MATHEMATICS OF COMPUTATION, VOLUME 25, NUMBER 116, OCTOBER, 1971

Roundoff Error Analysis of the Fast Fourier Transform

By George U. Ramos

Abstract. This paper presents an analysis of roundoff errors occurring in the floating-point computation of the fast Fourier transform. Upper bounds are derived for the ratios of the root-mean-square (RMS) and maximum roundoff errors in the output data to the RMS value of the output data for both single and multidimensional transformations. These bounds are compared experimentally with actual roundoff errors.

1. Introduction. The fast Fourier transform (FFT) is a very efficient algorithm for computing

$$y(j) = \sum_{k=0}^{N-1} \exp(i2\pi jk/N)x(k) \qquad (j = 0, 1, \cdots, N-1), \tag{1.1}$$

where $x(k)$ is a given set of complex numbers and $i = \sqrt{-1}$. Let $\mathbf{y}' = (y(0), \cdots, y(N-1))$ and $\mathrm{fl}(\mathbf{y})$ be the floating-point representation of $\mathbf{y}$. In this paper, we derive bounds for

$$||\mathrm{fl}(\mathbf{y}) - \mathbf{y}||_{\mathrm{RMS}}/||\mathbf{y}||_{\mathrm{RMS}} \quad \text{and} \quad ||\mathrm{fl}(\mathbf{y}) - \mathbf{y}||_{\infty}/||\mathbf{y}||_{\mathrm{RMS}},$$

where

$$||\mathbf{z}||_{\mathrm{RMS}} = \left(\left(\sum_k |z(k)|^2\right) \Big/ N\right)^{1/2} \quad \text{and} \quad ||\mathbf{z}||_{\infty} = \max_k |z(k)|.$$

These bounds include the effect of roundoff in computing sines and cosines and are obtained for both single and multidimensional transformations. Special consideration is given to cases when N is a multiple of 2 or 4.

The subject of roundoff error in the FFT has been studied and reported by others but with less generality or using a different approach. By comparing upper bounds, Gentleman and Sande [1] show that accumulated floating-point roundoff error is significantly less when one uses the FFT than when one computes (1.1) directly. In [2], Welch derives approximate upper and lower bounds on the RMS error in a fixed-point power-of-two algorithm. Weinstein [3] uses a statistical model for floating-point roundoff errors to predict the output-noise variance. Kaneko and Liu [4] also use a statistical approach and derive expressions for the mean-square error in a floating-point power-of-two transformation.

In the following sections, (1) the FFT algorithm is described in terms of a matrix factorization, (2) error bounds are derived, and (3) experimental comparisons of actual errors with error bounds are presented.

2. Matrix Factorization and the Fast Fourier Transform. In 1958, a matrix factorization for an algorithm similar to the FFT was described in a paper by I. J.

Received March 3, 1970, revised February 19, 1971.
AMS 1969 *subject classifications.* Primary 4225, 6535, 6580.
Key words and phrases. Fast Fourier transform, floating-point, roundoff errors.

Good [15]. This paper led to the 1965 paper of Cooley and Tukey [5] in which the fast Fourier transform was introduced. Since then, many authors have presented matrix factorizations for the FFT. (See, for example, [6], [7], [8], [9], [13], and [14].)

Consider the complex Fourier transform of (1.1). This transform can alternatively be expressed as the matrix-vector equation

$$\mathbf{y} = T\mathbf{x} \tag{2.1}$$

where T is an Nth-order matrix of complex exponentials

$$T(j, k) = \exp(i2\pi jk/N) \qquad (j, k = 0, 1, \cdots, N - 1).$$

If this were to be computed directly, it would require N^2 complex multiplications and $N(N - 1)$ complex additions. Instead, one can make use of the fact that

$$T = P_{M+1}(B_M P_M)(D_{M-1}B_{M-1}P_{M-1}) \cdots (D_1 B_1 P_1), \tag{2.2}$$

where P_l ($l = 1, 2, \cdots, M + 1$) are permutation matrices, D_l ($l = 1, 2, \cdots, M - 1$) are diagonal matrices of complex exponential elements (called rotation factors by Singleton [13], twiddle factors by Gentleman and Sande [1]), and B_l ($l = 1, 2, \cdots, M$) are block-diagonal matrices whose blocks on the diagonal are identical square submatrices, each the matrix of a complex Fourier transform of dimension N_l. In this case, the required number of operations is reduced to $N(M - 1 + \sum_{l=1}^{M} N_l)$ complex multiplications and $N(\sum_{l=1}^{M} (N_l - 1))$ complex additions.

Description of the fast Fourier transform as a matrix factorization simplifies the following roundoff error analysis; but, before proceeding, a few remarks are in order. First, most factored-matrix formulations include only a single permutation matrix. The additional permutation matrices of (2.2) preserve diagonality. Second, the factors $D_l B_l$ can be combined. The required multiplication count would then be $N(\sum_{l=1}^{M} N_l)$. Third, the above operation counts do not take into account the fact that many of the complex exponentials are ± 1 or $\pm\sqrt{-1}$. And fourth, (2.2) holds for the Sande-Tukey algorithm as well as the Cooley-Tukey algorithm but with different diagonal matrices, D_l (see [1]).

