpha_{j-1}(2^j m + 2^{j-1} + k)] \\
&\qquad + z_j(2^j m + k + b2^{j-1})\alpha(j, m, k, b)
\end{aligned} \tag{2.16}$$

and

$$\begin{aligned}
&\lambda m_j(2^j m + k + b2^{j-1}) \\
&\quad = \tfrac{1}{2} \cdot [\lambda m_{j-1}(2^j m + k) + (-1)^b \cdot w^{k2^{l-j}} \lambda m_{j-1}(2^j m + 2^{j-1} + k) \\
&\qquad\qquad + (-1)^b w^{k2^{l-j}} z_{j-1}(2^j m + 2^{j-1} + k)\pi(j, m, k, b)],
\end{aligned} \tag{2.17}$$

where b is either 0 or 1. The initial conditions are

$$\lambda\alpha_0(m) = 0 \tag{2.18}$$

and

$$\lambda\pi_0(m) = 0, \tag{2.19}$$

since we assume that the data are machine numbers and that the bit reversing function does not introduce any rounding error.

Let $\boldsymbol{\lambda a}_j$ be the vector of $\lambda a_j(p)$, for $0 \le p < n$, $\mathbf{z}_j$ the vector of the $z_j(p)$, and $\boldsymbol{\alpha}_j$ the vector of relative roundoff errors coming from additions introduced in the components of $\mathbf{z}_j$ at the jth iteration. Then, in vector notation,

$$\begin{aligned} \boldsymbol{\lambda a}_{j+1} &= W_{j+1}\boldsymbol{\lambda a}_j + \mathbf{z}_{j+1} \square \boldsymbol{\alpha}_{j+1}, \qquad j \ge 0, \\ \boldsymbol{\lambda a}_0 &= \mathbf{0}, \end{aligned}$$

where by $\mathbf{x} \square \mathbf{y}$ we indicate the componentwise multiplication of the two vectors $\mathbf{x}$ and $\mathbf{y}$. Therefore,

$$\boldsymbol{\lambda a}_{j+1} = \sum_{k=0}^{j-2} W_{j+1} \cdots W_{k+2}(\mathbf{z}_{k+1} \square \boldsymbol{\alpha}_{k+1}) + W_{j+1}(\mathbf{z}_j \square \boldsymbol{\alpha}_j) + (\mathbf{z}_{j+1} \square \boldsymbol{\alpha}_{j+1}).$$

In particular,

$$\boldsymbol{\lambda a}_l = \sum_{k=0}^{l-3} W_l \cdots W_{k+2}(\mathbf{z}_{k+1} \square \boldsymbol{\alpha}_{k+1}) + W_l(\mathbf{z}_{l-1} \square \boldsymbol{\alpha}_{l-1}) + (\mathbf{z}_l \square \boldsymbol{\alpha}_l). \tag{2.20}$$

Since each matrix W_p has exactly two nonzero entries in each row, each one of absolute value one-half, and for each k the $\alpha_k(p)$ are i.i.d. random variables with mean μ_a, we have that, for each component of $\boldsymbol{\lambda a}_l$,

$$|E(\lambda a_l)| \le \sum_{k=0}^{l-2} \|\mathbf{z}_{k+1}\|_\infty \mu_a + \|\mathbf{z}_l\|_\infty \mu_a.$$

Since $\mathbf{z}_k = \prod_{r=k}^{1} W_r \mathbf{z}_0$, we have $\|\mathbf{z}_k\|_\infty \le \|\mathbf{x}\|_\infty$. Hence, for each entry of $\boldsymbol{\lambda a}_l$,

$$|E(\lambda a)_l| \le \log_2 n \cdot \|\mathbf{x}\|_\infty \mu_a. \tag{2.21}$$

We now find a bound for the variance of the components of the vector $\boldsymbol{\lambda a}_j$, that is, for the diagonal entries of the covariance matrix of $\boldsymbol{\lambda a}_j$, $\operatorname{cov}(\boldsymbol{\lambda a}_j)$.

