

Dutt-Rokhlin の不等間隔高速順フーリエ変換の数値的研究

三輪俊一[†] 須田礼仁^{††} 杉原正顯[†]

Dutt-Rokhlin の不等間隔高速順フーリエ変換では、Toeplitz 系を高速に解く必要が生じる。本稿では、最近開発されたいくつかの Toeplitz 系に対する高速解法を組み込み、Dutt-Rokhlin の不等間隔高速順フーリエ変換の性能を調べる。

Numerical Study of the Dutt-Rokhlin Fast Forward Fourier Transform for Nonequispaced Data

SHUN-ICHI MIWA,[†] REIJI SUDA^{††} and MASAOKI SUGIHARA[†]

The Dutt-Rokhlin fast forward Fourier transform for nonequispaced data requires a fast solver for Toeplitz systems. In the present note the Dutt-Rokhlin fast forward Fourier transform is examined when incorporated with several recent developed solvers for Toeplitz systems.

1. はじめに

高速フーリエ変換 (FFT)³⁾ は信号処理から気象計算に至るまで広範な分野で用いられる強力な数値計算アルゴリズムである。しかしながら、FFT が使用できるためにはデータ点が等間隔でなければならないという強い制約がつく。これに対して、Dutt と Rokhlin は、近似計算であることを許すならば、この制約に縛られない高速フーリエ変換アルゴリズムが得られることを示した⁴⁾。すなわち、彼等は、 x_j ($j = 0, 1, \dots, N-1$) を区間 $[-\pi, \pi]$ 上の任意の値とすると、不等間隔散逆フーリエ変換

$$f_j = \sum_{k=-N/2}^{N/2-1} \hat{f}_k e^{ikx_j} \quad (j = 0, 1, \dots, N-1) \quad (1)$$

および、不等間隔散逆フーリエ変換— f_j ($j = 0, 1, \dots, N-1$) から (1) を満たすような \hat{f}_k ($k = -N/2, \dots, N/2-1$) を求めること—が、要求精度を ε とするとき、 $O(N \log N + N \log 1/\varepsilon)$ の計算の手間で実行できることを示した。彼等のアルゴリズムは、

不等間隔散逆フーリエ変換の場合、巧妙な近似式を用いて、その計算を等間隔散逆フーリエ変換の計算で近似するというものであり、また、不等間隔散逆フーリエ変換の場合には、さらに、問題を正定値 Hermite 行列を係数行列にもつ連立一次方程式を解く形に変形し、(前処理なしの) 共役勾配法を用いるというものである。Dutt-Rokhlin も指摘していることであるが、この不等間隔散逆フーリエ変換の場合に現れる連立一次方程式の係数行列は単に正定値 Hermite 行列というだけではなく Toeplitz 行列でもある。Toeplitz 行列を係数行列にもつ連立一次方程式、すなわち、Toeplitz 系に対しては、近年、前処理付き共役勾配法に関して様々な研究が行われている^{1),2)}。本稿では、これらの研究成果を踏まえて、不等間隔散逆フーリエ変換の計算において、いくつかの Toeplitz 系の高速解法の適用を試みたので、その結果を報告する。

本稿の構成は次の通りである。まず、次節で Dutt-Rokhlin の不等間隔高速フーリエ変換について説明し、続いて、第 3 節で Toeplitz 系の高速解法について説明する。第 4 節において数値実験結果を示し、第 5 節において結果のまとめを与える。

2. Dutt-Rokhlin の不等間隔高速フーリエ変換

2.1 不等間隔高速逆フーリエ変換

Dutt-Rokhlin の不等間隔高速逆フーリエ変換では近似式

[†] 名古屋大学大学院工学研究科計算理工学専攻
Department of Computational Science and Engineering, Graduate School of Engineering, Nagoya University

^{††} 東京大学大学院情報理工学系研究科コンピュータ科学専攻
Department of Computer Science, Graduate School of Information Science and Technology, The University of Tokyo

$$e^{icx} \simeq e^{bx^2} \sum_{k=[c]-q/2}^{[c]+q/2} \frac{1}{2\sqrt{b\pi}} e^{-(c-k)^2/4b} e^{ikx} \quad (2)$$