3. Roundoff Errors in the Fast Fourier Transform. In this section, we first explain the roundoff error models used and then state and prove a theorem bounding the RMS and maximum errors.

We use models due to Wilkinson to mathematically represent the effect of roundoff in floating-point arithmetic operations. In [10] Wilkinson shows that the floating-point sum and floating-point product of two floating-point numbers a and b can always be expressed as

$$\mathrm{fl}(a + b) = a(1 + \theta\epsilon) + b(1 + \theta\epsilon), \tag{3.1}$$

and

$$\mathrm{fl}(ab) = ab(1 + \theta\epsilon), \tag{3.2}$$

where ϵ is a computer-dependent constant and θ is a generic variable usually different in value at each occurrence but always within the range -1 to 1. (The error constant, ϵ, is $0.5\beta^{1-t}$ for rounded operations or β^{1-t} for chopped operations, where β is the

number base of the floating-point computing system, t is the number of base-β digits in the mantissa of the floating-point number, and at least $t + 1$ digits are used to accumulate sums. For example, $\epsilon = 16^{-5}$ in short-precision floating-point operations on the IBM 360. If sums are accumulated with only t digits, $\epsilon = 0.5(1 + 1/\beta)\beta^{1-t}$ for rounded operations or $(1 + 1/\beta)\beta^{1-t}$ for chopped operations.)

To represent roundoff in computing sines and cosines, we could model either the relative error or the absolute error. If the arguments of the sines and cosines are carefully computed, either model will result in approximately the same bound. However, small errors in the arguments can result in large relative error when the sine or cosine is very small. In this case, an absolute error model gives a better bound. Hence we introduce an absolute error constant, $\gamma \geqq 0$, such that

$$\text{fl}(\sin(\text{fl}(a))) = \sin(a) + \gamma\theta\epsilon \quad \text{and} \quad \text{fl}(\cos(\text{fl}(a))) = \cos(a) + \gamma\theta\epsilon,$$

where θ and ϵ are above. This constant depends on how sines and cosines and their arguments are computed for a transformation of a given order, but it is independent of the input data.

Let $\mathbf{x}' = (x(0), \cdots, x(N-1))$, $\mathbf{y}' = (y(0), \cdots, y(N-1))$ and $\text{fl}(\mathbf{y})$ be the floating-point representation of $\mathbf{y}$ and let

$$||\mathbf{z}||_{\text{RMS}} = \left(\left(\sum_k |z(k)|^2\right) \Big/ N\right)^{1/2} \quad \text{and} \quad ||\mathbf{z}||_\infty = \max_k |z(k)|.$$

Then we have the following:

THEOREM 1. *If* $\mathbf{y} = T\mathbf{x}$ *is computed by a floating-point fast Fourier transform of order* $N = N_1 N_2 \cdots N_M$, *then*

a. $||\text{fl}(\mathbf{y}) - \mathbf{y}||_{\text{RMS}}/||\mathbf{y}||_{\text{RMS}} < K(N, \gamma)\epsilon + O(\epsilon^2)$ *and*

b. $||\text{fl}(\mathbf{y}) - \mathbf{y}||_\infty/||\mathbf{y}||_{\text{RMS}} < \sqrt{N}\, K(N, \gamma)\epsilon + O(\epsilon^2)$, *where*

$$K(N, \gamma) = \sum_{l=1}^{M} \alpha(N_l) + (M - 1)(3 + 2\gamma)$$

and

$$\begin{aligned} \alpha(N_l) &= \sqrt{2} && (N_l = 2), \\ &= 5 && (N_l = 4), \\ &= 2\sqrt{N_l}\,(N_l + \gamma) && \textit{otherwise.} \end{aligned}$$

(*In the case of radix-2 or radix-4 algorithms the important constants are*

$$K(2^M, \gamma) = (3 + \sqrt{2} + 2\gamma)M - (3 + 2\gamma), \quad \textit{and}$$

$$K(4^M, \gamma) = (8 + 2\gamma)M - (3 + 2\gamma).$$

These constants include the effect of separate multiplications by D_l $(l = 1, 2, \cdots, M - 1)$.)

Proof of a. First consider computation of the inner product $v = \sum_{l=1}^{n} a(l)u(l)$ by the algorithm: *begin* $v := a(1) \otimes u(1)$; *for* $l := 2$ *step* 1 *until* n *do* $v := v + a(l) \otimes u(l)$ *end* where it is known that $\mathbf{u}$ is exactly representable in floating point while $\mathbf{a}$ satisfies

$\mathrm{fl}(a(l)) = a(l) + \gamma\theta\epsilon$ $(l = 1, 2, \cdots, n)$ for γ, θ and ϵ as above. By repeated application of (3.1) and (3.2), as in Wilkinson [10], one finds that

$$\mathrm{fl}(v) = (a(1) + \gamma\theta\epsilon)u(1)(1 + \theta\epsilon)^n + (a(2) + \gamma\theta\epsilon)u(2)(1 + \theta\epsilon)^n$$
$$+ (a(3) + \gamma\theta\epsilon)u(3)(1 + \theta\epsilon)^{n-1} + \cdots + a(n)(1 + \gamma\theta\epsilon)u(n)(1 + \theta\epsilon)^2.$$

