

Toeplitz逆行列を求めるトレンチゾーハー法の処理と応用

佐田, 大明 / SATA, Hiroaki

(出版者 / Publisher)

法政大学大学院理工学・工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学・工学研究科編

(巻 / Volume)

56

(開始ページ / Start Page)

1

(終了ページ / End Page)

4

(発行年 / Year)

2015-03-24

(URL)

<https://doi.org/10.15002/00011236>

Toeplitz 逆行列を求める Trench-Zohar 法の処理と応用

Processing and application of the Trench-Zohar method
to determine the Toeplitz matrix inverse

佐田 大明

Sata Hiroaki

指導教員 李 磊

法政大学大学院理工学研究科応用情報工学専攻修士課程

Solution of large-scale simultaneous linear equations have a high demand in various fields. For example, a linear system of the signal processing field and the autocorrelation matrix, etc. As the fast direct methods Trench-Zohar algorithms and E.H. Bareiss of algorithm is famous. In this paper, we studied Trench-Zohar direct method, and extend the algorithm to be able to adapt to a variety of matrix.

Key Words : Matrix Algorithm, Toeplitz matrix

1. はじめに

連立一次方程式 $Ax=b$ の係数行列が Toeplitz 行列の場合がある。これは信号処理分野の線形システムや自己相関行列などで現れる工学上有用な行列として知られ、これらの問題を解く際にアルゴリズムは重要な役割を果たす。アルゴリズムが改良されていくことで、安定性や速さなどの改善が見込まれる。この方程式の解法には直接法や反復法等、様々なものがある。本研究では直接法について紹介をしていくが、その中で知られているものとしては Trench-Zohar のアルゴリズムや E.H. Bareiss のアルゴリズムが挙げらる。本稿では Trench-Zohar 法に研究対象をおき、Trench-Zohar 法を拡張し様々な行列へ適応させると共に、それらの結果を報告する。

2. Toeplitz 行列

Toeplitz 行列は以下のような対角一定行列のことを指す。

$$T = \begin{bmatrix} a & e & f & g \\ b & a & e & f \\ c & b & a & e \\ d & c & b & a \end{bmatrix}$$

連立一次方程式 $Ax=b$ を解くとき、 A が Toeplitz 行列であると通常よりも解き易いと言われ、2 つの Toeplitz 行列の加算は $O(n)$ でなされ Toeplitz 行列とベクトルの乗算は $O(n \log n)$ 、2 つの Toeplitz 行列の乗算は $O(n^2)$ の時間でなされる。

連立一次方程式を解くとき、Toeplitz 行列以外の行列の場合 $O(n^3)$ であり、Toeplitz 行列の時は $O(n^2)$ ができる。

3. Trench-Zohar のアルゴリズム

まず、Toeplitz 行列を以下のようにする。

$t_k (k = 0 \sim n-1)$, $\tau_k (k = 1 \sim n-1)$ は実数とする。

r と c はそれぞれ定数とする。

$$T = \begin{bmatrix} t_0 & \tau_1 & \tau_2 & \cdots & \tau_{n-1} \\ t_1 & t_0 & \tau_1 & \cdots & \tau_{n-2} \\ t_2 & t_1 & t_0 & \cdots & \tau_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & \cdots & t_{n-2} & t_{n-1} & t_0 \end{bmatrix}$$

T の逆行列が存在すると仮定し、その行列を T^{-1} とする。またこの逆行列を B とし、その各要素を $B_{i,j} (i, j = 1 \sim n)$ とする。

$$1. r_{11} = r_1/t_0 \quad c_{11} = t_1/t_0$$

2. For $i=1 \sim k$ do ($k=n-1$)

$$\alpha_k = t_0 - \sum_{j=1}^k r_{k,j} t_j$$

For $i = 1 \sim k$ do ($k=n-2$)

$$r_{k+1,k+1} = 1/\alpha_k (r_{k+1} - \sum_{j=1}^k r_{k,k+1-j} r_j)$$

$$r_{k+1,i} = r_{k,i} - c_{k,k+1-i} r_{k+1,k+1}$$

$$c_{k+1,k+1} = 1/\alpha_k (t_{k+1} - \sum_{j=1}^k c_{k,k+1-j} t_j)$$

$$c_{k+1,i} = c_{k,i} - r_{k,k+1-i} c_{k+1,k+1}$$

3. For $i, j=1 \sim n-1$ do

$$B_{11} = \frac{1}{\alpha_{n-1}}$$

$$B_{1,j+1} = -\frac{1}{\alpha_{n-1}} r_{n-1,j}$$

$$B_{i+1,1} = -\frac{1}{\alpha_{n-1}} c_{n-1,i}$$

$$B_{i+1,j+1} = B_{i,j} + \frac{1}{\alpha_{n-1}} (c_{n-1,i} r_{n-1,j} - r_{n-1,n-i} c_{n-1,n-j})$$

