論　文

UDC 517.443-37:[681.322:681.325.538]

浮動小数点FFTの精度改善と誤差評価

正員　本間　仁志†　　正員　佐川　雅彦†

On Improving the Accuracy and Error Analysis in Floating Point FFT Computation

Hitoshi HONMA† *and* Masahiko SAGAWA†, *Regular Members*

あらまし　FFTの誤差原因は，入力データの量子化誤差，四則演算による丸め誤差，$W^{nk}=\exp(-j2\pi nk/N)$の計算誤差に大別できる．その中で，W^{nk}の誤差表現は複雑であり，従来の誤差解析では，その誤差を有限ビットによる打ち切り誤差で近似している．そこで，本論文では，実際のW^{nk}の計算精度とその影響について検討する．その結果，W^{nk}の計算誤差は，有限ビットによる打ち切り誤差に比べ，はるかに大きく，アルゴリズムを不安定にしていることが分った．次に，本論文では，これらの問題を解決するために，W^{nk}の計算精度を改善する方法を提案する．提案する方法は，FFTの直接計算に比べ，処理時間，メモリをほとんど増すことなしに実行できる．この処理によって，従来の誤差解析の信頼性が向上し，さらに，FFTに伴う誤差を軽減することができる．最後に，本理論の有効性を，幾つかの例題で確認している．

1. まえがき

離散フーリエ変換（Discrete Fourier Transform, 以下DFTと略記する）を効率良く計算するための手法として，高速フーリエ変換（Fast Fourier Transform, 以下FFTと略記する）アルゴリズムは，Cooley及びTukeyによって研究が始められた[1]．最近では，ディジタル信号処理の分野に限らず，回路網の記号解析[6],[7]，非線形系の周期解を求める等の数値計算の手法としても広く用いられている[8],[9]．それに伴い，FFTは，大型計算機はもちろん，パーソナルコンピュータでも使用されるようになっている．

小型の計算機を使用する場合は，大型計算機を用いる場合に比べ，過剰精度で計算するには限界があり，アルゴリズムとして誤差が少なく，さらに誤差評価できることが特に必要になる．浮動小数点演算によるFFTの誤差原因は，入力データの量子化誤差，四則演算による丸め誤差，$W^{nk}=\exp(-j2\pi nk/N)$の計算誤差に大別できる．その中で，W^{nk}の誤差表現は，複雑であり，計算機によって異なる．従来の誤差解析の研究では，W^{nk}の誤差を，計算誤差ではなく，常に有限ビットによる打切り誤差として評価している[2]～[4]．また，その近似の妥当性については，ほとんど論じられていない．従って，導かれた結果は，絶対的誤差の大きさとしては信頼性がなく，安心して誤差評価に使用することができない．

そこで，本論文では，まず，W^{nk}の計算精度とその影響について検討している．その結果，W^{nk}の計算誤差は，有限ビットによる打切り誤差に比べ，はるかに大きく，標本点数Nと共に増大することが確められた．また，FFTの結果に及ぼす影響も，非常に大きく，結果に含まれる誤差の上限はW^{nk}の計算誤差が支配している．さらに，この誤差は，各係数間の誤差のばらつきを大きくし，アルゴリズムを不安定にしてしまうことも分った．

これらの改善には，2つの方法が考えられる．その1つは，W^{nk}の計算誤差を正確に評価し，誤差解析を進める方法である．しかし，複雑な結果を導くことは，その実用性の点で好ましくないし，アルゴリズムの不安定性等の本質的な解決にはならない．本論文では，W^{nk}の精度を改善し，その誤差を有限ビットによる打切り誤差に近づける方法を提案する．これにより，すでに報告されている誤差解析の結果は，絶対的誤差の大きさを示すものとして有効になる．さらに，この方法は，FFTの精度を改善し，アルゴリズムを安定させることができる．この処理に必要なプログラムの変更は，非常にわずかであり，必要なメモリ，処理時間

† 東京都立大学工学部電気工学科，東京都
Faculty of Technology, Tokyo Metropolitan University, Tokyo, 158 Japan

は，直接処理の場合とほとんど同じである．

最後に，具体的な例題で，本理論の有効性を確認している．

2．FFTの誤差表現

ここでは，3.以後の準備として，浮動小数点演算でFFTを行った場合の誤差表現を与える．また，すでに報告されている最終結果に含まれる誤差の上限の導出を述べる．

2.1 FFT

DFTの定義式は，次式になる．

$$X(k)=\sum_{n=0}^{N-1} x(n)\cdot W^{nk},\quad k=0,1,\cdots,N-1 \tag{1}$$

$$W=\exp(-j2\pi/N),\qquad j=\sqrt{-1} \tag{2}$$

以下では，基数2のDIF法（Decimation In Frequency Algorithm）に基づくFFTを検討する．

まず，標本点数Nを，

$$N=2^m \tag{3}$$

と選ぶことにより，整数k（$0\leqq k<N$）は，

$$\left.\begin{aligned} k&=2^{m-1}\cdot b_0+2^{m-2}\cdot b_1+\cdots+2b_{m-2}+b_{m-1}\\ b_i&=1\quad\text{or}\quad 0\end{aligned}\right\} \tag{4}$$

の様にmビットの2進数で展開できる．その逆転2進数は，式(5)になる．

$$\overline{k}=2^{m-1}\cdot b_{m-1}+2^{m-2}\cdot b_{m-2}+\cdots+b_0 \tag{5}$$

FFTアルゴリズムは，よく知られている様に，次の様に一般的に表現できる．

$$X_{q+1}(k)=\begin{cases} X_q(k)+X_q(k+2^{m-1-q}), & b_q=0\\ [X_q(k-2^{m-1-q})-X_q(k)]W^{q(k)}, & b_q=1\end{cases} \tag{6}$$