Expanding factors $(1 + \theta\epsilon)^l$ and regrouping terms, this becomes

$$\mathrm{fl}(v) = v + \epsilon[(a(1)n\theta + \gamma\theta)u(1) + (a(2)n\theta + \gamma\theta)u(2)$$
$$+ (a(3)(n - 1)\theta + \gamma\theta)u(3) + \cdots + (a(n)2\theta + \gamma\theta)u(n)] + O(\epsilon^2),$$

where $O(\epsilon^2)$ includes all terms of order ϵ^2. Thus, it follows that floating-point computation of the matrix vector product $\mathbf{v} = A\mathbf{u}$, where $\mathrm{fl}(A(j, l)) = A(j, l) + \gamma\theta\epsilon$ and $\mathrm{fl}(u(l)) = u(l)$, is given exactly by

(3.3)

$$\begin{bmatrix} \mathrm{fl}(v(1)) \\ \mathrm{fl}(v(2)) \\ \vdots \\ \mathrm{fl}(v(n)) \end{bmatrix} = \begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(n) \end{bmatrix}$$
$$+ \epsilon \begin{bmatrix} A(1,1)n\theta + \gamma\theta & A(1,2)n\theta + \gamma\theta & \cdots & A(1,n)2\theta + \gamma\theta \\ A(2,1)n\theta + \gamma\theta & A(2,2)n\theta + \gamma\theta & \cdots & A(2,n)2\theta + \gamma\theta \\ \vdots & \vdots & & \vdots \\ A(n,1)n\theta + \gamma\theta & A(n,2)n\theta + \gamma\theta & \cdots & A(n,n)2\theta + \gamma\theta \end{bmatrix} \begin{bmatrix} u(1) \\ u(2) \\ \vdots \\ u(n) \end{bmatrix} + \begin{bmatrix} O(\epsilon^2) \\ O(\epsilon^2) \\ \vdots \\ O(\epsilon^2) \end{bmatrix}.$$

Next, consider computation of (1.1) without using the FFT. We write this complex computation as its real equivalent:

$$\left[\begin{array}{c} \mathbf{y}_R \\ \hline \mathbf{y}_I \end{array}\right] = \left[\begin{array}{c|c} C & -S \\ \hline S & C \end{array}\right] \left[\begin{array}{c} \mathbf{x}_R \\ \hline \mathbf{x}_I \end{array}\right],$$

where C and S are real matrices with elements $C(j, k) = \cos(2\pi(j - 1)(k - 1)/N)$ and $S(j, k) = \sin(2\pi(j - 1)(k - 1)/N)$ $(j, k = 1, 2, \cdots, N)$, and $\mathbf{x}_R$, $\mathbf{x}_I$, $\mathbf{y}_R$, $\mathbf{y}_I$ are the real and imaginary parts of $\mathbf{x}$ and $\mathbf{y}$. Note that the RMS value of a complex vector is $\sqrt{2}$ times as large as the RMS value of its real equivalent and that the RMS value of any vector is a multiple of the Euclidean norm and therefore is consistent with the same matrix norms as the Euclidean norm. [I.e., if $\mathbf{v} = A\mathbf{u}$, then $||v||_{\mathrm{RMS}} \leqq ||A||\,||u||_{\mathrm{RMS}}$, where $||A||$ is the Frobenius norm (the square root of the sum of the squared magnitudes of all elements) or the spectral norm (the square root of the largest eigenvalue of A^*A). See Wilkinson [10] or Isaacson and Keller [11].] Therefore, by (3.3) and the properties of norms,

$$||\mathrm{fl}(\mathbf{y}) - \mathbf{y}||_{\mathrm{RMS}} \leqq \epsilon\,||M||\,||\mathbf{x}||_{\mathrm{RMS}} + O(\epsilon^2), \tag{3.4}$$

where M is the matrix of Fig. 1. Using the fact that $|C(j, k)|^2 + |S(j, k)|^2 = 1$, the