Proposition 2.2.1. *If x is an n-vector of independent random variables and A is an $n \times n$ matrix, then*

$$\operatorname{cov}(A\mathbf{x}) = A \operatorname{cov}(\mathbf{x}) A',$$

where $\operatorname{cov}(\mathbf{x})$ *is the diagonal matrix such that* $\operatorname{cov}(\mathbf{x})_{ii} = \operatorname{var}(x_i)$ *and* A' *is the transpose of* A.

Proof. See Arnold [2, p. 41]. □

Since the components of the vectors α_k are assumed to be i.i.d. random variables with mean μ_a and variance σ_a^2, the covariance matrix of $(\mathbf{z}_k \square \alpha_k)$ is a diagonal matrix, A_k, such that $A_k(d, d) = \sigma_a^2 (z_k(d))^2$. From (2.20) and Proposition 2.2.1 it follows that

$$\begin{aligned}\operatorname{cov}(\boldsymbol{\lambda}\mathbf{a}_j) = \sigma_a^2 \sum_{k=0}^{j-3} W_j \cdots W_{k+2} Z_{k+1}^2 W_{k+2}' \cdots W_j' \\ + \sigma_a^2 W_j Z_{j-1}^2 W_j' + \sigma_a^2 Z_j^2,\end{aligned}$$

where Z_{k+1}^2 is the $n \times n$ diagonal matrix such that

$$Z_{k+1}^2(d, d) = (z_{k+1}(d))^2.$$

In particular,

$$\operatorname{cov}(\boldsymbol{\lambda}\mathbf{a}_l) = \sigma_a^2 \left[\sum_{k=0}^{l-3} W_l \cdots W_{k+2} Z_{k+1}^2 W_{k+2}' \cdots W_l' + W_l Z_{l-1}^2 W_l' + Z_l^2 \right].$$

Since each W_k has exactly two entries in each row and column, each one of absolute value one-half, and $|z_k| \le \|\mathbf{x}\|_\infty$, then for each entry of the matrix $\operatorname{cov}(\boldsymbol{\lambda}\mathbf{a}_l)$

$$\operatorname{cov}(\lambda a_l) \le \log_2 n \sigma_a^2 \|\mathbf{x}\|_\infty^2. \tag{2.22}$$

We now turn our attention to λm. Let $\boldsymbol{\lambda}\mathbf{m}_j$ be the vector with components $\lambda m_j(p)$, and $\boldsymbol{\pi}_j$ the vector of the relative roundoff errors coming from multiplication introduced at the jth iteration. Then

$$\begin{aligned}\boldsymbol{\lambda}\mathbf{m}_j(2^j m + k + b2^{j-1}) = W_j \boldsymbol{\lambda}\mathbf{m}_{j-1}(2^j m + k) \\ + \frac{1}{2}(-1)^b w^{k2^{l-j}} (\mathbf{z}_{j-1} \square \boldsymbol{\pi}_j)(2^j m + k), \qquad j \ge 0,\end{aligned}$$

$\boldsymbol{\lambda}\mathbf{m}_0 = \mathbf{0}$.

If we define the vector $\mathbf{b}_j$ as

$$b_j(p) = \tfrac{1}{2} w^{p2^{l-j}} (-1)^{p_{j-1}},$$

where $p = \sum_{s=0}^{l-1} p_s 2^s$, then (2.17) can be written, in vector notation, as

$$\boldsymbol{\lambda}\mathbf{m}_{j+1} = \prod_{r=j+1}^{1} W_r \boldsymbol{\lambda}\mathbf{m}_0 + \sum_{k=0}^{j} \prod_{t=j+1}^{1} W_t \prod_{s=1}^{k} W_s^{-1} \cdot (\mathbf{b}_k \square \mathbf{z}_k \square \boldsymbol{\pi}_{k+1}).$$

In particular,

$$\begin{aligned}\boldsymbol{\lambda}\mathbf{m}_l = \sum_{k=0}^{l-3} W_l \cdots W_{k+2} (\mathbf{b}_{k+1} \square \mathbf{z}_k \square \boldsymbol{\pi}_{k+1}) \\ + W_l (\mathbf{b}_{l-1} \square \mathbf{z}_{l-2} \square \boldsymbol{\pi}_{l-1}) + (\mathbf{b}_l \square \mathbf{z}_{l-1} \square \boldsymbol{\pi}_l).\end{aligned} \tag{2.23}$$