ここで，

$$q=0,1,2,\cdots,m-1 \tag{7}$$

$$q(k)=2^{m-2}\cdot b_{q+1}+2^{m-3}\cdot b_{q+2}+\cdots+2^q\cdot b_{m-1} \tag{8}$$

$$X_0(k)=x(k),\quad X(k)=X_m(\overline{k}) \tag{9}$$

式(6)の$X_q(\ )$とは，FFTの中間の結果を表している．式(6)を$q=0$から$m-1$までくり返すことにより，FFTは実行される．また，式(6)では，$b_q=0$の場合，$W^{q(k)}$の計算誤差の影響を受けないことに注目してもらいたい．

2.2 FFTの誤差表現

FFTは，2進数の浮動小数点演算で実行されるとし，仮数部の桁数をtビットとする．この時，2進t桁で与えられるaとbの四則演算に関して，次の関係が成立する．

$$fl(a\pm b)=a(1+\alpha_1)\pm b(1+\alpha_2) \tag{10}$$

$$fl(a\cdot b)=a\cdot b(1+\alpha_3) \tag{11}$$

$$fl(a/b)=(a/b)(1+\alpha_4) \tag{12}$$

$$-2^{-t}<\alpha_i<2^{-t}\quad\text{0捨1入の場合} \tag{13}$$

$$-2^{-t+1}<\alpha_i\leqq 0\quad\text{切捨ての場合} \tag{14}$$

ここで，$fl(\ )$は，浮動小数点で計算された結果を表している．この様に，相対誤差α_iの形で誤差解析を展開する方法は，Wilkinsonによって研究され，後向き誤差解析（Backward Analysis）と呼ばれている[10]．本論文でも，この表現方法を用いる．

2進t桁で式(6)を実行した結果を$X'_{q+1}(k)$で表すと，それに含まれる誤差$e_{q+1}(k)$は，

$$e_{q+1}(k)=X'_{q+1}(k)-X_{q+1}(k) \tag{15}$$

と定義される．式(6)の$X_q(k)$，$W^{q(k)}$は，一般的に複素数なので，

$$B_q(k)=\mathrm{Re}[X_q(k)],\quad C_q(k)=\mathrm{Im}[X_q(k)] \tag{16}$$

$$U_q(k)=\mathrm{Re}[W^{q(k)}],\quad V_q(k)=\mathrm{Im}[W^{q(k)}] \tag{17}$$

の様に，実数部と虚数部に分けて表す．式(16)，(17)を用いて式(6)を表現すると次式になる．

$$\left.\begin{aligned} B_{q+1}(k)&=B_q(k)+B_q(p)\\ C_{q+1}(k)&=C_q(k)+C_q(p)\end{aligned}\right\} b_q=0 \tag{18}$$

$$\left.\begin{aligned} B_{q+1}(k)&=[B_q(h)-B_q(k)]U_q(k)\\ &\quad-[C_q(h)-C_q(k)]V_q(k)\\ C_{q+1}(k)&=[C_q(h)-C_q(k)]U_q(k)\\ &\quad+[B_q(h)-B_q(k)]V_q(k)\end{aligned}\right\} b_q=1 \tag{19}$$

ここで，

$$p=k+2^{m-1-q},\quad h=k-2^{m-1-q} \tag{20}$$

式(19)の$U_q(k)$，$V_q(k)$が正確であることを仮定し，式(18)，(19)を浮動小数点演算すると，

$$\left.\begin{aligned} B'_{q+1}(k)&=B'_q(k)(1+\tau_1)+B'_q(p)(1+\tau_2)\\ C'_{q+1}(k)&=C'_q(k)(1+\tau_3)+C'_q(p)(1+\tau_4)\end{aligned}\right\} b_q=0 \tag{21}$$

$$\left.\begin{aligned} B'_{q+1}(k)&=[B'_q(h)(1+\delta_1)-B'_q(k)(1+\delta_2)]\\ &\quad\cdot U_q(k)(1+\zeta_1)(1+\lambda_1)-[C'_q(h)(1+\delta_3)\\ &\quad-C'_q(k)(1+\delta_4)]V_q(k)(1+\zeta_2)(1+\lambda_2)\\ C'_{q+1}(k)&=[C'_q(h)(1+\varepsilon_1)-C'_q(k)(1+\varepsilon_2)]\\ &\quad\cdot U_q(k)(1+\eta_1)(1+\lambda_3)+[B'_q(h)(1+\varepsilon_3)\\ &\quad-B'_q(k)(1+\varepsilon_4)]V_q(k)(1+\eta_2)(1+\lambda_4),\\ b_q&=1\end{aligned}\right\} \tag{22}$$

と表される．ここで，$B'_{q+1}(k)$らの“，”は，式(15)と同様に，浮動小数点演算により導かれた誤差を含んだ結果を意味する．また，r_i，δ_i，ζ_i，λ_i，ε_i，η_iは，式(10)，(11)に従って発生しており，その値は，式(13)，(14)の範囲にある．$q=m-1$，$m-2$の場合は，$W^{q(k)}$の値が1あるいは$-j$となるので，$U_q(k)=0$あるいは$V_q(k)=$

0が成立する．従って，その演算には誤差が生じず，式(22)において，

$$\zeta_i=\lambda_i=\eta_i=0,\quad \text{for}\quad q=m-1,\ m-2 \tag{23}$$

と置くことができる．

式(15)から，実数部と虚数部に含まれる誤差は，次の様に表される．

$$e_{q+1}(k)=B'_{q+1}(k)-B_{q+1}(k)+j\{C'_{q+1}(k)-C_{q+1}(k)\} \tag{24}$$

$$e_0(k)=0,\quad k=0,1,2,\cdots,N-1 \tag{25}$$

ただし，式(25)は，入力データに誤差が含まれていないことを意味している．式(21)，(22)を式(24)に代入することによって，誤差の漸化式が以下の様に導出される．

$$e_{q+1}(k)=\begin{cases} e_q(k)+e_q(p)+f_q(k), & b_q=0\\ \{e_q(k)-e_q(k)\}W^{q(k)}+f_q(k), & b_q=1\end{cases} \tag{26}$$