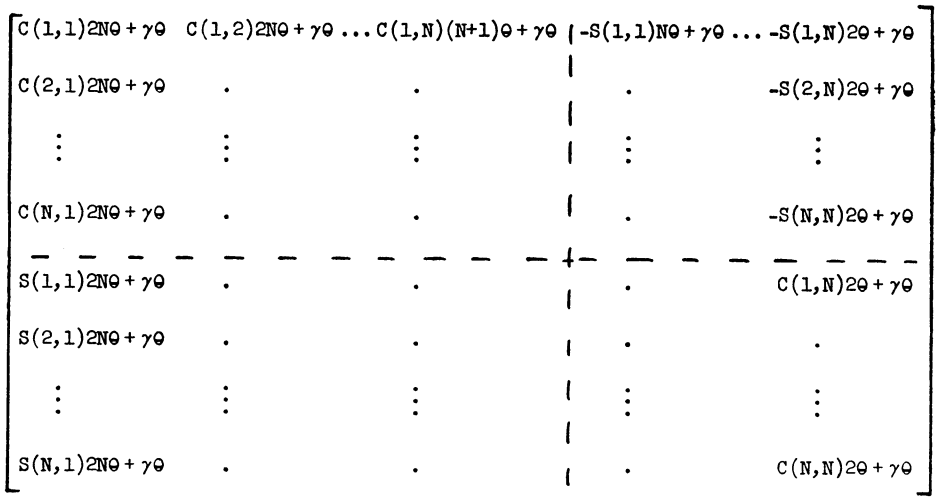

FIGURE 1. *Direct Transformation Error Matrix*

Frobenius norm of M is bounded by

$$(3.5)\qquad \begin{aligned} ||M|| &\leqq \{N[(2N)^2 + (2N)^2 + (2N-1)^2 + \cdots + 3^2 + 2^2]\}^{1/2} + 2N\gamma \\ &< 2N(N+\gamma) \end{aligned}$$

when N is greater than 2.

Finally, we analyze the fast Fourier transform. Let $\mathbf{z}_1 = D_1B_1P_1\mathbf{x}$. Since the permutation matrix simply reorders vector values, it introduces no roundoff error. Assume $\mathrm{fl}(\mathbf{x}) = \mathbf{x}$. Then

$$(3.6)\qquad \begin{aligned} \mathrm{fl}(\mathbf{z}_1) - \mathbf{z}_1 &= \mathrm{fl}(D_1\,\mathrm{fl}(B_1P_1\mathbf{x})) - D_1B_1P_1\mathbf{x} \\ &= \mathrm{fl}(D_1\,\mathrm{fl}(\mathbf{v})) - D_1\,\mathrm{fl}(\mathbf{v}) + D_1[\mathrm{fl}(B_1\mathbf{u}) - B_1\mathbf{u}], \end{aligned}$$

where $\mathbf{u} = P_1\mathbf{x}$ and $\mathbf{v} = B_1\mathbf{u}$. To bound $\mathrm{fl}(B_1\mathbf{u}) - B_1\mathbf{u}$, recall that B_1 is a block-diagonal matrix whose blocks are Fourier transform matrices of order N_1. Let $\mathbf{u}_l$, $\mathbf{v}_l$ $(l = 1, 2, \cdots, N/N_1)$ be N_1-vectors such that

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{N/N_1} \end{bmatrix} \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{N/N_1} \end{bmatrix}.$$

Then by (3.4) and (3.5), $||\mathrm{fl}(\mathbf{v}_l) - \mathbf{v}_l||_{\mathrm{RMS}} \leqq \epsilon 2N_1(N_1 + \gamma)||\mathbf{u}_l||_{\mathrm{RMS}} + O(\epsilon^2)$ $(l = 1, 2, \cdots, N/N_1)$ when N_1 is greater than 2. If $N_1 = 2$, this inequality still holds. In fact, we can do much better. Figs. 2 and 3 show the block-diagonal factor matrices

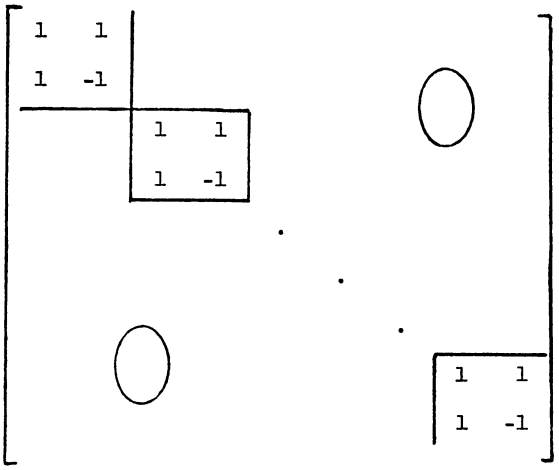

FIGURE 2. *The Block-Diagonal Factor Matrix with 2nd-Order Blocks*

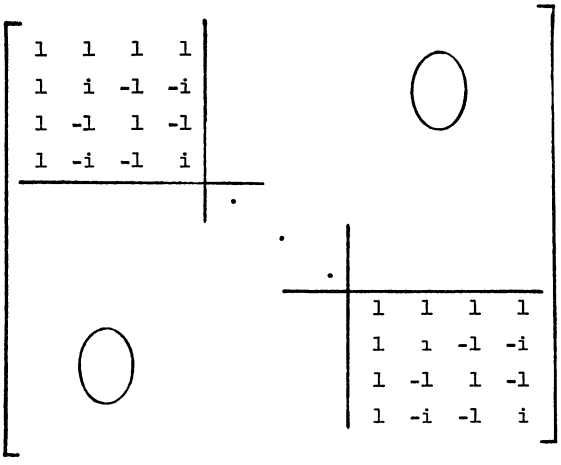

FIGURE 3. *The Block-Diagonal Factor Matrix with 4th-Order Blocks*

for the cases when N has factors 2 or 4. By inspection, one can see that in these cases no sines and cosines are computed, no multiplications are required, and there are only N_1 elements to be summed as compared with $2N_1 - 1$ in other cases. Thus, one can easily show that

$$||\mathrm{fl}(\mathbf{v}_l) - \mathbf{v}_l||_{\mathrm{RMS}} \leqq \epsilon \sqrt{N_1}\, \alpha(N_1)\, ||\mathbf{u}_l||_{\mathrm{RMS}} + O(\epsilon^2) \qquad (l = 1, 2, \cdots, N/N_1),$$

where

$$\begin{aligned} \alpha(N_1) &= \sqrt{2} && (N_1 = 2), \\ &= 5 && (N_1 = 4), \\ &= 2\sqrt{N_1}\,(N_1 + \gamma) && \text{otherwise.} \end{aligned}$$