を用いて、不等間隔離散逆フーリエ変換の計算を等間隔離散逆フーリエ変換の計算に帰着させる。(2)に現れる実数 b 、整数 q は精度を決めるパラメータであり、 b, q を適切に選ぶことにより精度保証をすることができ⁷⁾、結果的に、要求精度を ε とするとき、不等間隔離散逆フーリエ変換 (1) の計算が $O(N \log N + N \log 1/\varepsilon)$ の計算の手間で実行できる。

なお、つぎの 2.2 で必要となる周波数データが不等間隔である離散逆フーリエ変換

$$g_j = \sum_{k=0}^{N-1} \hat{g}_k e^{i\omega_k \cdot 2\pi j/N} \left(j = -\frac{N}{2}, \dots, \frac{N}{2} - 1 \right) \quad (3)$$

(ただし $\omega_k \in [-N/2, N/2]$) の計算も、近似式 (2) を用いて等間隔離散逆フーリエ変換の計算に帰着させることができ、要求精度を ε とするとき、 $O(N \log N + N \log 1/\varepsilon)$ の手間で計算できる。

2.2 不等間隔高速順フーリエ変換

まず、(3)において、 $x_k = -\omega_k \cdot 2\pi/N (\in [-\pi, \pi])$ とおくと、

$$g_j = \sum_{k=0}^{N-1} \hat{g}_k e^{-ijx_k} \left(j = -\frac{N}{2}, \dots, \frac{N}{2} - 1 \right) \quad (4)$$

となることに注意する。このとき、(1) および (4) は、 (j, k) 成分が $F_{jk} = e^{ikx_j}$ で与えられる行列 F を用いると、行列・ベクトル形式で

$$\mathbf{f} = F \hat{\mathbf{f}}, \quad \mathbf{g} = F^* \hat{\mathbf{g}} \quad (5)$$

と書ける (ただし、ここで、 F^* は行列 F の共役転置行列を表す)。したがって、2.1 の結果によれば、ベクトルに F または F^* を掛ける演算が、要求精度を ε とするとき、 $O(N \log N + N \log 1/\varepsilon)$ の手間で計算できることが分かる。

一方、不等間隔離散逆フーリエ変換とは、方程式

$$F \hat{\mathbf{f}} = \mathbf{f} \quad (6)$$

を解く計算であるが、この方程式は

$$F^* F \hat{\mathbf{f}} = F^* \mathbf{f} \quad (7)$$

と変形できる。明らかに $F^* F$ は正定値 Hermite 行列であり、共役勾配法が適用可能であり、反復回数 $O(\kappa(F))$ (κ は行列 F の条件数) で近似解を与える。共役勾配法の演算のうち主要なものは、ベクトルに F または F^* を掛ける演算であり、上記に見たようにこれ

は $O(N \log N + N \log 1/\varepsilon)$ の手間で実行できる。したがって、(7) を解く計算、すなわち、不等間隔離散逆フーリエ変換が計算量 $O(\kappa(F)(N \log N + N \log 1/\varepsilon))$ で実行できることが分かる。

以上が Dutt-Rokhlin の不等間隔高速順フーリエ変換のアルゴリズムであるが、素朴に前処理のない共役勾配法を用いている。しかし、 $F^* F$ を詳しくみると、その成分が

$$[F^* F]_{jk} = \sum_{l=0}^{N-1} e^{i(k-j)x_l} \quad 0 \leq j, k \leq N-1 \quad (8)$$

で与えられることから分かるように、Toeplitz 行列である。Toeplitz 行列を係数行列にもつ連立一次方程式、すなわち、Toeplitz 系に対しては、近年、前処理付き共役勾配法に関して様々な研究が行われており、その成果を上記の不等間隔高速順フーリエ変換の計算に取り入れてみるのは自然なことである。

3. Toeplitz 系の高速解法

ここでは、我々が、不等間隔高速順フーリエ変換の計算に組入れた、Toeplitz 系のための前処理付き共役勾配法、および、Toeplitz 系の直接解法であるが計算量が $O(N^2)$ の Levinson-Rybyckii のアルゴリズム⁵⁾ [pp.193-205] について説明する。