ただし，

$$\begin{aligned} f_q(k)&=X'_{q+1}(k)-\{X'_q(k)+X'_q(p)\}\\ &=B'_q(k)r_1+B'_q(p)r_2+j\{C'_q(k)r_3+C'_q(p)r_4\},\quad b_q=0 \end{aligned} \tag{27}$$

$$\begin{aligned} f_q(k)&=X'_{q+1}(k)-\{X'_q(k)-X'_q(k)\}W^{q(k)}\\ &=\{B'_q(k)\{(1+\delta_1)(1+\zeta_1)(1+\lambda_1)-1\}\\ &\quad -B'_q(k)\{(1+\delta_2)(1+\zeta_1)(1+\lambda_1)-1\}\}U_q(k)\\ &\quad -\{C'_q(k)\{(1+\delta_3)(1+\zeta_2)(1+\lambda_2)-1\}\\ &\quad -C_q(k)\{(1+\delta_4)(1+\zeta_2)(1+\lambda_2)-1\}\}\\ &\quad \cdot V_q(k)+j\{C'_q(k)\{(1+\varepsilon_1)(1+\eta_1)(1+\lambda_3)\\ &\quad -1\}-C'_q(k)\{(1+\varepsilon_2)(1+\eta_1)(1+\lambda_3)-1\}\}\\ &\quad \cdot U_q(k)+j\{B'_q(k)\{(1+\varepsilon_3)(1+\eta_2)(1+\lambda_4)\\ &\quad -1\}-B'_q(k)\{(1+\varepsilon_4)(1+\eta_2)(1+\lambda_4)-1\\ &\quad -1\}\}V_q(k),\quad b_q=1 \end{aligned} \tag{28}$$

式(27)，(28)は，0捨1入演算とすると，式(13)より次の不等式が成立する．

$$|f_q(k)|\leqq 2^{-t}\{\{|B'_q(k)|+|B'_q(p)|\}^2+\{|C'_q(k)|+|C'_q(p)|\}^2\}^{1/2},\quad b_q=0 \tag{29}$$

$$\begin{aligned} |f_q(k)|\leqq{}&\{(1+2^{-t})^3-1\}\{\{|B'_q(k)|+|B'_q(k)|\}^2\\ &+\{|C'_q(k)|+|C'_q(k)|\}^2+4\{|B'_q(k)|\\ &+|B'_q(k)|\}\{|C'_q(k)|+|C'_q(k)|\}|U_q(k)\\ &\cdot V_q(k)|\}^{1/2},\quad b_q=1 \end{aligned} \tag{30}$$

次に，$|U_q(k)\cdot V_q(k)|\leqq 1/2$，$(1+2^{-t})^3-1\approx 3\cdot 2^{-t}$，$X'_q(k)\approx X_q(k)$なる近似を用いると，

$$|f_q(k)|\leqq 2^{-t}\{\{|B_q(k)|+|B_q(p)|\}^2+\{|C_q(k)|+|C_q(p)|\}^2\}^{1/2},\quad b_q=0 \tag{31}$$

$$|f_q(k)|\leqq 3\cdot 2^{-t}\{|B_q(k)|+|B_q(k)|+|C_q(k)|+|C_q(k)|\},\quad b_q=1 \tag{32}$$

となる．さらに，基本的な不等式を適用することによって，上式は式(33)，(34)となる．

$$|f_q(k)|\leqq 2^{-t}\{|X_q(k)|+|X_q(p)|\},\quad b_q=0 \tag{33}$$

$$|f_q(k)|\leqq 3\sqrt{2}\cdot 2^{-t}\{|X_q(k)|+|X_q(k)|\},\quad b_q=1 \tag{34}$$

以上の結果は，$U_q(k)$，$V_q(k)$つまり$W^{q(k)}$が正確であることを仮定して導かれている．また，切捨て演算の際の$|f_q(k)|$の上限は，式(13)，(14)から明らかな様に，式(33)，(34)の右辺の2倍になる．

2.3 誤差の上限[2]

ここでは，FFTの最終結果に含まれる誤差の上限の導出法を，簡単に述べる．やはり，$W^{q(k)}$が正確であることを仮定する．

式(26)で各々の項の絶対値をとると，次の不等式が成立する．

$$|e_{q+1}(k)|\leqq|e_q(k)|+|e_q(p)|+|f_q(k)|,\quad b_q=0 \tag{35}$$

$$|e_{q+1}(k)|\leqq|e_q(k)|+|e_q(k)|+|f_q(k)|,\quad b_q=1 \tag{36}$$

上式において，$q=m-1$と置くと，

$$|e_m(k)|\leqq|e_{m-1}(k)|+|e_{m-1}(p)|+|f_{m-1}(k)|,\quad b_q=0 \tag{37}$$

$$|e_m(k)|\leqq|e_{m-1}(k)|+|e_{m-1}(k)|+|f_{m-1}(k)|,\quad b_q=1 \tag{38}$$

と表される．$e_m(k)$は，最終的結果に含まれる誤差である．また，式(37)，(38)の右辺の$|e_{m-1}(\ )|$は，式(35)，(36)で$q=m-2$と置くことによって，再び$|e_{m-2}(\ )|$と$|f_{m-2}(\ )|$で表すことができる．この操作を$m-1$回くり返し，式(25)を用いることによって，式(37)，(38)は，以下の様になる．

$$\begin{aligned} |e_m(k)|\leqq{}&|f_{m-1}(s_{m-1,k})|+\sum_{i_{m-1}=0}^{1}|f_{m-2}(s_{m-2,k})|\\ &+\cdots+\sum_{i_2=0}^{1}\cdots\sum_{i_{m-1}=0}^{1}|f_1(s_{1,k})|+\sum_{i_1=0}^{1}\cdots\sum_{i_{m-1}=0}^{1}|f_0(s_{0,k})| \end{aligned} \tag{39}$$