It immediately follows that

$$||\mathrm{fl}(B_1\mathbf{u}) - B_1\mathbf{u}||_{\mathrm{RMS}} \leqq \epsilon\sqrt{N_1}\,\alpha(N_1)\,||\mathbf{u}||_{\mathrm{RMS}} + O(\epsilon^2) \tag{3.7}$$

for $\alpha(N_1)$ as above.

In the same way, we obtain a bound on the error in multiplication by the complex-diagonal matrix D_1. The bound is given by

$$||\mathrm{fl}(D_1\,\mathrm{fl}(\mathbf{v})) - D_1\,\mathrm{fl}(\mathbf{v})||_{\mathrm{RMS}} \leqq \epsilon(2\sqrt{2} + 2\gamma)\,||\mathrm{fl}(\mathbf{v})||_{\mathrm{RMS}} + O(\epsilon^2). \tag{3.8}$$

From (3.7) it follows that $||\mathrm{fl}(\mathbf{v})||_{\mathrm{RMS}} = ||\mathbf{v}||_{\mathrm{RMS}} + O(\epsilon)$. Furthermore, the spectral norms of D_1, B_1, and P_1 are 1, $\sqrt{N_1}$, and 1, respectively, since $D_1^*D_1 = I$, $B_1^*B_1 = N_1I$ and $P_1^*P_1 = I$, where I is the N-by-N identity matrix. So from (3.6), (3.7) and (3.8), we get

$$||\mathrm{fl}(\mathbf{z}_1) - \mathbf{z}_1||_{\mathrm{RMS}} \leqq \epsilon\sqrt{N_1}\,(\alpha(N_1) + 3 + 2\gamma)\,||\mathbf{x}||_{\mathrm{RMS}} + O(\epsilon^2),$$

where $\alpha(N_1)$ is given above.

The next step is to let $\mathbf{z}_2 = D_2B_2P_2\mathbf{z}_1$. Then

$$\mathrm{fl}(\mathbf{z}_2) - \mathbf{z}_2 = \mathrm{fl}(D_2B_2P_2\,\mathrm{fl}(\mathbf{z}_1)) - D_2B_2P_2\,\mathrm{fl}(\mathbf{z}_1) + D_2B_2P_2[\mathrm{fl}(\mathbf{z}_1) - \mathbf{z}_1]$$

and

$$||\mathrm{fl}(\mathbf{z}_2) - \mathbf{z}_2||_{\mathrm{RMS}} \leqq \epsilon(N_1N_2)^{1/2}(\alpha(N_1) + \alpha(N_2) + 2(3 + 2\gamma))\,||\mathbf{x}||_{\mathrm{RMS}} + O(\epsilon^2).$$

The proof of part a is completed by continuing in this manner and using (2.2).

Proof of b. The proof is extremely simple. Let $e(j) = \mathrm{fl}(y(j)) - y(j)$. Then

$$\max_j |e(j)|^2 \leqq \sum_{j=0}^{N-1} |e(j)|^2,$$

from which it follows that

$$\max_j |e(j)| \leqq \sqrt{N}\,||\mathbf{e}||_{\mathrm{RMS}}.$$

Substituting the bound of part a for $||\mathbf{e}||_{\mathrm{RMS}}$ completes the proof.

It is not necessary to obtain a bound on the maximum error by using part a. Instead, one can use matrix infinity norms in the same fashion that matrix spectral norms were used above. But the infinity norms of the factor matrices, B_l, are proportional to N_l rather than $\sqrt{N_l}$, and so a higher bound results.

4. Roundoff Errors in Multidimensional Transformations. The efficiency of the fast Fourier transform has made it economically feasible to compute higher-dimensional Fourier transformations in applications such as picture processing and x-ray diffraction studies. In this section, bounds on roundoff errors in multidimensional FFT's are derived.