3.1 前処理付き共役勾配法 (PCG 法)

方程式 $Ax = b$ に対する PCG 法のアルゴリズムは以下の通りである。

- (1) $\mathbf{r}^{(0)} = b - A \mathbf{x}^{(0)}, k = 0$
- (2) $\mathbf{p}^{(0)} = C^{-1} \mathbf{r}^{(0)}$
- (3) $\alpha_k = (\mathbf{r}^{(k)}, \mathbf{p}^{(k)}) / (\mathbf{p}^{(k)}, A \mathbf{p}^{(k)})$
- (4) $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$
- (5) $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A \mathbf{p}^{(k)}$
- (6) $\beta_k = -(C^{-1} \mathbf{r}^{(k+1)}, A \mathbf{p}^{(k)}) / (\mathbf{p}^{(k)}, A \mathbf{p}^{(k)})$
- (7) $\mathbf{p}^{(k+1)} = C^{-1} \mathbf{r}^{(k+1)} + \beta_k \mathbf{p}^{(k)}$
- (8) 残差 $\|\mathbf{r}^{(k+1)}\| < \varepsilon \|\mathbf{r}^{(0)}\|$ 収束判定。
 $k = k + 1 \rightarrow (3) \rightarrow$

行列 A が正定値 Hermite-Toeplitz 行列の場合 (我々が扱う場合もこの場合に当たる) には、計算量の観点から前処理行列 C として循環行列が用いられる。

なお、Toeplitz 行列においては、行列成分 $A_{jk} = a_{j-k}$ が母関数 $f(\theta)$ により

$$a_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta \quad (9)$$

$k = -N + 1, \dots, N - 1$ と表されることが知られている。さらに、正定値 Hermite であれば、母関数は非負値をとる実数値関数にとれる。また、母関数が条件

$\sum_{k=0}^{\infty} |a_k| < \infty$ を満たすとき, Wiener class に入るといわれる.

以下, いくつかの有名な前処理について述べる.

3.1.1 Strang の前処理

行列成分 $A_{jk} = a_{j-k}$ とするとき, 循環行列 C の行列成分 $C_{jk} = c_{j-k}$ を

$$c_j = \begin{cases} a_j & (0 \leq j \leq [N/2]) \\ a_{j-N} & ([N/2] < j < N) \\ c_{N+j} & (0 < -j < N) \end{cases}$$

とするのが Strang の前処理である.

ここで, 循環行列 C による前処理「 $C^{-1} \times$ ベクトル」の計算について注意しておく. 循環行列 C は $C = U^* \Lambda U$ と分解できることが知られている. ここで, Λ は C の固有値

$$\lambda_k = \sum_{j=0}^{N-1} c_j e^{2\pi i j k / N}$$

を対角成分にもつ対角行列, U は行列成分が

$$U_{jk} = \frac{1}{\sqrt{N}} e^{2\pi i j k / N}$$

で与えられるユニタリ行列である. したがって, $C^{-1} = U^* \Lambda^{-1} U$ であり, FFT を用いて「 $C^{-1} \times$ ベクトル」の計算が $O(N \log N)$ の計算の手間で実行できる.

3.1.2 T.Chan の前処理

循環行列 C として, 行列成分 $C_{jk} = c_{j-k}$ を

$$c_j = \begin{cases} \frac{(N-j)a_j + ja_{j-N}}{N} & (0 \leq j < N) \\ c_{N+j} & (0 < -j < N) \end{cases}$$

とするのが T.Chan の前処理である.

3.1.3 Jackson 核関数を用いた前処理²⁾

3.1.1 において説明したように, 循環行列を用いた前処理においては, その固有値が指定されれば十分である. ここ 3.1.3, および, つぎの 3.1.4 では, この固有値が指定される形の前処理について述べる.