ここで，

$$\begin{aligned} s_{q,k}={}&2^{m-1}\cdot b_0+2^{m-2}\cdot b_1+\cdots+2^{m-1-q}\cdot b_q\\ &+2^{m-2-q}\cdot i_{q+1}+\cdots+2\cdot i_{m-2}+i_{m-1} \end{aligned} \tag{40}$$

式(39)は，誤差の上限が，$f_q(s_{q,k})$だけで決定できることを示している．また，$f_q(s_{q,k})$と入力データの間には，次の不等式が成立する[2]．

$$\sum_{i_{q+1}=0}^{1}\cdots\sum_{i_{m-1}=0}^{1}|f_q(s_{q,k})|\leqq 2^{-t}\sum_{n=0}^{N-1}|x(n)|,\quad b_q=0 \tag{41}$$

$$\leqq 3\sqrt{2}\cdot 2^{-t}\sum_{n=0}^{N-1}|x(n)|,\quad b_q=1 \tag{42}$$

最終的誤差の上限は，式(41)，(42)を式(39)に代入し，式(9)より$e(k)=e_m(k)$であることと式(23)を考慮すること

により次式になる．

$$|e(k)| \leqq \left[m+(3\sqrt{2}-1)\sum_{q=2}^{m-1} b_q \right] 2^{-t} \sum_{n=0}^{N-1} |x(n)| \tag{43}$$

上式は，kを2進に直したb_qの値（式(4)参照）によって，誤差の上限が異なることを示している．b_qの値が全てのqに対して0の時の上限と，全てのqに対して1の時の上限の比は，$3\sqrt{2}$倍以下である．この$3\sqrt{2}$倍という値は，次章で示すように，$W^{q(k)}$に誤差が含まれると大きくなってしまう．切捨て演算の場合の上限は，式(43)の2^{-t}の代りに，2^{-t+1}を適用すればよい．

3. $W^{q(k)}$の計算精度の影響

実際の$W^{q(k)}$には，計算誤差が含まれている．ここでは，計算誤差の大きさと，その誤差がFFTの結果に及ぼす影響について検討する．

3.1 $W^{q(k)}$の計算精度

全ての場合に$W^{q(k)}$を正確に表すには，仮数部の桁数tが無限である必要がある[12]．今，正確な$W^{q(k)}$を有限なtビットで丸めたとすると，次の表現が可能になる．

$$〔U_q(k)〕_t = U_q(k)(1+\alpha_4) \tag{44}$$

$$〔V_q(k)〕_t = V_q(k)(1+\alpha_5) \tag{45}$$

α_4，α_5の値は，式(13)，(14)の範囲にある．また，$〔X〕_t$は，Xをtビットで丸めた数を表している．従来の誤差解析では，$W^{q(k)}$の誤差を上式の形でしか評価していない[2]～[4]．

しかし，実際のFFTのプログラムは，$W^{q(k)}=\exp(-j2\pi q(k)/N)$の値を，直接計算して求めている．まず，$2\pi q(k)/N$を求め，次に正弦関数及び余弦関数を計算する．表1は，余弦関数の計算結果を，単精度と倍精度演算で比較している．用いた計算機は，NECのPC-8800で，演算は切捨て演算である．倍精度演算の結果を正しい値とした際の単精度演算の相対誤差は，結果が0に近いほど大きくなっているのが分る．これは，余弦関数の角度に対する導関数が正弦関数であることからも分る様に，結果が0の付近では角度の偏差に対する感度が大きくなるためである．また，その相対誤差の大きさは，式(44)，(45)のα_4，α_5の値（式(14)より，$\alpha_i < 2^{-t+1} = 1.192\times10^{-7}$）より明らかに大きくなってしまう．次に，$2\pi q(k)/N$を倍精度で求め，値が求まった後で単精度に丸め，余弦関数を計算した場合の相対誤差を，表1に示している．直

表1　$\cos(2\pi q(k)/N)$の計算精度

$q(k)$	倍精度演算 $t=56$ビット	単精度演算 $t=24$ビット		$2\pi q(k)/N$ の精度改善
		計算結果	相対誤差	
0	1.0000000	1.000000	1.192E-7	1.192E-7
1	0.99518472	0.9951848	1.126E-7	1.126E-7
2	0.98078528	0.9807853	9.122E-8	9.122E-8
3	0.95694033	0.9569403	1.795E-8	1.795E-8
4	0.92387953	0.9238795	3.063E-8	3.387E-8
5	0.88192126	0.8819212	3.061E-8	3.061E-8
6	0.83146961	0.8314696	5.139E-8	5.139E-8
7	0.77301045	0.7730103	1.809E-7	2.046E-7
8	0.70710678	0.7071066	1.857E-7	6.717E-8
9	0.63439328	0.6343931	2.026E-7	7.917E-8
10	0.55557023	0.5555701	1.933E-7	2.118E-8
11	0.47139673	0.4713965	3.003E-7	1.575E-8
12	0.38268343	0.3826830	8.729E-7	6.161E-8
13	0.29028467	0.2902843	1.176E-6	5.507E-8
14	0.19509032	0.1950899	1.824E-6	9.251E-7
15	0.098017140	0.09801676	3.791E-6	9.114E-9