The problem is to bound roundoff errors in computing

$$Y(t_1, t_2, \cdots, t_m) = \sum_{s_1} \sum_{s_2} \cdots \sum_{s_m} \mathrm{e}(s_1t_1/N_1 + s_2t_2/N_2 + \cdots + s_mt_m/N_m)\,X(s_1, s_2, \cdots, s_m) \tag{4.1}$$

$$(s_l, t_l = 0, 1, \cdots, N_l - 1;\ l = 1, 2, \cdots, m).$$

Let

$$E(t_1, t_2, \cdots, t_m) = \mathrm{fl}(Y(t_1, t_2, \cdots, t_m)) - Y(t_1, t_2, \cdots, t_m),$$

$$[\mathrm{fl}(Y) - Y]_{\mathrm{RMS}} = \left\{\left[\sum_{t_1}\sum_{t_2}\cdots\sum_{t_m}|E(t_1, t_2, \cdots, t_m)|^2\right]/N_1N_2\cdots N_m\right\}^{1/2},$$

and

$$[\mathrm{fl}(Y) - Y]_{\mathrm{MAX}} = \max_{t_1, t_2, \ldots, t_m}|E(t_1, t_2, \cdots, t_m)|.$$

Then we have

THEOREM 2. *The RMS and maximum error due to roundoff in a multidimensional fast Fourier transform are bounded by*

a. $[\mathrm{fl}(Y) - Y]_{\mathrm{RMS}}/Y_{\mathrm{RMS}} \leqq \epsilon \sum_{l=1}^{m} K(N_l, \gamma) + O(\epsilon^2)$ *and*

b. $[\mathrm{fl}(Y) - Y]_{\mathrm{MAX}}/Y_{\mathrm{RMS}} \leqq \epsilon(N_1N_2\cdots N_m)^{1/2}\sum_{l=1}^{m} K(N_l, \gamma) + O(\epsilon^2)$, *where* $K(N_l, \gamma)$ $(l = 1, 2, \cdots, m)$ *is the error constant given in Theorem* 1.

Proof. Let (4.1) be rewritten as the system of equations

$$Z_{l-1}(s_1, \cdots, s_{l-1}, t_l, \cdots, t_m) = \sum_{s_l} \mathrm{e}(s_lt_l/N_l)Z_l(s_1, \cdots, s_l, t_{l+1}, \cdots, t_m) \qquad (l = 1, 2, \cdots, m)$$

with $Z_0 = Y$ and $Z_m = X$, and describe this system of equations by the notation

$$Z_{l-1} = T_lZ_l \qquad (l = 1, 2, \cdots, m).$$

Then by adding and subtracting identical terms to the equation

$$\mathrm{fl}(Y) - Y = \mathrm{fl}(T_1\,\mathrm{fl}(T_2\cdots\mathrm{fl}(T_mX)\cdots)) - T_1T_2\cdots T_mX,$$

one gets

$$\begin{aligned}\mathrm{fl}(Y) - Y = {} & \mathrm{fl}(T_1\,\mathrm{fl}(Z_1)) - T_1\,\mathrm{fl}(Z_1)\\ & + T_1\,\mathrm{fl}(T_2\,\mathrm{fl}(Z_2)) - T_1T_2\,\mathrm{fl}(Z_2)\\ & + T_1T_2\,\mathrm{fl}(T_3\,\mathrm{fl}(Z_3)) - T_1T_2T_3\,\mathrm{fl}(Z_3)\\ & + \cdots + T_1T_2\cdots T_{m-1}\,\mathrm{fl}(T_mX) - T_1T_2\cdots T_mX.\end{aligned}$$

Now take the RMS value of both sides and use the Cauchy-Schwarz inequality to get

$$\begin{aligned}[\mathrm{fl}(Y) - Y]_{\mathrm{RMS}} \leqq {} & [\mathrm{fl}(T_1\,\mathrm{fl}(Z_1)) - T_1\,\mathrm{fl}(Z_1)]_{\mathrm{RMS}}\\ & + [T_1[\mathrm{fl}(T_2\,\mathrm{fl}(Z_2)) - T_2\,\mathrm{fl}(Z_2)]]_{\mathrm{RMS}}\\ & + \cdots + [T_1T_2\cdots T_{m-1}[\mathrm{fl}(T_mX) - T_mX]]_{\mathrm{RMS}}.\end{aligned} \tag{4.2}$$

Using Theorem 1, it is not difficult to prove that

$$[\mathrm{fl}(Z_{l-1}) - Z_{l-1}]_{\mathrm{RMS}}/[Z_l]_{\mathrm{RMS}} \leqq \epsilon\sqrt{N_l}\,K(N_l, \gamma) + O(\epsilon^2). \tag{4.3}$$

Nor is it difficult to prove

$$[Z_{l-1}]_{\mathrm{RMS}} = \sqrt{N_l}\,[Z_l]_{\mathrm{RMS}}. \tag{4.4}$$

Therefore, by (4.2), (4.3) and (4.4),

$$[\mathrm{fl}(Y) - Y]_{\mathrm{RMS}} \leqq \epsilon\{(N_1)^{1/2} K(N_1, \gamma)[\mathrm{fl}(Z_1)]_{\mathrm{RMS}} + (N_1 N_2)^{1/2} K(N_2, \gamma)[\mathrm{fl}(Z_2)]_{\mathrm{RMS}} + \cdots + (N_1 N_2 \cdots N_m)^{1/2} K(N_m, \gamma)[\mathrm{fl}(X)]_{\mathrm{RMS}}\} + O(\epsilon^2).$$

But by (4.3), $[\mathrm{fl}(Z_l)]_{\mathrm{RMS}} = [Z_l]_{\mathrm{RMS}} + O(\epsilon)$ ($l = 1, 2, \cdots, m - 1$), and by (4.4), $[Z_l]_{\mathrm{RMS}} = (N_{l+1} N_{l+2} \cdots N_m)^{1/2} [X]_{\mathrm{RMS}}$. Assuming that $\mathrm{fl}[(X)]_{\mathrm{RMS}} = [X]_{\mathrm{RMS}}$, or at least $[\mathrm{fl}(X)]_{\mathrm{RMS}} = [X]_{\mathrm{RMS}} + O(\epsilon)$, the proof of part a is complete.