Jackson 核関数は

$$K_{m,2r}(\theta) = \frac{k_{m,2r}}{m^{2r-1}} \left(\frac{\sin(m\theta/2)}{\sin(\theta/2)} \right)^{2r}, \quad r = 1, 2, \dots$$

と定義される関数である. Jackson 核関数を用いた前処理では, Jackson 核関数と母関数の畳み込み $(K_{m,2r} * f)(\theta)$ を用いて, C の固有値 $\Lambda = \text{diag}(\lambda_0, \dots, \lambda_{N-1})$ を

$$\lambda_j(C[(K_{m,2r} * f)]) = (K_{m,2r} * f) \left(\frac{2\pi j}{N} \right)$$

と指定する.

$r = 1$ の時の Jackson 核関数 (この場合 Fejér 核関数とよばれる) を用いた前処理は T.Chan の前処理となっており, また, Jackson 核関数をいわゆる Dirichlet 核関数に換えて得られる前処理は Strang の前処理となっており, その意味で, Jackson 核関数を用いた前処理はこれらの前処理の拡張といえる.

3.1.4 熱伝導方程式の混合問題に現れる核関数を用いた前処理

熱伝導方程式の混合問題の場合には, 次のような核関数が現れる⁶⁾:

$$K_t(\theta) = \sum_{k=-\infty}^{\infty} e^{-k^2 t} e^{ik\theta}, \quad \theta \in [-\pi, \pi]$$

これは Jackson 核関数のある種の極限形と考えられる. そこで, 我々は, この核関数を有限で切った

$$K_t^{(N)}(\theta) = \sum_{k=-N+1}^{N-1} e^{-k^2 t} e^{ik\theta}$$

を用いた前処理についても, 我々の数値実験の対象とした.

3.1.5 注 意

Strang の前処理は, 母関数が偶関数, 正, Wiener class に入るときに有効であることが知られている¹⁾. また, T.Chan の前処理は, 偶関数である必要がないことが知られている¹⁾. しかし, 両者とも, 母関数が零点を持つと有効でなくなることが知られており, R. Chan, A. Yip, M. Ng はこの場合にも対処するため Jackson 核関数を用いた前処理を提案した²⁾.

いま, 我々の問題に戻って考えるとき, 行列成分 $[F^* F]_{jk} = a_{j-k}$ は (8) 式で与えられるので, 母関数は

$$f(\theta) = \sum_{j=0}^{N-1} \delta(\theta - x_j), \quad \theta \in [-\pi, \pi] \quad (10)$$

とするのが自然である. しかし, この母関数は特異であり, いずれの前処理についても, このような関数についてはその有効性は確立されてはいない.

3.2 Levinson-Rybicki のアルゴリズム

Toeplitz 行列 A に対して, $Ax = b$, つまり,

$$\begin{bmatrix} 1 & r_1 & \cdots & r_{N-1} \\ p_1 & 1 & r_1 & \vdots \\ \vdots & p_1 & \ddots & \ddots \\ & & \ddots & \ddots & r_1 \\ p_{N-1} & \cdots & & p_1 & 1 \end{bmatrix} x = b$$

を解く .

$k = 1, 2, \dots, N$ に対して

$$\begin{aligned} p_k &= [p_1 \cdots p_k]^T, & r_k &= [r_1 \cdots r_k]^T, \\ x_k &= [x_1 \cdots x_k]^T, & y_k &= [y_1 \cdots y_k]^T, \\ w_k &= [w_1 \cdots w_k]^T, \end{aligned}$$

とおくとき, $x_1 = b_1, y_1 = -r_1, w_1 = -p_1$ として

漸化式

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} x_k + (b_{k+1} - p_k^T E_k x_k)/(1 + p_k^T y_k) E_k y_k \\ (b_{k+1} - p_k^T E_k x_k)/(1 + p_k^T y_k) \end{bmatrix} \\ y_{k+1} &= \begin{bmatrix} y_k - (r_{k+1} + r_k^T E_k y_k)/(1 + r_k^T w_k) E_k w_k \\ -(r_{k+1} + r_k^T E_k y_k)/(1 + r_k^T w_k) \end{bmatrix} \\ w_{k+1} &= \begin{bmatrix} w_k - (p_{k+1} + p_k^T E_k w_k)/(1 + p_k^T y_k) E_k y_k \\ -(p_{k+1} + p_k^T E_k w_k)/(1 + p_k^T y_k) \end{bmatrix} \end{aligned}$$

にしたがって, $k = 1, \dots, N - 1$ と計算を進めれば, x_N として解が得られる . ただし, ここで, $E_k = [e_k, \dots, e_1]$ である . この算法の計算量は $O(N^2)$ である .