接計算と比較して非常に大きな精度改善効果があり，$2\pi q(k)/N$の計算精度が，正弦関数及び余弦関数の精度を決める重要な要素であることが分る．

この様な演算誤差を含んだ表現として，ここでは，次の様な表現を用いる．

$$fl〔U_q(k)〕 = U_q(k)(1+K_1\alpha_6) \tag{46}$$

$$fl〔V_q(k)〕 = V_q(k)(1+K_2\alpha_7) \tag{47}$$

α_6，α_7の値は，式(13)，(14)の範囲にある．$K_i\cdot\alpha_i$の値は，$W^{q(k)}$の計算に伴う相対誤差であり，標本点数Nや計算機によって異なる．$N=64$で$W^{q(k)}$を直接計算した場合は，$|K_i\cdot\alpha_i|$の範囲が，次の範囲になることが計算機の内部表現を出力させることによって確められた．

$$0 \leqq |K_i\cdot\alpha_i| < 2^{-t+9} \tag{48}$$

$N=2048$では，

$$0 \leqq |K_i\cdot\alpha_i| < 2^{-t+15} \tag{49}$$

となる．ただし，これらは，PC-8800を用い，N88-BASICで計算された場合である．次に，式(48)，(49)を用い，式(46)，(47)を切捨て演算の上限の形に書き直すと，

$$|fl〔U_q(k)〕| \leqq |U_q(k)|(1+H_N\cdot2^{-t+1}) \tag{50}$$

$$|fl〔V_q(k)〕| \leqq |V_q(k)|(1+H_N\cdot2^{-t+1}) \tag{51}$$

ただし，

$$H_N = \begin{cases} 2^8, & N=64 \\ 2^{14}, & N=2048 \end{cases} \tag{52}$$

と表される．また，HITAC，M-280HのFORTRAN77で，同様に$W^{q(k)}$を計算すると，H_N

の値は，式(52)に対して式(53)になる．

$$H_N=\begin{cases}2^6, & N=64\\ 2^{11}, & N=2048\end{cases} \tag{53}$$

以上の結果は，$W^{q(k)}$ の計算誤差は，無視したり，従来の誤差解析の様に $H_N=1$ として近似できるものではないことを示している．また，その誤差の大きさは，標本点数 N や計算機によって異なり，正確な誤差表現が複雑になってしまうことが分る．

0捨1入演算の場合は，切捨て演算の際と比べ式(48)，(49)の上限は小さくなる．しかし，$|\alpha_i|<2^{-t}$ であることから，式(52)，あるいは式(53)の H_N は，切捨て演算の場合とほぼ同程度であることが予想できる（ここでは，H_N の正確な値は，重要な問題ではない．$H_N\gg 1$ であることに注目してもらいたい）．

3.2 誤差の上限への影響

$W^{q(k)}$ を求める際に発生する相対誤差は，四則演算や有限ビットによる丸めの際の相対誤差に比べ，非常に大きい．以下では，$W^{q(k)}$ の計算誤差が，FFT の最終結果に含まれる誤差の上限に及ぼす影響について検討する．

$W^{q(k)}$ の誤差表現として式(46)，(47)を用いると，**2.** の式(22)に対して次式が成立する．

$$\left.\begin{aligned}B'_{q+1}(k)=&[B'_q(h)(1+\delta_1)-B'_q(k)(1+\delta_2)]\\&\cdot U_q(k)(1+\zeta_1)(1+\lambda_1)(1+K_1\cdot\alpha_6)\\&-[C'_q(h)(1+\delta_3)-C'_q(k)(1+\delta_4)]\\&\cdot V_q(k)(1+\zeta_2)(1+\lambda_2)(1+K_2\cdot\alpha_7)\\C'_{q+1}(k)=&[C'_q(h)(1+\varepsilon_1)-C'_q(k)(1+\varepsilon_2)]\\&\cdot U_q(k)(1+\eta_1)(1+\lambda_3)(1+K_1\cdot\alpha_6)\\&+[B'_q(h)(1+\varepsilon_3)-B'_q(k)(1+\varepsilon_4)]\\&\cdot V_q(k)(1+\eta_2)(1+\lambda_4)(1+K_2\cdot\alpha_7)\end{aligned}\right\} \tag{54}$$

$W^{q(k)}$ を正確であると仮定した場合と同様に，式(54)は，式(28)に考慮され，その上限は，式(30)に対して，

$$\begin{aligned}|f_q(k)|\leqq&[(1+2^{-t})^3(1+H_N\cdot 2^{-t})-1]\{[|B'_q(h)|\\&+|B'_q(k)|]^2+[|C'_q(h)|+|C'_q(k)|]^2\\&+4[|B'_q(h)|+|B'_q(k)|][|C'_q(h)|\\&+|C'_q(k)|]|U_q(k)\cdot V_q(k)|\}^{1/2}\end{aligned} \tag{55}$$

となる．ただし，上式は，0捨1入演算の場合の上限である．次に，$|U_q(k)\cdot V_q(k)|\leqq 1/2$，$X'_q(k)\approx X_q(k)$，$(1+2^{-t})^3(1+H_N\cdot 2^{-t})-1\approx 3\cdot 2^{-t}+H_N\cdot 2^{-t}$ なる近似を用いると，式(34)に対して式(56)になる．

$$|f_q(k)|\leqq(3+H_N)\sqrt{2}\cdot 2^{-t}[|X_q(k)|+|X_q(k)|] \tag{56}$$

以下の手順は，**2.** と同様であり，最終的結果に含まれる誤差の上限は，

$$|e(k)|\leqq\left\{m+[(3+H_N)\cdot\sqrt{2}-1]\sum_{q=2}^{m-1}b_q\right\}2^{-t}\cdot\sum_{n=0}^{N-1}|x(n)| \tag{57}$$