Part b is proved by arguments identical to those used in the proof of part b of Theorem 1.

5. Experimental Results. Roundoff error bounds are always pessimistic—sometimes so much so that they give no indication of the true error. To find out how pessimistic the error bounds of Section 3 are, the following experiment was performed. Using two different FORTRAN programs, one by N. M. Brenner [12] and the other by R. C. Singleton [13], a mixed radix fast Fourier transform of Gaussian data with mean 0 and variance 2 was computed in both short and long precision on the Stanford IBM 360/67. The actual error was computed as the difference between the short precision results and the truncated long precision results. The constant γ, used in determining the error bound, was computed by taking the difference between short precision and truncated long precision numbers representing sines and cosines. The results of this experiment are given in Table 1. Note that the RMS error bound is roughly 20 times larger than the RMS error and the MAX error bound is roughly 2 orders of magnitude larger than the MAX error. Also note the relative size of the error bounds with respect to values of the transformed data. Even though the bounds are pessimistic, they might be used as a threshold for deciding what confidence to place in transformed data of relatively small magnitude.

6. Conclusion. In the preceding sections, roundoff errors in the floating-point fast Fourier transform have been analyzed. Bounds on RMS and maximum errors in transformed data were determined for both single and multidimensional transforms, and in the case of a one-dimensional transform results of a computational experiment show how close these bounds are to the actual roundoff errors. The bounds include the effect of roundoff in computing sines and cosines and, if contributions to the actual errors are in the same proportion as to the error bounds, a close look at the error bounds shows that the effect of roundoff in computing sines and cosines is not negligible but in fact contributes the same order of magnitude to the total error as the roundoff in additions and multiplications.

So far, nothing has been said about floating-point representation of input data. It was assumed that these numbers were exactly representable in machine precision. If not, an additional term must be added to the roundoff error to account for rounding input data. Suppose $\mathrm{fl}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\delta}$. Then the additional term is

$$||T\boldsymbol{\delta}||_{\mathrm{RMS}} \leqq \sqrt{N}\, ||\boldsymbol{\delta}||_{\mathrm{RMS}}.$$

On the other hand, suppose that the input data is known to a number of significant digits fewer than that of machine precision. For example, the data might have come from an analog device of limited accuracy. Then the bounds on roundoff error can

Table 1

Comparison of Actual Errors with Error Bounds

Order of Transform and Factorization	Values of Transformed Data				Errors in Transformed Data		A Priori Bounds on Errors	
	MIN	RMS	MAX	γ	RMS	MAX	RMS	MAX
128 = 4 4 4 2 *	0.9543	16.54	36.13	3.1	0.000032	0.000082	0.000698	0.007897
128 = 4 2 2 2 4**	0.9543	16.54	36.13	1.7	0.000026	0.000064	0.000631	0.007138
256 = 4 4 4 4 *	1.4436	21.78	53.48	3.1	0.000047	0.000153	0.000992	0.015875
256 = 4 4 4 4 **	1.4436	21.78	53.48	4.7	0.000070	0.000216	0.001187	0.018992
512 = 4 4 4 4 2*	1.4158	31.20	81.04	4.2	0.000101	0.000306	0.001994	0.045121
512 = 4 4 2 4 4**	1.4158	31.20	81.04	4.6	0.000106	0.000307	0.002083	0.047141
1024 = 4 4 4 4 4*	2.2109	44.38	130.41	9.3	0.000202	0.000648	0.004720	0.151041
1024 = 4 4 4 4 4**	2.2110	44.38	130.41	8.9	0.000291	0.001163	0.004572	0.146301
100 = 4 5 5 *	1.5535	14.98	29.17	5.2	0.000129	0.000491	0.001755	0.017554
100 = 5 4 5 **	1.5534	14.98	29.17	7.7	0.000043	0.000122	0.002218	0.022176
200 = 4 2 5 5 *	1.3670	19.50	45.60	6.8	0.000175	0.000560	0.003014	0.042628
200 = 5 2 2 2 5**	1.3670	19.50	45.60	3.4	0.000046	0.000109	0.002223	0.031432
300 = 4 3 5 5 *	0.6539	23.64	54.42	8.1	0.000239	0.000663	0.004905	0.084952
300 = 5 2 3 2 5**	0.6539	23.64	54.42	7.0	0.000098	0.000301	0.004802	0.083172
400 = 4 4 5 5 *	2.8367	27.50	66.63	7.1	0.000243	0.000743	0.004440	0.088793
400 = 4 5 5 4 **	2.8368	27.50	66.63	7.7	0.000120	0.000430	0.004685	0.093692