4. 数値実験

今回の数値実験では, Intel Xeon 2.0GHz を使用し, 言語は FORTRAN90, コンパイラは ifc -O3 を用いる .

データ点 $\{x_0, \dots, x_{N-1}\} \in [-\pi, \pi]$ は次のようにとる :

$$x_k = -\pi + 2\pi \frac{k + 0.5 + \delta_k}{N}. \quad (11)$$

ここで, δ_k は区間 $[-0.1, 0.1]$ または $[-0.5, 0.5]$ 上の一様分布からランダムに選ぶ . この一様分布の幅はデータ点が等間隔点からどの程度離れた位置に点があるかを定めるものである .

4.1 各種 PCG 法の比較

方程式 $F^* F y = b$ を 3.1 であげた前処理を用いた PCG 法で解いてその性能を比較する . 右辺ベクトル b は, 乱数ベクトル $\text{Re}(\hat{f}), \text{Im}(\hat{f}) \in [0, 1]^N$ から, 直接計算 $b = F^* F \hat{f}$ を行って与える . また, F^* または F とベクトルの積を計算する際の精度を決めるパラメータは倍精度を保証するものを選び, 問題のサイズ N は 32 ~ 1024 とした . CG 法による収束判定は, $\|r^{(k+1)}\| / \|r^{(0)}\| < 10^{-14}$ とした . 計算は $\{x_k\}$ と $\{b_j\}$ の 10 セットの値に関して行った .

結果 (PCG 法の反復回数の 10 回の平均) を表 1 ~ 表 2 に示す . 表では I を前処理なし, Strang, T.Chan

の方法を S, T とする . また, Jackson 核関数, 混合問題の核関数による前処理を $K_{m,2r}, K_t$ とする . Strang の前処理の欄で "—" は解へ収束しなかったことを示す .

T.Chan の前処理が最も効果があり, 次に適切な t を

表 1 PCG 法の反復回数: $\delta \in [-0.1 : 0.1]$

| N | I | S | T | $K_{m,4}$ | $K_{m,6}$ | K_t |
|------|------|------|------|-----------|-----------|-------|
| 32 | 16.2 | 13.0 | 12.3 | 13.7 | 14.9 | 12.6 |
| 64 | 17.0 | 14.0 | 13.2 | 14.3 | 15.1 | 13.7 |
| 128 | 17.6 | 14.2 | 13.8 | 15.0 | 15.7 | 14.0 |
| 256 | 17.9 | 14.7 | 14.0 | 15.0 | 15.9 | 14.0 |
| 512 | 18.1 | 14.7 | 14.2 | 15.1 | 16.0 | 14.2 |
| 1024 | 18.5 | 15.0 | 14.4 | 15.3 | 16.1 | 14.5 |

表 2 PCG 法の反復回数: $\delta \in [-0.5 : 0.5]$

| N | I | S | T | $K_{m,4}$ | $K_{m,6}$ | K_t |
|------|-------|---|-------|-----------|-----------|-------|
| 32 | 33.3 | — | 27.0 | 29.4 | 31.4 | 26.9 |
| 64 | 59.2 | — | 41.9 | 46.7 | 51.1 | 42.0 |
| 128 | 97.2 | — | 60.1 | 70.6 | 79.5 | 60.5 |
| 256 | 143.2 | — | 83.4 | 98.1 | 111.2 | 84.3 |
| 512 | 195.0 | — | 112.4 | 132.6 | 151.1 | 114.2 |
| 1024 | 289.3 | — | 164.5 | 195.8 | 223.4 | 167.0 |

選んだ核関数 K_t の前処理がよい . ただし $t = \pi/N^2$ とした .

最も収束が速かった T.Chan の前処理について, 反復回数がどの程度減っているかを, $N = 512$ とし, $\delta \in [-0.5 : 0.5]$ の場合について, サンプルを多くして見てみた . 結果を図 1 に示す . 前処理なしに比べて約 57% になっていることがみとれる .