となる．切捨て演算の場合の上限は，上式 2^{-t} の代わりに 2^{-t+1} を代入するだけである．

$W^{q(k)}$ の計算誤差の影響は，式(43)と式(57)の比較から分る様に，式(43)の $3\sqrt{2}$ に対して $(3+H_N)\sqrt{2}$ になることである．一般的に，$3\ll H_N$ であることから，誤差の上限は，$W^{q(k)}$ の計算誤差が支配的であると言える．もちろん，これらの誤差は，従来の誤差解析が行っている様に，$H_N=1$ として近似できるものではないことが分る．また，k を2進に直した b_q の値によって異なる上限の比は，$W^{q(k)}$が正確な場合は $3\sqrt{2}$ 倍であったのに対し，ここでは $(3+H_N)\sqrt{2}$ 倍と大きくなってしまう．これは，FFT の結果に含まれる誤差が，各係数間で非常にばらつく可能性があることを示している．つまり，求められたある係数は6桁の精度が維持できているが，他の係数は2桁の精度しか維持できないということが生じる可能性を示しており，アルゴリズムとして不安定となり好ましくない．この様な $W^{q(k)}$ の計算誤差の影響は，標本点数 N が大きいほど顕著になる．

以上の様な $W^{q(k)}$ の誤差の影響は，$W^{q(k)}$ の計算誤差を正確に評価し，誤差解析を進めることによって，前もって予想することはできる．しかし，複雑な結果を導くことは，その実用性の点で好ましくないし，アルゴリズムの不安定性等の本質的な解決にはならない．次章では，$W^{q(k)}$ の精度を改善し，その誤差を有限ビットによる打切り誤差に近づける方法を提案する．

4. 誤差の軽減と評価

3. では，$W^{q(k)}$ の計算誤差が，有限ビットによる丸め誤差に比べ非常に大きく，FFT アルゴリズムを不安定にしていることを述べた．これらの問題は，$W^{q(k)}$ の精度を改善することによって解決できる．そこで，ここでは，$W^{q(k)}$ の精度を，FFTの処理時間，メモリをほとんど増すことなしに改善できることを指摘する．さらに，この処理は，FFT 演算に伴う誤差を軽減することができる．

$W^{q(k)}=\exp(-j2\pi q(k)/N)$ の精度は，$2\pi q(k)/N$ の計算精度と正弦関数及び余弦関数の計算精度で決まる．今，この部分に使われる仮数部の桁数を，t ビットから t' ビットに上げてやることを考える．具体的に

は，t が単精度であれば t' は倍精度，t が倍精度であれば t' は4倍精度と言うことになる．t' ビットの演算で求められた $W^{q(k)}$ は，t ビットの精度を保証できる[11]．従って，その $W^{q(k)}$ を t 桁で丸めると，式(44)，(45)の表現が可能となり，報告されている誤差解析の表現に一致する．しかし，この処理では，倍精度あるいは4倍精度の外部関数（$\cos\theta$ 及び $\sin\theta$）を使用するために，FFT の処理速度が遅くなってしまう．そこで，ここでは，$2\pi q(k)/N$ の計算だけを t' 桁で行い，その後の処理は，また t 桁で行う方法を提案する．

表2は，FFT を直接 t 桁や t' 桁で行った場合と，$W^{q(k)}$ を t' 桁，$2\pi q(k)/N$ だけを t' 桁で行った場合の処理時間を比較している．FFT の処理時間は，t 桁（(a)では単精度，(b)では倍精度）で直接行った場合と $2\pi q(k)/N$ の精度を上げて行った場合とでほとんど変わらない．これより，$2\pi q(k)/N$ の精度は，FFT の処理時間をほとんど増すことなしに，改善できることが分る．また，この処理を行うためのプログラムの変更は，$2\pi q(k)/N$ の計算に関係するわずか3，4の変数を倍精度あるいは4倍精度に定義し直すだけである．メモリの増加は，全く問題にならない．

次に，$2\pi q(k)/N$ の精度改善による $W^{q(k)}$ の精度改善効果について検討する．まず，H_N について比較すると，$2\pi q(k)/N$ の精度を改善することによって，式(52)に対して次式になる．

$$H_N=\begin{cases}2^6, & N=64\\ 2^8, & N=2048\end{cases} \tag{58}$$

特に，$N=2048$ で，その改善効果が大きいことが分る．これは，表1の結果からも分る様に，$2\pi q(k)/N$ の誤差が，結果が0に近い場合の主な誤差原因になっているためである．標本点数 N と共に誤差が増大するという不安定な要因は，$2\pi q(k)/N$ の精度を上げることにより，押さえることができる．

表3は，$W^{q(k)}$ の平均相対誤差で，精度改善効果を比較している．平均相対誤差では，$N=64$ で約1／7，$N=2048$ で約1／11 に改善できることが分る．有限ビットによる打ち切り誤差（式(14)より，$|\alpha_i|<2^{-t+1}=1.192\times10^{-7}$）とこの値を比べると，ほぼ打ち切り誤差の上限の2倍となっている．α_i の確立密度関数が一様分布すると仮定すると，平均打切り誤差は 2^{-t} となる[3]．表3の結果は，その 2^{-t} に対して5ビットないし6ビットの誤差があったものを，$2\pi q(k)/N$ の精度を上げることにより，2ビットないし3ビットの誤差に改善できることを示している．

この処理によって，$W^{q(k)}$ を打切り誤差として導かれた従来の誤差解析の結果は，FFT の直接計算に適用するよりはるかに信頼性を向上することができる．また，**3.** で検討した $W^{q(k)}$ の計算誤差によるアルゴリズムの不安定性も改善できる．さらに，FFT に伴う演算誤差を軽減する効果がある．以上の利点を持つ $2\pi q(k)/N$ の精度改善処理は，FFT の直接計算と比べ，計算速度，メモリをほとんど増すことなしに，プログラムの簡単な変更で実行できる．

表2　$W^{q(k)}$ の精度改善に伴う処理時間の比較

(a)　NEC PC-8800　N88-BASIC

N	FFTの単精度演算	FFTの倍精度演算	$W^{q(k)}$ を倍精度演算	$2\pi q(k)/N$を倍精度演算
64	44秒	1分35秒	2分16秒	49秒
128	1分42秒	3分55秒	3分31秒	1分57秒