Table 1 (continued)

Order of Transform and Factorization	Values of Transformed Data				Errors in Transformed Data		A Priori Bounds on Errors	
	MIN	RMS	MAX	γ	RMS	MAX	RMS	MAX
125 = 5 5 5 *	0.6356	16.42	37.31	4.2	0.000161	0.000558	0.002304	0.025765
125 = 5 5 5 **	0.6355	16.42	37.31	3.1	0.000037	0.000085	0.001998	0.022333
243 = 3 3 3 3 3*	0.0957	21.24	53.30	4.1	0.000171	0.000424	0.003379	0.052672
243 = 3 3 3 3 3**	0.0957	21.24	53.30	9.0	0.000089	0.000323	0.005911	0.092140
343 = 7 7 7 *	0.9315	25.67	60.21	7.4	0.000160	0.000536	0.006487	0.120138
343 = 7 7 7 **	0.9315	25.67	60.21	7.9	0.000123	0.000384	0.006700	0.124080

* Brenner's Program

** Singleton's Program

be used in reverse as suggested by the following: Let the roundoff error be given exactly by the complex N-vector $\mathbf{e}$. This vector can be considered the exact solution of the equation $\mathbf{e} = T\boldsymbol{\delta}$ for some fictional $\boldsymbol{\delta}$ bounded by

$$||\boldsymbol{\delta}||_{\text{RMS}} = ||\mathbf{e}||_{\text{RMS}}/\sqrt{N} \leqq \epsilon K(N, \gamma)\ ||\mathbf{x}||_{\text{RMS}} + O(\epsilon^2),$$

and

$$||\boldsymbol{\delta}||_{\infty} \leqq \epsilon\sqrt{N}\ K(N, \gamma)\ ||\mathbf{x}||_{\text{RMS}} + O(\epsilon^2).$$

If it should turn out that $\epsilon\sqrt{N}K(N, \gamma)||\mathbf{x}||_{\text{RMS}}$ is smaller than the least significant digit of the input data, the roundoff error is negligible.

Acknowledgments. The author wishes to thank Professor Gene Golub of Stanford University for his advice and encouragement during the research for this paper. Special thanks also go to GTE Sylvania Incorporated for support received while the author was at Stanford. This research was done in partial fulfillment of the requirements for a doctoral degree in Computer Science.

GTE Sylvania Incorporated
Mountain View, California 94040

1. W. M. GENTLEMAN & G. SANDE, *Fast Fourier Transforms—For Fun and Profit,* Fall Joint Computer Conference AFIPS Proc., 1966, v. 29, Spartan, Washington, D.C., 1966, pp. 563–578.
2. P. D. WELCH, "A fixed-point fast Fourier transform error analysis," *IEEE Trans. Audio and Electroacoustics,* v. AU-17, 1969, pp. 151–157.
3. C. J. WEINSTEIN, "Roundoff noise in floating point fast Fourier transform computation," *IEEE Trans. Audio and Electroacoustics,* v. AU-17, 1969, pp. 209–215.
4. T. KANEKO & B. LIU, "Accumulation of roundoff error in fast Fourier transforms," *J. ACM,* v. 17, 1970, pp. 637–654.
5. J. W. COOLEY & J. W. TUKEY, "An algorithm for the machine calculation of complex Fourier series," *Math. Comp.,* v. 19, 1965, pp. 297–301. MR **31** #2843.
6. W. M. GENTLEMAN, "Matrix multiplication and fast Fourier transforms," *Bell System Tech. J.,* v. 47, 1968, pp. 1099–1103.
7. R. C. SINGLETON, "On computing the fast Fourier transform," *Comm. ACM,* v. 10, 1967, pp. 647–654. MR **39** #2362.
8. F. THEILHEIMER, "A matrix version of the fast Fourier transform," *IEEE Trans. Audio and Electroacoustics,* v. AU-17, 1969, pp. 158–161.
9. D. K. KAHANER, *Matrix Description of the Fast Fourier Transform,* Los Alamos Scientific Laboratory Report LA-4275-MS, 1969.
10. J. H. WILKINSON, *Rounding Errors in Algebraic Processes,* Prentice-Hall, Englewood Cliffs, N.J., 1963. MR **28** #4661.
11. E. ISAACSON & H. B. KELLER, *Analysis of Numerical Methods, Wiley,* New York, 1966. MR **34** #924.
12. N. M. BRENNER, *Three FORTRAN Programs that Perform the Cooley-Tukey Fourier Transform,* Technical Note 1967-2, M.I.T. Lincoln Lab., Lexington, Mass., 1967.
13. R. C. SINGLETON, "An algorithm for computing the mixed radix fast Fourier transform," *IEEE Trans. Audio and Electroacoustics,* v. AU-17, 1969, pp. 93–103.
14. M. C. PEASE, "An adaptation of the fast Fourier transform for parallel processing," *J. ACM,* v. 15, 1968, pp. 252–264.
15. I. J. GOOD, "The interaction algorithm and practical Fourier analysis," *J. Roy. Statist. Soc. Ser. B,* v. 20, 1958, pp. 361–372. MR **21** #1674.