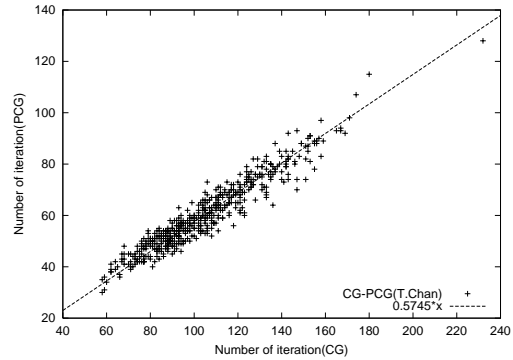


図 1 CG 法と PCG 法 (T.Chan の前処理) の反復回数

前処理行列の固有値 - $N = 128$ の場合に, $\delta \in [-0.1, 0.1], [-0.5, 0.5]$ のそれぞれの場合に対して, 前

処理なし, Strang, T.Chan の前処理, Jackson, K_t の核関数による前処理をした行列の固有値分布を図 2, 図 3 に示す.

図 2 からは, 固有値は前処理により 1 に集まって

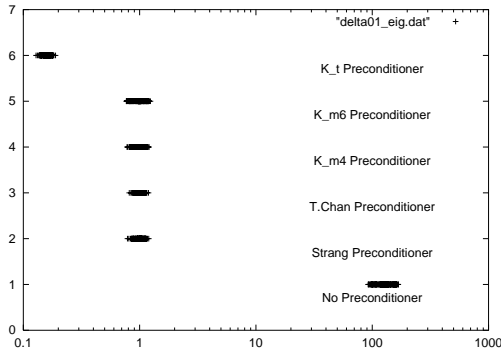


図 2 前処理後の行列の固有値: $\delta \in [-0.1 : 0.1]$

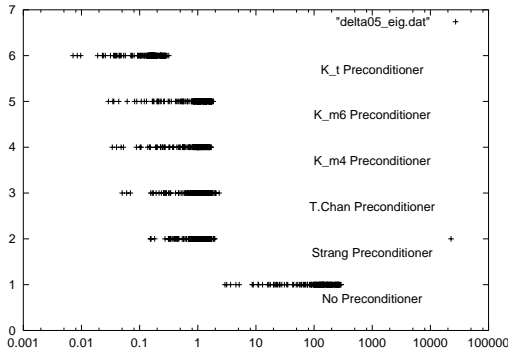


図 3 前処理後の行列の固有値: $\delta \in [-0.5, 0.5]$

いるが最大固有値と最小固有値の幅はそれほど減っていないこと, すなわち, 最大固有値と最小固有値の比より与えられる条件数はそれほど減っていないことがわかる.

また図 3 では, Strang の前処理については $\delta \in [-0.40815, 0.40815]$ のときの固有値を載せた. δ の幅がこの値に近づくにつれ固有値の 1 つが限りなく大きくなってゆき, その値より大きくなると固有値のいくつかは負となっていた. これは $\delta \in [-0.5 : 0.5]$ のときに Strang の PCG 法が収束しなかったことの原因になると考えられる.

4.2 Toeplitz 系の高速解法を用いた不等間隔高速順フーリエ変換

CG 法, PCG 法の中で最も収束が速くなった T.Chan の前処理付き CG 法, Levinson-Rybyckii の

アルゴリズム, 直接法 (ガウスの消去法) を用いて不等間隔高速順フーリエ変換 ($\hat{f} = (F)^{-1}f = (F^*F)^{-1}F^*f$) の計算を行った.

F^* または F とベクトルの積を計算する際の精度を決めるパラメータとして倍精度を保証するものを選び計算を行い, N は 32 ~ 4096 とした. CG 法による収束判定は, $\|r\| / \|r^{(0)}\| < 10^{-14}$ とする. 右辺ベクトル f は乱数ベクトル $\text{Re}(\hat{f}), \text{Im}(\hat{f}) \in [0, 1]^N$ から直接計算 $f = F\hat{f}$ を行って与えた. 計算は $\{x_k\}$ と $\{f_j\}$ の 20 セットに対して行った.

