精度保証付き高速フーリエ変換

2006MI125 野村久美子
指導教員：杉浦洋

1　はじめに

本研究では，複素データ $(f_l)_{i=0}^{n-1}$ に対する n 項複素離散型フーリエ変換 (DFT$_n$)

$$c_k = \sum_{l=0}^{n-1} \omega_n^{kl} f_l \quad (0 \le k < n) \tag{1}$$

に対する高速フーリエ変換 (FFT) の精度保証について考察した．

この問題においては，計算過程で発生する丸め誤差を確実に評価する必要がある．ここでは，精度保証付四則演算システム上で FFT を実行することで，丸め誤差の影響を確実にとらえることにした．そのために，複素数を複素平面の矩形領域として演算する矩形四則演算システム，円板領域として演算する円板四則演算システムを Mathematica 上で作成した．そして，２つのシステムを用いた FFT を作成し，特性を比較した．

いずれのシステムを用いても，FFT に現れる四則演算を精度保証付四則演算で置き換える方法を用いるため，演算量のオーダーは FFT の $O(n \log_2 n)$ となり，高速である．

2　基数２の FFT

本論文では，項数 $n = 2^m$ に対する基数２の FFT を用いた．基本アルゴリズムは以下である．

$$c_k^{(0)} = \sum_{l=0}^{n/2-1} \omega_{n/2}^{kl} f_{2l} \quad (0 \le k < n/2), \tag{2}$$

$$c_k^{(1)} = \sum_{l=0}^{n/2-1} \omega_{n/2}^{kl} f_{2l+1} \quad (0 \le k < n/2), \tag{3}$$

$$\left\{ \begin{array}{l} c_k = c_k^{(0)} + \omega_n^k c_k^{(1)}, \\ c_{k+n/2} = c_k^{(0)} - \omega_n^k c_k^{(1)} \quad (0 \le k < n/2). \end{array} \right. \tag{4}$$

式 (2)(3) の計算に基本アルゴリズムを再帰的に用いたものが基数２の FFT であり，必要とする四則演算数は $O(n \log_2 n)$ である．

3　演算システム

$\langle$ 区間演算システム $\rangle$
両端を浮動小数点数全体とする $\mathbb{IF}$ を

$$\mathbb{IF} = \{[\bar{x}, \underline{x}] \in \mathbb{IR} \mid \bar{x}, \underline{x} \in \mathbb{F}\}$$

とする．$\mathbb{IF}$ の要素を機械区間という．機械区間 $[x], [y]$ に対して，区間四則を

$$[x] + [y] = [\bigtriangledown(\underline{x} + \underline{y}), \bigtriangleup(\bar{x}, \bar{y})]$$

$$[x] - [y] = [\bigtriangledown(\underline{x} + \bar{y}), \bigtriangleup(\bar{x}, \underline{y})]$$

$$\begin{aligned}[x] \times [y] = [&\min\{\bigtriangledown\underline{x}\underline{y}, \bigtriangledown\underline{x}\bar{y}, \bigtriangledown\bar{x}\underline{y}, \bigtriangledown\bar{x}\bar{y}\}, \\ &\max\{\bigtriangleup\underline{x}\underline{y}, \bigtriangleup\underline{x}\bar{y}, \bigtriangleup\bar{x}\underline{y}, \bigtriangleup\bar{x}\bar{y}\}]\end{aligned}$$

$$\begin{aligned}[x]/[y] = [&\min\{\bigtriangledown\underline{x}/\underline{y}, \bigtriangledown\underline{x}/\bar{y}, \bigtriangledown\bar{x}/\underline{y}, \bigtriangledown\bar{x}/\bar{y}\}, \\ &\max\{\bigtriangleup\underline{x}/\underline{y}, \bigtriangleup\underline{x}/\bar{y}, \bigtriangleup\bar{x}/\underline{y}, \bigtriangleup\bar{x}/\bar{y}\}]\end{aligned}$$

で定義する．(ただし，$\bigtriangledown$:丸め下げ，$\bigtriangleup$:丸め上げ) 任意の区間 $[x] \in \mathbb{IR}$ について

$$[x] \subseteq [\bigtriangledown\underline{x}, \bigtriangleup\bar{x}].$$

ゆえに機械区間演算において，
$\bigcirc : \mathbb{IR} \to \mathbb{IF}$ を $\bigcirc[\underline{x}, \bar{x}] \to [\bigtriangledown\underline{x}, \bigtriangleup\bar{x}]$ で定義すると，

$$[x] \odot [y] = \bigcirc[x] \cdot [y] \qquad \cdot \in \{+, -, \times, \div\}$$

が成立する．

今回の実験では，Mathematica に実装されている区間演算システムを用いた．

$\langle$ 矩形演算システム $\rangle$
$[a_R], [a_I] \in \mathbb{IF}$ について，複素矩形領域を

$$[a] = [[a_R], [a_I]] = \{a_R + ia_I : a_R \in [a_R], a_I \in [a_I]\}$$

で定義する．複素数の四則 $c = a \cdot b, \cdot \in \{+, -, \times, /\}$ について，$a \in [a], b \in [b]$ ならば $c \in [c]$ であるような $[c]$ を $[a], [b]$ の矩形四則と呼び，

$$[c] = [a] \cdot [b]$$

と書く．
本論文では，次のように定義される矩形四則を Mathematica の区間演算システムを用いて実現した．
$[a] = [[a_R], [a_I]], [b] = [[b_R], [b_I]]$ として，

$$[a] \pm [b] = [[a_R] \pm [b_R], [a_I] \pm [b_I]]$$

$$[a] \times [b] = [[a_R][b_R] - [a_I][b_I], [a_R][b_I] + [a_I][b_R]]$$

$$[a]/[b] = [\frac{[a_R][b_R] + [a_I][b_I]}{[b_R]^2 + [b_I]^2}, \frac{-[a_R][b_I] + [a_I][b_R]}{[b_R]^2 + [b_I]^2}]$$