Since, for each k, the $\pi_k(p)$ are i.i.d. random variables with mean μ_m, we have, for each component of $\boldsymbol{\lambda}\mathbf{m}_l$,

$$|E(\lambda m_l)| \leq \frac{1}{2}\sum_{k=0}^{l-2} \|\mathbf{z}_k\|_\infty \mu_m + \frac{1}{2}\|\mathbf{z}_{l-1}\|_\infty \mu_m,$$

hence

$$|E(\lambda m_l)| \leq \tfrac{1}{2}\log_2 n \|\mathbf{x}\|_\infty \mu_m. \tag{2.24}$$

Now we obtain an estimate for a bound for the variance of the components of the vector $\boldsymbol{\lambda}\mathbf{m}_j$. Since the components of the vectors $\boldsymbol{\pi}_k$ are assumed to be i.i.d. random variables with mean μ_m and variance σ_m^2, the covariance matrix of $(\mathbf{b}_k \square \mathbf{z}_k \square \boldsymbol{\alpha}_k)$ is a diagonal matrix, A_k, such that $A_k(d, d) = \frac{1}{4}\sigma_a^2(z_k(d))^2$. From (2.23) and Proposition 2.2.1 it follows that

$$\operatorname{cov}(\boldsymbol{\lambda}\mathbf{m}_j) = \frac{1}{4}\dot{\sigma}_m^2 \left[\sum_{k=0}^{j-3} W_j \cdots W_{k+2} Z_k^2 W_{k+2}' \cdots W_j' + \sigma_m^2 W_j Z_{j-2}^2 W_j' + \sigma_m^2 Z_{j-1}^2\right],$$

where Z_{k+1}^2 is the $n \times n$ diagonal matrix such that

$$Z_{k+1}^2(d, d) = (z_{k+1}(d))^2.$$

In particular,

$$\operatorname{cov}(\boldsymbol{\lambda}\mathbf{m}_l) = \frac{1}{4}\sigma_m^2 \left[\sum_{k=0}^{l-3} W_l \cdots W_{k+2} Z_k^2 W_{k+2}' \cdots W_l' + W_l Z_{l-2}^2 W_l' + Z_{l-1}^2\right].$$

Hence, for each entry of the matrix $\operatorname{cov}(\boldsymbol{\lambda}\mathbf{m}_l)$,

$$\operatorname{cov}(\boldsymbol{\lambda}\mathbf{m}_l) \leq \tfrac{1}{4}\sigma_m^2 \|\mathbf{x}\|_\infty^2 \cdot \log_2 n. \tag{2.25}$$

3. A ROUNDING MODEL AND EXPERIMENTAL RESULTS

The purpose of this section is to provide estimates of the first and second moment of the relative errors due to the elementary operations used by the algorithms that we have analyzed, that is, addition and multiplication. These relative errors are for a particular type of rounding and are not necessarily applicable for any real machine. However, they will illustrate the type of rounding error magnitude that can be encountered and will serve as a model for establishing the validity of the error estimates in the Fast Fourier Transform.

In our rounding model we assume that b, an even positive integer, is the base of the number system in the machine and each floating-point number $\tilde{x}$ is represented by the pair (m, l) such that

$$\tilde{x} = mb^l, \qquad -L \leq l \leq U,$$

and m is a T-digit number, with first digit different from zero if $x \neq 0$. Given any number x between $b^{-(L+1)}$ and b^{U-1}, we round it to T-digit precision by adding $\frac{b}{2}^{-T-1}$ to it, then writing the sum in the form $\hat{m}b^{l}$, where $\hat{m}$ is a number in base b whose first digit is different from zero, and finally by truncating $\hat{m}$ to T digits. Note that if $x \in [0, b^{-(L+1)})$, the roundoff process we use is undefined (on a machine, an underflow message would be returned). In this case we will set $\tilde{x} = 0$. In the present paper we will not worry about the machine representation of numbers greater than or equal to b^{U}, which on a machine would cause an overflow error.