(b)　HITAC M-280H　FORTRAN77

N	FFTの倍精度演算	FFTの4倍精度演算	$W^{q(k)}$を4倍精度演算	$2\pi q(k)/N$を4倍精度演算
1024	0.0412秒	0.169秒	0.149秒	0.0460秒
2048	0.0987秒	0.403秒	0.354秒	0.110　秒

表3　$W^{q(k)}$ の平均相対誤差

（NEC PC-8800 N88-BASIC）

N	$W^{q(k)}$の直接計算	$2\pi q(k)/N$の精度改善
64	1.39×10^{-6}	2.10×10^{-7}
2048	1.86×10^{-6}	2.68×10^{-7}

5. 例題及び検討

FFTの処理を直接行った場合と，$W^{q(k)}$ の精度を改善して行った場合を比較検討する．

表4は，$N=16$ の場合の例であり，倍精度演算の結果を正しい値として誤差を計算している．$N=16$ の場合でも，$W^{q(k)}$ の計算誤差は，打ち切り誤差に比べ大きく，FFT の計算精度を悪くしているのが分る．誤差が最も少ない係数は，演算の際に $W^{q(k)}$ が全く関係しない $k=0$，4，8，12 の場合で，その値は 2.18×10^{-4} である．この値は，当然 $W^{q(k)}$ の精度を改善しても変わらない．逆に，誤差が大きいのは，$W^{q(k)}$ が最も多く関係している $k=3$，7，11，15 の場合である．このことからも，誤差の上限は，$W^{q(k)}$ の計算精度が支

表4　$W^{q(k)}$の精度とFFTの誤差の関係

（NEC PC-8800 N88-BASIC）

n	入力データ $x(n)$	k	FFTに伴う誤差 $\|e(k)\|$			
			FFTの直接処理	$2\pi q(k)/N$ を倍精度	$W^{q(k)}$ を倍精度	誤差の上限
0	1	0	2.18E-4	2.18E-4	2.18E-4	2.68E-3
1	10000	1	4.64E-3	1.61E-3	1.12E-3	5.79E-3
2	1000	2	6.21E-3	1.47E-3	5.25E-4	5.79E-3
3	100	3	9.44E-3	1.92E-3	1.04E-3	8.91E-3
4	10	4	2.18E-4	2.18E-4	2.18E-4	2.68E-3
5	1	5	3.38E-3	9.42E-4	9.23E-4	5.79E-3
6	0	6	6.98E-3	1.47E-3	5.26E-4	5.79E-3
7	−1	7	1.14E-2	3.01E-3	1.88E-3	8.91E-3
8	−10	8	2.18E-4	2.18E-4	2.18E-4	2.68E-3
9	−100	9	3.85E-3	1.47E-3	1.26E-3	5.79E-3
10	−1000	10	6.21E-3	1.47E-3	5.26E-4	5.79E-3
11	−10000	11	8.42E-3	1.76E-3	1.16E-3	8.91E-3
12	1	12	2.18E-4	2.18E-4	2.18E-4	2.68E-3
13	1.11111	13	3.98E-3	1.01E-3	1.01E-3	5.79E-3
14	100	14	6.98E-3	1.47E-3	5.27E-4	5.79E-3
15	10	15	1.24E-2	3.18E-3	2.53E-3	8.91E-3

配していると言うことができる．また，FFT の直接処理の際の誤差は，$W^{q(k)}$ の誤差を有限ビットによる打切り誤差として導かれた誤差の上限よりも大きくなっている．このことは，従来の誤差解析が，FFT の直接処理の結果に対して，信頼性がないことを裏付けている．さらに，FFT の直接処理では，最悪の場合，10進で3桁の精度しか維持できないのに対し，$2\pi q(k)/N$ の精度を改善することによって5桁まで復元できることを確認している．

次に，以下のデータについてFFTを行った．

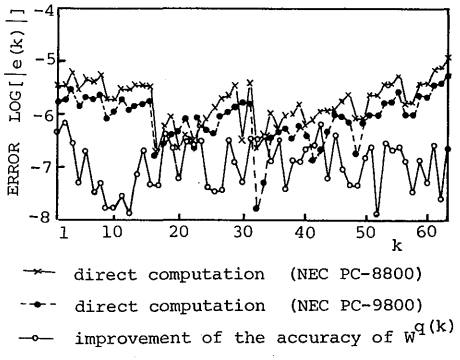

図1　$W^{q(k)}$の計算誤差によるFFTの誤差
Fig.1－Errors in the FFT computation by the inaccuracy of $W^{q(k)}$, (N=64).

$$\begin{cases} x(n)=1.0+j\,0.0, & (n<4,\ n>N-4) \\ x(n)=0.5+j\,0.5, & (n=4,\ n=N-4) \\ x(n)=0.0+j\,0.0, & (4<n<N-4) \end{cases}$$

用いた計算機は，NEC PC-8800 と PC-9800 であり，N88-BASIC で作成された全く同じプログラムを使用した．ただし，PC-9800では，$W^{q(k)}$の計算に，数値データプロセッサPC-9806を用いている．

図1は，FFTを直接計算した場合と，$W^{q(k)}$の誤差を打ち切り誤差まで改善して行った場合の誤差を比較している．誤差の計算は，倍精度演算の結果を正しい値として行っている．また，誤差が零の項は，あらかじめ取除いている．図1より，計算機によって誤差が異なること，$W^{q(k)}$ の計算誤差を打ち切り誤差で近似することは適切でないことが分る．

図2は，$2\pi q(k)/N$の精度改善効果を示している．$2\pi q(k)/N$ の精度を改善することによって，FFTの計算誤差は，軽減され，$W^{q(k)}$ の誤差を打ち切り誤差まで改善した場合の誤差に近づいているのが分る．この誤差の差は，Nの値を増やした場合，あるいはその他の例でも，約10進1桁に押さえることができることを確認している．従って，$W^{q(k)}$ の誤差を打ち切り誤差として導かれた誤差解析の結果は，直接計算のFFTにではなく，この処理を行ったFFTに適用することによって，その信頼性を向上することが可能になる．