矩形演算は，実部，虚部をそれぞれ，区間演算したものである．
注意：除算の分母に $[b_R][b_R] + [b_I][b_I]$ を使うことはできない．

$\langle$ 円板演算システム $\rangle$
計算機上において，実際に円板四則演算を行うため，ま

ず論理の上で円板四則演算を定義する必要がある．中心 $c \in \mathbb{C}$，半径 $r \in \mathbb{R}$ とする．このとき複素閉円板領域

$$Z = \{z : \mid z - c \mid \leq r\} \tag{5}$$

は簡単に，

$$Z = \langle c; r \rangle \tag{6}$$

と表し，以後これを単に円板と呼ぶ．円板の中心は $c = \text{mid } Z$，半径は $r = \text{rad } Z$ と表す．また，円板全体の集合を $\mathbb{KC}$ とする．
次に，円板四則演算について定義する．２つの円板 $A = \langle a; r \rangle, B = \langle b; s \rangle \in \mathbb{KC}$ に対する四則演算を次のように定義し，Mathematica の区間演算システムを用いて実装した．
加減算

$$A \pm B \stackrel{\text{def}}{=} \langle a \pm b; r + s \rangle = \{z \pm w : z \in A, w \in B\}$$

逆円板

$$\frac{1}{A} \stackrel{\text{def}}{=} \langle \frac{\bar{a}}{|a|^2 - r^2}; \frac{r}{|a|^2 - r^2} \rangle = \{\frac{1}{z} : z \in A\} \qquad (0 \notin A)$$

乗算

$$AB \stackrel{\text{def}}{=} \langle ab; |a|s + |b|r + rs \rangle \supseteq \{zw : z \in A, w \in B\}$$

除算

$$A/B \stackrel{\text{def}}{=} A(1/B) \supseteq \{\frac{z}{w} : z \in A, w \in B\}$$

四則演算子を $* \in \{+, -, \cdot, /\}$ とすると，円板四則演算により $\alpha \in A, \beta \in B$ なら，$\alpha * \beta \in A * B$ が保証される．このように円板四則演算は，四則演算結果の精度保証を可能にする．
また，複素数と円板からなる四則演算は複素数を $Z = \langle c; 0 \rangle$ とみなして演算を行う．

4 実験結果

矩形演算においては，入力データ $z = x + iy$ をすべて，辺長０の点矩形

$$[z] = [[x, x], [y, y]]$$

に変換した後 FFT の計算を矩形演算で行う．円板演算においては，入力データ z をすべて

$$\langle z \rangle = \langle z; 0 \rangle$$

に変換した後，FFT の計算を円板演算で行う．

計算量は，矩形化しても円板化しても $O(n \log n)$ であり，通常の FFT とオーダーは変わらない．

図１は通常の数値計算，矩形演算，円板演算の計算時間を比較したものである．横軸は m，縦軸は計算時間の常用対数である．通常の計算に比べ矩形演算は２０倍，円板演算は５０倍程度の時間がかかる．しかし，グラフはほぼ直線で計算量が $O(n \log n)$ であることが分かる．

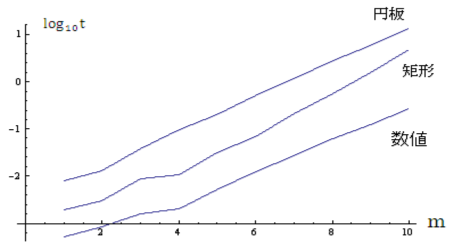

図 1 要する時間に関するグラフ

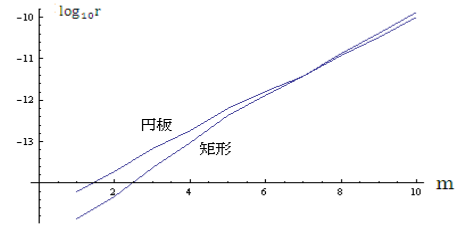

図 2 半径のグラフ

図２はフーリエ係数を含む矩形と円板の最大半径を表す．横軸は m，縦軸は半径の常用対数である．グラフはほぼ直線で，矩形演算においては半径 $r = O(n^{1.8})$，円板演算においては $r = O(n^{1.5})$ であることが分かる．n が小さいときには矩形演算のほうが半径は小さいが，n が大きくなるにつれて円板演算のほうが半径が小さく，精密な精度保証であることがわかる．

5 おわりに

矩形演算システムと円板演算システムを用いて FFT の精度保証を行った．演算量はどちらも $O(n \log n)$ の高速アルゴリズムである．しかし，係数が大きいため通常の数値計算と比べ矩形演算で２０倍，円板演算で５０倍程度の時間がかかる．また，m が小さいときには矩形演算のほうが，大きいときには円板演算のほうが精度保証が精密である．

これらの現象の原因を分析し，矩形演算と円板演算の長所を兼ね備えた計算法を開発することが今後の課題である．また，円板演算と結果が等価でより高速な計算法を開発することも課題である．

参考文献

[1] . Cooley, J. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comp. 19 (1965), 297-301.

[2] 二宮一三:『数値計算のつぼ』. 共立出版，東京，2004.