If we assume the real data to have a uniform distribution, the relative roundoff error for the operations of addition and multiplication inherit a stochastic structure. However, owing to the complexity of the model, we decided to perform numerical experiments to estimate the expected value and the variance of the relative roundoff error for the operations of addition and multiplication in our model.

In order to estimate numerically the first and second moment of the relative roundoff error for addition, we generated 60,000 pairs of independent, pseudo-random numbers from the uniform $(-1, 1)$ distribution. The interval $(-1, 1)$ was chosen to be consistent with the numerical test of the roundoff error propagation for the FFT on independent random data from the uniform (0,1) distribution, where pairs of numbers may be either added or subtracted. Each number was rounded to a T-digit base 10 floating-point number (where T ranged from 3 to 8), and the sum of the two rounded numbers, $x + y$, was evaluated in 18-digit precision and rounded to T digits, $(x + y)^{\Delta}$. The quantity $((x + y) - (x + y)^{\Delta})/(x + y)$ was computed in 18-digit precision and its mean and variance were evaluated over the 60,000 samples. It is interesting to notice that the values obtained for the mean are extremely consistent with a value of approximately $-.12 \times 10^{-T}$ for all values of T chosen, while the value of the variance is clearly of order 10^{-2T}. It is important to note that a common assumption that the mean is zero (e.g., Weinstein [24]) is not valid, at least for our model.

The expected value and variance of the relative roundoff error for multiplication was estimated numerically in a very similar way. For each value of T chosen, 100,000 pairs of independent random numbers from the uniform (0,1) distribution were generated and then rounded to T-digit floating-point numbers in base 10. The product of each pair of rounded numbers was computed in 18-digit precision, $x \cdot y$, then rounded to T-digit precision, $(x \cdot y)^{\Delta}$. The quantity $((x \cdot y) - (x \cdot y)^{\Delta})/(x \cdot y)$ was computed and its mean and variance evaluated for the 100,000 pairs of numbers. The value of the sample variance was 1.5×10^{-2T} for all values of T, while the value of the sample mean ranged from $-.1 \times 10^{-T-1}$ to $.3 \times 10^{-T-2}$. To test our theoretical predictions on the mean and variance of the global error for the Radix-2 FFT, we generated several vectors of complex numbers whose real and imaginary parts are independent

TABLE 3.1

Infinity norm of the mean and variance of the global error in the FFT. Sample size is 10,000 *for* $M = 8, 16, 32$, *and* 5,000 *for* $M = 64$.

T	M	mean	variance
4	8	.28 E−3	.21 E−07
5	8	.28 E−4	.21 E−09
6	8	.28 E−5	·.21 E−11
4	16	.44 E−3	.24 E−07
5	16	.44 E−4	.25 E−09
6	16	.45 E−5	.25 E−11
4	32	.73 E−3	.35 E−07
5	32	.74 E−4	.36 E−09
6	32	.73 E−5	.35 E−11
4	64	.11 E−2	.37 E−07
5	64	.11 E−3	.38 E−09
6	64	.11 E−4	.37 E−11

pseudorandom numbers from the uniform (0,1) distribution. For each value of $T = 4, 5, 6$ we generated 10,000 vectors of length $M = 8$, 16, and 32, and 5,000 vectors of length $M = 64$. For each vector we rounded the real and imaginary part of each entry to T digits and we computed the Discrete Fourier Transform via the Radix-2 Fast Fourier Transform algorithm, where the result of each addition and multiplication, other than those used for the computation of the w's, was rounded to T digits. We then computed the Discrete Fourier Transform of the same rounded data via the FFT algorithm with all operations performed in double precision (18 digits in the machine we used). The mean and the variance of the absolute value of the difference in each component of the Discrete Fourier Transform was then computed.

In order to compare the sizes of the vectors of means and variances, we computed their infinity norms. The infinity norms of these vectors for different length of the data vector and different values of T are listed in Table 3.1 and Table 3.2. Notice that the values are extremely consistent, and their dependence on T as well as on the length of the data vector is quite clear.