6. むすび

本論文では，FFTの誤差原因の1つである$W^{q(k)}=\exp(-j\,2\pi q(k)/N)$の計算誤差について検討を行った．まず，$W^{q(k)}$ の誤差は，計算機により異なり，標本点数Nと共にその上限が増加することを示した．その値は，有限ビットによる打ち切り誤差よりはるかに大きく，その結果，従来の誤差解析での $W^{q(k)}$ の近似が適切でないことが分った．

FFT の正確な誤差解析は，標本点数，計算機の種類で異なる $W^{q(k)}$ の誤差を，実際に調べて始めて可能になる[5]．しかし，誤差解析に複雑な手間が必要であることは，その実用性の点で好ましくない．そこで，本論文では，$2\pi q(k)/N$の精度を改善し，$W^{q(k)}$ の誤差を打ち切り誤差に近づける方法を提案した．その結果，従来の誤差解析の信頼性が向上し，さらに，FFTに伴う誤差を軽減できることを確認できた．この処理

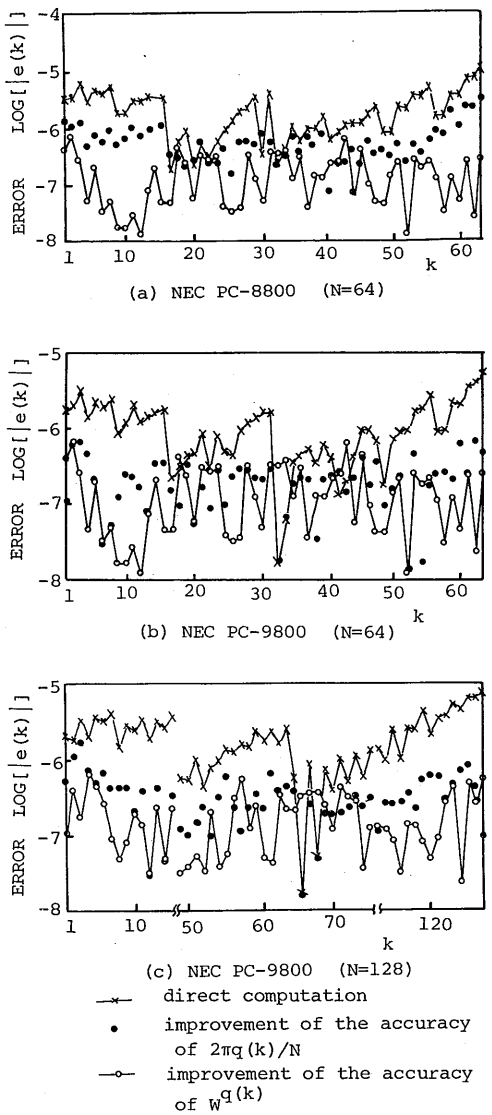

図2　$2\pi q(k)/N$の精度改善効果
Fig.2－Effects of improving the accuracy of $2\pi q(k)/N$.

は，FFTの直接計算に比べ，処理時間，メモリをほとんど増すことなしに実行できる．

なお，$W^{q(k)}$の精度が固定小数点演算によるFFTに及ぼす影響は，今後の興味ある問題である．

謝辞　日頃暖かい御教示を頂く，長岡技術科学大学神林紀嘉助教授に厚く感謝する．また，有益な御教示を頂いた，本学佐藤正光助教授，関本仁助手に感謝する．

文　献

(1) Cooly, J.W. and Tukey, J.W. : "An algorithm for the machine of complex Fourier series", Math. of Computation, 19, 90, pp.297-301 (1965).
(2) Kaneko, T. and Liu, B. : "Computation error in fast Fourier transform", in Proc. 3 rd Asilomar Conf. Circuit and Systems, pp.207-211 (1969).
(3) Kaneko, T. and Liu, B. : "Accumulation of roundoff errors in fast Fourier transforms", J. Ass. Comput. Mach., 17, pp.637-654 (Oct. 1970).
(4) Weinstein. C. J. : "Roundoff noise in floating point fast Fourier transform computation", IEEE Trans. Audio Electroacoust., **AU-17**, pp.209-215 (Sept. 1969).
(5) Ramos, G. U. : "Roundoff error analysis of the fast Fourier transform", Math. of Computation, 25, pp.757-768 (Oct. 1971).
(6) Singhal, K. and Vlack, J. : "Symbolic analysis of analog and digital circuits", IEEE Trans. Circuits and Systems, **CAS-24**, pp.598-609 (Nov. 1977).
(7) 本間，佐川："複数個の同種記号について整理された回路関数を与える記号解析"，信学論(A)，**J66-A**, 5, pp.424-431 (昭58-05).
(8) 奥村，木嶋："離散的フーリエ変換により非線形系の周期解を求める"，信学論(A)，**J62-A**, 3, pp.191-196 (昭54-03).
(9) 岡本，山下："巡回構造をした2次元ディジタルフィルタの伝達関数行列および出力をFFTアルゴリズムを用いて求める方法"，信学論(A)，**J63-A**, 1, pp.17-24 (昭55-01).
(10) 戸川隼人："計算機のための誤差解析の基礎"，サイエンス社，2章(昭49-10).
(11) 本間，佐川："FFT演算に伴う誤差軽減の一手法"，信学技報，**CAS83-31** (1983-06).
(12) Rader, C.M. : "A note on exact discrete Fourier transforms", IEEE Trans. Audio Electroacoust., **AU-21**, pp.558-559 (Dec. 1973)

（昭和58年7月12日受付，11月10日再受付）