誤差の評価は, 数値解を \hat{f} とするとき,

$$E = \max_{0 \leq k \leq N-1} |\hat{f}_k - \tilde{f}_k| / \max_{0 \leq k \leq N-1} |\hat{f}_k| \quad (12)$$

とした. 直接法ではガウスの消去法により逆行列を計算したあと, 直接計算で行列・ベクトル積を求めている.

結果は表 3 ~ 表 6 のようになった. 不等間隔点 x_k の取り方により結果が異なるため, 平均をとり計算時間や誤差にばらつきが出るものについては \pm の後に標準偏差を示す.

5. まとめ

- δ によらずガウスの消去法による直接計算は計算時間が最も短く誤差も小さいのだが, 予め計算できる部分 (ガウスの消去法) の計算時間が非常に大きい.
- $\delta \in [-0.1 : 0.1]$ の場合には $N = 1024$ 以上で計算時間が最も短く, 誤差もそれほど悪くならない CG 法がよい.
- $\delta \in [-0.5 : 0.5]$ の場合には N が比較的小さいときには計算時間が最も短く, 誤差もそれほど悪くならない Toeplitz の直接解法がよい. しかし, N が大きくなるにつれ誤差が大きくなる.

参考文献

- 1) R. H. Chan and M. K. Ng, Preconditioning Toeplitz Systems with Circulant Preconditioners, in: R. H. Chan, T. F. Chan and G. H. Golub eds. *Iterative Methods in Scientific Computing*, pp.243-270, Springer, 1997.
- 2) R. H. Chan, A. M. Yip and M. K. Ng, The Best Circulant Preconditioners for Hermitian Toeplitz Systems, *SIAM J. Numer. Anal.* Vol. 38(2000), pp. 876-896.
- 3) J.W.Cooley and J.W.Tukey, An algorithm for the machine computation of complex Fourier series, *Math. Comp.* Vol. 19(1965), pp. 297-301.
- 4) A. Dutt and V. Rokhlin, Fast Fourier Trans-

表 3 計算時間 (sec.): $\delta \in [-0.1 : 0.1]$

| N | CG | PCG T.Chan | Toeplitz Solver | Gaussian Elimination | Direct |
|------|-------------------|-------------------|-----------------|----------------------|---------|
| 32 | 0.00066 ± 0.00002 | 0.00073 ± 0.00002 | 0.00021 | 0.00082 | 0.00001 |
| 64 | 0.00129 ± 0.00003 | 0.00144 ± 0.00004 | 0.00045 | 0.00453 | 0.00004 |
| 128 | 0.00265 ± 0.00007 | 0.00293 ± 0.00007 | 0.00114 | 0.06481 | 0.00024 |
| 256 | 0.00538 ± 0.00007 | 0.00621 ± 0.00006 | 0.00314 | 0.48300 | 0.00084 |
| 512 | 0.01151 ± 0.00016 | 0.01338 ± 0.00023 | 0.00984 | 3.76850 | 0.00322 |
| 1024 | 0.02407 ± 0.00051 | 0.02860 ± 0.00073 | 0.03764 | 30.3005 | 0.01250 |
| 2048 | 0.05166 ± 0.00095 | 0.06216 ± 0.00117 | 0.13670 | 247.597 | 0.04945 |
| 4096 | 0.12367 ± 0.00043 | 0.14567 ± 0.00265 | 0.86441 | 2191.53 | 0.19750 |