The theoretical results suggested that the infinity norm of the mean absolute error should grow like $\log_2 n \cdot \|\mathbf{x}\|_\infty \mu$; therefore we expect the ratio between the infinity norms of the mean absolute error for consecutive values of $l = \log_2 n$ to be approximately

$$\frac{\log_2(2n)}{\log_2 n} \cdot h = \left[1 + \frac{1}{\log_2 n}\right] \cdot h,$$

where h is the ratio of the infinity norms of the $2n$-dimensional data vectors to the infinity norm of n-dimensional data vectors. This value approaches 1

TABLE 3.2
Infinity norm of the mean and variance of the global error in the FFT. Sample size is 1, 000 *for each* T.

T	M	mean	variance
5	128	.16E−3	.40E−09
6	128	.16E−4	.41E−11
7	128	.16E−5	.40E−13
5	256	.25E−3	.59E−09
6	256	.25E−4	.58E−11
7	256	.25E−5	.58E−13
5	512	.37E−3	.63E−09
6	512	.37E−4	.63E−11
7	512	.37E−5	.63E−13
5	1024	.55E−3	.67E−09
6	1024	.55E−4	.68E−11
7	1024	.55E−5	.66E−13

as n becomes large. The numerical experiments indicate that the ratio of the infinity norm of the mean absolute error for consecutive values of l decreases as n becomes larger, going from about 1.9 to approximately 1.4. It should be pointed out that even though the results of the numerical experiments are in quite good agreement with the theoretical results, this is not sufficient to assume that the predicted expected value and variance of the global error in the output of the Fast Fourier Transform will be as close to the sample values when the distribution of the real and imaginary part of the data is not uniform on (0,1). Numerical experiments testing the deviation of the theoretical expected value and variance of the global error from the sample values corresponding to different distributions of the data are currently being performed.

4. SUMMARY OF RESULTS

A statistical model of error propagation has been used to compare bounds on the absolute roundoff error of the Radix-2 Fast Fourier Transform (FFT) and the traditional Fourier Transform (TFT). In this paper the bounds are given in terms of the norm of the input data, the size of the input vector, and the expected values of the relative roundoff errors arising from the operations of addition and multiplication. The bounds themselves are expected values and variances of the linear part of the absolute roundoff error for the output vector of the Fourier Transform. These estimates agree with earlier results in the sense that the upper bound on the variance of the absolute error is essentially the same as in Kanero and Liu [15], and the bounds on the mean absolute error are of the same order as those found by Henrici [14] when the same scalar is used in the definition of FFT. A new contribution given by our work is that we apply

TABLE 4.1

	TFT	FFT
$\lvert E(\lambda a)\rvert \leq$	$\frac{1}{2}(n+1-\frac{2}{n})\mu_a\lVert\mathbf{x}\rVert_\infty$	$\log_2 n\mu_a\lVert\mathbf{x}\rVert_\infty$
$\lvert E(\lambda m)\rvert \leq$	$(1-\frac{1}{n})\mu_m\lVert\mathbf{x}\rVert_\infty$	$\frac{1}{2}\log_2 n\mu_m\lVert\mathbf{x}\rVert_\infty$
$\lvert \operatorname{var}(\lambda a)\rvert \leq$	$(\frac{n-1}{n^2})\sigma_a^2\lVert\mathbf{x}\rVert_\infty^2$	$\log_2 n\sigma_a^2\lVert\mathbf{x}\rVert_\infty^2$
$\lvert \operatorname{var}(\lambda m)\rvert \leq$	$\frac{1}{n^2}\sigma_m^2\lVert\mathbf{x}\rVert_\infty^2$	$\frac{1}{4}\log_2 n\sigma_a^2\lVert\mathbf{x}\rVert_\infty^2$

our roundoff error analysis also to the traditional algorithm (pre 1965), which computes the discrete Fourier transform directly from the definition, and we are therefore able to give an indication of which algorithm produces more accurate output in the context of the rounding scheme used. In particular, it will turn out that for very small expected value of the relative error for addition and multiplication, the traditional algorithm will produce more accurate results.