このアルゴリズムは逆行列の成分を計算するので毛お産量は $\frac{13}{2}n^2$ となる。

4. 提案手法

連立一次方程式の問題では対称 Toeplitz 行列が多く現れる.一般的な行列での計算量は $O(n^2)$ となっているが,提案手法は従来の手法を特殊解へ適用するアルゴリズムである.この手法は計算量 $O(n^2)$ というものは変わらないが,従来の手法よりも係数を減らすことが可能となる.提案手法は,前者で Trench-Zohar のアルゴリズムを紹介したが,これを様々な行列にも適応できるように拡張する.

5. Trench-Zohar 法による対称 Toeplitz 行列の特殊解

従来の手法は,Toeplitz 行列に対してのアルゴリズムである.これを実対称 Toeplitz 行列に適用する.この時,行列の特性により,実対称 Toeplitz 行列の逆行列は実中心対称行列となることから,アルゴリズムの簡略化が可能である.

以下のような対称行列があるとする.

$$T = \begin{bmatrix} t_0 & t_1 & t_2 & \cdots & t_{n-1} \\ t_1 & t_0 & t_1 & \cdots & t_{n-2} \\ t_2 & t_1 & t_0 & \cdots & t_{n-3} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ t_{n-1} & \cdots & t_{n-2} & t_{n-1} & t_0 \end{bmatrix}$$

$$1. r_{11} = t_1/t_0$$

$$2. \text{For } i=1 \sim k \text{ do } (k=n-1)$$

$$\alpha_k = t_0 - \sum_{j=1}^k r_{k,j} t_j$$

$$\text{For } i = 1 \sim k \text{ do } (k=n-2)$$

$$r_{k+1,k+1} = 1/\alpha_k (t_{k+1} - \sum_{j=1}^k r_{k,k+1-j} t_j)$$

$$r_{k+1,i} = r_{k,i} - r_{k,k+1-i} r_{k+1,k+1}$$

$$3. \text{For } i=1 \sim n-1 \quad j=1 \sim i \text{ do}$$

$$B_{11} = \frac{1}{\alpha_{n-1}}$$

$$B_{i+1,1} = -\frac{1}{\alpha_{n-1}} r_{n-1,i}$$

$$B_{i+1,j+1} = B_{i,j} + \frac{1}{\alpha_{n-1}} (r_{n-1,i} r_{n-1,j} - r_{n-1,n-i} r_{n-1,n-j})$$

行列 T は実対称 Toeplitz 行列であることから r_{nm} と c_{nn} の値が同値となる.このことから Trench-Zohar のアルゴリズムが上記のように簡略化させることが出来る.また,このアルゴリズムは行列の $\frac{1}{4}$ の成分を計算するので $\frac{9}{4} n^2$ となる.

6. アルゴリズムの拡張 1

1 つ目は以下のように行列が $n \times n$ 且つ内部が 2×2 のブロック行列である時のアルゴリズム拡張を行う.

例として以下の行列があったとする.

$$T = \begin{bmatrix} a & c & d & f & g & h & i & k \\ c & b & e & d & h & g & j & i \\ d & f & a & c & d & f & g & h \\ e & d & c & b & e & d & h & g \\ g & h & d & f & a & c & d & f \\ h & g & e & d & c & b & e & d \\ i & k & g & h & d & f & a & c \\ j & i & h & g & e & d & c & b \end{bmatrix}$$

$$T = \begin{bmatrix} T_0 & T_1 & T_2 & T_3 \\ T_1 & T_0 & T_1 & T_2 \\ T_2 & T_1 & T_0 & T_1 \\ T_3 & T_2 & T_1 & T_0 \end{bmatrix}$$

これらは対称行列であるため上記のように変換できる.