表 4 計算時間 (sec.): $\delta \in [-0.5 : 0.5]$

| N | CG | PCG T.Chan | Toeplitz Solver | Gaussian Elimination | Direct |
|------|-------------------|-------------------|-----------------|----------------------|---------|
| 32 | 0.00120 ± 0.00005 | 0.00129 ± 0.00007 | 0.00021 | 0.00073 | 0.00001 |
| 64 | 0.00369 ± 0.00027 | 0.00337 ± 0.00031 | 0.00045 | 0.00418 | 0.00004 |
| 128 | 0.01160 ± 0.00108 | 0.00923 ± 0.00097 | 0.00113 | 0.06453 | 0.00024 |
| 256 | 0.03351 ± 0.00339 | 0.02596 ± 0.00259 | 0.00314 | 0.50433 | 0.00085 |
| 512 | 0.10516 ± 0.01062 | 0.07891 ± 0.00787 | 0.00984 | 3.96250 | 0.00326 |
| 1024 | 0.28184 ± 0.04674 | 0.22477 ± 0.03508 | 0.03928 | 29.2680 | 0.01265 |
| 2048 | 0.85546 ± 0.12841 | 0.66560 ± 0.09635 | 0.13432 | 261.030 | 0.04965 |
| 4096 | 2.89343 ± 0.42100 | 2.10864 ± 0.28332 | 0.87366 | 2194.85 | 0.19733 |

表 5 誤差: $\delta \in [-0.1 : 0.1]$

| N | CG | PCG T.Chan | Toeplitz Solver | Gaussian Elimination |
|------|----------------|----------------|-----------------|----------------------|
| 32 | 7.19±3.67 E-15 | 7.20±3.58 E-15 | 3.00±0.56 E-15 | 9.22±4.09 E-16 |
| 64 | 7.30±2.94 E-15 | 5.41±1.66 E-15 | 4.88±1.04 E-15 | 1.37±0.47 E-15 |
| 128 | 1.08±0.41 E-14 | 9.69±3.77 E-15 | 9.00±2.02 E-15 | 2.33±0.54 E-15 |
| 256 | 1.60±0.33 E-14 | 1.43±0.28 E-14 | 1.71±0.24 E-14 | 4.28±1.51 E-15 |
| 512 | 2.88±0.38 E-14 | 2.83±0.49 E-14 | 3.94±0.58 E-14 | 5.49±1.20 E-15 |
| 1024 | 5.75±0.72 E-14 | 5.68±0.66 E-14 | 8.46±1.11 E-14 | 1.07±0.30 E-14 |
| 2048 | 1.13±0.11 E-13 | 1.13±0.09 E-13 | 1.87±0.20 E-13 | 1.50±0.31 E-14 |
| 4096 | 2.39±0.22 E-13 | 2.39±0.21 E-13 | 3.90±0.40 E-13 | 2.53±0.67 E-14 |

表 6 誤差: $\delta \in [-0.5 : 0.5]$

| N | CG | PCG T.Chan | Toeplitz Solver | Gaussian Elimination |
|------|----------------|----------------|-----------------|----------------------|
| 32 | 2.26±0.94 E-14 | 2.12±0.78 E-14 | 1.75±0.72 E-14 | 2.99±3.84 E-15 |
| 64 | 3.64±1.21 E-14 | 3.62±1.34 E-14 | 2.90±1.83 E-14 | 4.20±3.25 E-15 |
| 128 | 5.49±2.10 E-14 | 4.85±1.69 E-14 | 8.23±9.79 E-14 | 7.63±3.95 E-15 |
| 256 | 1.09±0.81 E-13 | 1.12±0.83 E-13 | 2.08±2.44 E-13 | 1.60±0.80 E-14 |
| 512 | 2.35±1.09 E-13 | 2.26±1.01 E-13 | 3.67±2.52 E-13 | 3.90±3.41 E-14 |
| 1024 | 6.26±4.83 E-13 | 6.29±4.88 E-13 | 1.12±1.46 E-12 | 1.01±0.54 E-13 |
| 2048 | 2.12±1.45 E-12 | 2.12±1.54 E-12 | 1.98±7.48 E-11 | 1.96±1.72 E-13 |
| 4096 | 4.74±3.59 E-12 | 4.68±3.57 E-12 | 1.12±2.27 E-11 | 3.29±2.15 E-13 |

forms for Nonequispaced Data, *SIAM J. Sci. Comput.* Vol. 14(1993), pp. 1368-1393.

Vol. 40(1998), pp. 838-856.

- 5) G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, 1996.
- 6) 金子晃, 偏微分方程式入門, pp. 175-177, 東京大学出版会, 1998
- 7) A. F. Ware, Fast Approximate Fourier Transforms for Irregularly Spaced Data, *SIAM Rev.*