The type of error analysis carried out in this paper is believed to be more useful than those performed previously. Because the bounds are given in terms of the expected values of the relative roundoff errors of addition and multiplication, μ_a and μ_m, they are more sensitive to the particular rounding scheme of a given computer. Moreover, they are in some ways more realistic than worst-case bounds involving the absolute roundoff error ε_M, as these worst-case bounds are often severe overestimates (see Ramos [17]).

Table 4.1 shows the derived bounds on the expected values and variances of the absolute roundoff error for the transform algorithms FFT and TFT due to both addition and multiplication. Several of these results are worthy of special comment.

First of all, it should be noticed that the bounds on the expected value on the accompanying linear forms for both addition and multiplication for the FFT grow like $\log_2 n \cdot \mu$, where μ is the expected value of the relative error for the elementary operation, while for the TFT the bound on the expected value of the accompanying linear form for addition is of the order of $n \cdot \mu_a$, and the bound on the accompanying linear form for multiplication is of the order of μ_m. Therefore we come to the conclusion that the FFT can be considered more accurate than the TFT only if the expected value of the relative error for addition is of the same size or larger than the expected value of the relative error for multiplication. In the case that both expected relative errors are zero, which was actually assumed by Weinstein [24] and by Kanero and Liu [15], a measure of the numerical accuracy of the algorithms is given by the variance of the absolute error.

It is also interesting to notice that while for the FFT the bounds on the variance for the accompanying linear forms for both addition and multiplication are of the order of $\log_2 n \cdot \sigma^2$, where σ^2 is the variance of the relative error for the operation considered, for the TFT the bound on the variance for the

accompanying linear form for addition is of the order of $\frac{1}{n} \cdot \sigma_a^2$ and the bound on the variance for the accompanying linear form for multiplication is of the order of $\frac{1}{n^2} \cdot \sigma_m^2$. Therefore, in case the rounding scheme used has zero mean relative error for both addition and multiplication, as assumed by Kanero and Liu [15] and in Weinstein [24], the TFT should be considered more accurate than the FFT in the sense that the absolute error in the output is expected to deviate less from the zero mean.

The bounds in Table 4.1 were obtained by assuming that the relative roundoff errors introduced at each step are mutually independent and independent of the relative roundoff errors introduced at previous steps, and by neglecting second-order effects. Both these hypotheses should be tested numerically, given a particular rounding scheme. For example, the $E(\lambda a)$ term ignores terms of the order μ_a^2, and as long as μ_a^2 is smaller than σ_a^2 the contribution of the neglected terms will not be significant. However, should the rounding scheme used have larger μ_a, the contribution of the quadratic terms in the local errors need be considered in order to have an accurate roundoff error analysis.

In an attempt to verify our results, we constructed a model that simulates a rounding scheme. This model was defined in §3, and numerical attempts were made to predict the values of μ_a and μ_m. Numerical estimates of μ_a and μ_m were determined experimentally on large samples, for $b = 10$ and different values of T. The numerical experiments consistently indicated the value of the variance for the relative error for both operations to be of the order of 10^{-2T}, which implies that in both cases the standard deviation is about 10^{-T}. The values of the sample mean for addition were so consistent as to suggest that the true value of μ_a is, when $b = 10$, very close to 10^{-T}. It should be emphasized that these numerical results are only valid for the rounding model described in §3.

The numerical results from this rounding model suggest that the hypothesis that the expected values of the local relative errors for addition and multiplication are zero, which was assumed in the two previous statistical analyses of roundoff errors for the Radix-2 FFT (see [15, 24]) may not be valid.

The numerical experiments were performed to verify our results on the FFT when the real and imaginary parts of the data were uniformly distributed in the interval (0,1). The base of the number system was chosen to be 10, the number of digits of the mantissa, T, ranged from 3 to 6, and the size n of the data vector varied from 8 to 1024. While the agreement of the experimental results with the theory was quite good, numerical experiments testing the validity of the theoretical results for different distributions of the data are called for and are in progress.

ACKNOWLEDGMENT

This research was conducted using the Cornell National Supercomputer Facility, a resource of the Center for Theory and Simulation in Science and

Engineering at Cornell University, which is funded in part by the National Science Foundation, New York State, and the IBM Corporation. The author thanks Jon Tolle for his valuable comments and suggestions.

BIBLIOGRAPHY

1. R. C. Agarwal and J. W. Cooley, *Fourier transform and convolution subroutines for the IBM 3090 Vector Facility*, IBM J. Res. Develop. **30** (1986), 145–162.
2. S. Arnold, *The theory of linear models and multivariate analysis*, Wiley, New York, 1981.
3. P. Bloomfield, *Fourier analysis of time series*, Wiley, New York, 1976.
4. D. Calvetti, *A stochastic roundoff error analysis for the FFT*, Doctoral Dissertation, University of North Carolina at Chapel Hill, 1989.
5. J. W. R. Cooley, A. W. Lewis, and P. D. Welsh, *The Fast Fourier Transform and its applications*, Research Paper Rc-1743, IBM Research, 1967.
6. ——, *The Fast Fourier Transform algorithm: Programming considerations in the calculation of sine, cosine, and Laplace transforms*, J. Sound Vibration **12** (1970), 315–337.
7. J. W. R. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp. **19** (1965), 297–301.
8. C. de Boor, *FFT as nested multiplication with a twist*, SIAM J. Sci. Statist. Comput. **1** (1980), 173–178.
9. W. Gander and A. Mazzario, *Numerische Prozeduren*. I (in memoriam Heinz Rutishauser), Ber. Fachgruppe Comput. Wiss., vol. 4, ETH, Zurich, 1972.
10. W. M. Gentleman and G. Sande, *Fast Fourier Transforms—For fun and profit*, Fall Joint Computer Conf., AFIPS Proc., vol. 29, Spartan, Washington, D. C., 1966, pp. 563–578.
11. P. Henrici, *Discrete variable methods in ordinary differential equations*, Wiley, New York, 1962.
12. ——, *A model for the propagation of rounding error in floating arithmetic*, Interval Mathematics, 1980 (Karl L. E. Nickel, ed.), Academic Press, New York, 1980, pp. 49–73.
13. ——, *Essentials of numerical analysis with pocket calculator demonstrations*, Wiley, New York, 1982.
14. ——, *Applied and computational complex analysis*, vol. 3, Wiley, New York, 1986.
15. T. Kanero and B. Liu, *Accumulation of roundoff error in fast Fourier transforms*, J. Assoc. Comput. Mach. **17** (1970), 637–654.
16. G. Merz, *Fast Fourier transform algorithm with applications*, Computational Aspects of Complex Analysis (H. Werner et al., eds.), Reidel, Dordrecht, 1983, pp. 249–278.
17. G. U. Ramos, *Roundoff error analysis of the Fast Fourier Transform*, Math. Comp. **25** (1971), 757–768.
18. R. C. Singleton, *Algorithm* 339. *An Algol procedure for the Fast Fourier Transform with arbitrary factors*, Comm. ACM **11** (1968), 776.
19. F. Stummel, *Forward error analysis of Gaussian elimination*, Numer. Math. **46** (1985), 365–395; 397–415.
20. F. Stummel and K. Heiner, *Praktische Mathematik*, 2nd ed., Teubner, Stuttgart, 1982.
21. C. Temperton, *Self-sorting mixed radix Fast Fourier Transforms*, J. Comput. Phys. **52** (1983), 1–23.
22. M. L. Uhrich, *Fast Fourier Transform without sorting*, IEEE Trans. Audio Electroacoust. **17** (1969), 170–172.
23. P. D. Welsh, *A fixed-point fast Fourier transform error analysis*, IEEE Trans. Audio Electroacoust. **17** (1969), 151–157.

24. C. J. Weinstein, *Roundoff noise in floating point fast Fourier transform computation*, IEEE Trans. Audio Electroacoust. **17** (1969), 209–215.

25. J. H. Wilkinson, *Rounding errors in algebraic processes*, Prentice-Hall, Englewood Cliffs, N. J., 1963.

Department of Mathematics, University of Southern Colorado, Pueblo, Colorado 81001

Current address: Department of Pure and Applied Mathematics, Stevens Institute of Technology, Hoboken, New Jersey 07030

E-mail address: roe-dcalvett@vaxa.stevens-tech.edu