$$1. S_{11} = T_1 \cdot T_0^{-1}$$

$$2. \text{For } i=1 \sim k \text{ do } (k=n-1)$$

$$\alpha_k = T_0 - \sum_{j=1}^k S_{k,j} T_j$$

$$\text{For } i = 1 \sim k \text{ do } (k=n-2)$$

$$S_{k+1,k+1} = (T_{k+1} - \sum_{j=1}^k S_{k,k+1-j} T_j) \alpha_k^{-1}$$

$$S_{k+1,i} = S_{k,i} - S_{k,k+1-i} S_{k+1,k+1}$$

$$3. \text{For } i=n-1 \quad j=1 \sim i \text{ do}$$

$$B_{11} = \alpha_{n-1}^{-1}$$

$$\text{if } ((i+1) \% 2 = 0)$$

$$B_{i+1,1} = -S_{n-1,i} \alpha_{n-1}^{-1}$$

$$\text{else}$$

$$B_{i+1,1} = -\alpha_{n-1}^{-1} S_{n-1,i}$$

$$B_{i+1,j+1} = B_{i,j} + (-B_{i+1,1} S_{n-1,j} + B_{n-i+1,1} S_{n-1,n-j})$$

7. アルゴリズムの拡張 2

2 つ目は以下のように行列が $n \times n$ 且つ内部が $(n/2) \times (n/2)$ のブロック行列である時のアルゴリズム拡張を行う.例として以下の行列があったとする.

$$T = \begin{bmatrix} a_1 & b_1 & b_2 \cdots & b_{n/2-1} & e_1 & f_1 & f_2 \cdots & f_{\frac{n}{2}-1} \\ a_2 & a_1 & b_1 \cdots & b_{\frac{n}{2}-2} & e_2 & e_1 & f_1 \cdots & f_{\frac{n}{2}-2} \\ \vdots & a_2 & \ddots & \vdots & \vdots & e_2 & \ddots & \vdots \\ a_{n/2} & a_{\frac{n}{2}-1} & \cdots & a_1 & e_{n/2} & e_{\frac{n}{2}-1} & \cdots & e_1 \\ c_1 & d_1 & d_2 \cdots & d_{\frac{n}{2}-1} & g_1 & h_1 & h_2 \cdots & h_{\frac{n}{2}-1} \\ c_2 & c_1 & d_1 \cdots & \vdots & g_2 & g_1 & h_1 \cdots & h_{\frac{n}{2}-2} \\ \vdots & c_2 & \ddots & d_{\frac{n}{2}-2} & \vdots & g_2 & \ddots & \vdots \\ c_{n/2} & c_{\frac{n}{2}-1} & \cdots & c_1 & g_{n/2} & g_{\frac{n}{2}-1} & \cdots & g_1 \end{bmatrix}$$

$$T = \begin{bmatrix} T_0 & T_1 \\ T_1 & T_0 \end{bmatrix}$$

上記のように,行列の要素が $n \times n$ であっても,それが $(n/2) \times (n/2)$ のブロック行列であった場合,以下のようにアルゴリズムが書き換えられる,

$$1.S_{11} = T_1 \cdot T_0^{-1}$$

$$2.\alpha_1 = T_0 - S_{1,1}T_1$$

$$3.B_{11} = \alpha_{n-1}^{-1}$$

$$B_{2,1} = B_{1,2} = -\alpha_1^{-1}S_{1,1}$$

$$B_{11} = B_{22}$$

上記のようにブロック行列が T_0, T_1 のみで表せる場合, アルゴリズムは非常に短く表すことが出来る.

尚,拡張アルゴリズム 1,2は二重 Toeplitz 行列の時も使用できる.つまり以下のような行列 T があった時もアルゴリズムは同じように使用できる.

8. 数値実験

テスト行列はまず $n \times n$ の実対称 Toeplitz 行列を作成し, それぞれのアルゴリズムで行った.従来手法は一定対角 Toeplitz 行列,特殊解は実対称 Toeplitz 行列,提案手法 1 は 2×2 の対称ブロック Toeplitz 行列,提案手法 2 は $(n/2) \times (n/2)$ のブロック Toeplitz 行列でそれぞれ比較を行った. また各要素は 1 から 10 の乱数で生成した.また,アルゴリズムの処理の後に求められた逆行列のノルムを計測し比較した.

9. 実験結果

表 1.従来と特殊解の速さの比較

	1000	2000	3000	4000	5000	6000
従来	0.010	0.042	0.088	0.18	0.28	0.42
特殊解	0.008	0.03	0.07	0.118	0.182	0.278

表 2.従来と特殊解のノルムの比較

	1000	2000	3000	4000	5000	6000
従来	34	43	56.9	86.2	88.5	141.8
特殊解	42.6	38.2	82	112.4	73.6	145

表 3.提案 1 と提案 2 の速さ

	1000	2000	3000	4000	5000	6000
提案 1	0.015	0.052	0.14	0.28	0.52	0.98
提案 2	1.2	5.1	21.3	83.1	338.2	1469.7

表 4.提案 1 と提案 2 のノルム

	1000	2000	3000	4000	5000	6000
提案 1	117.9	186.5	273	367.7	513.4	702.4
提案 2	180.3	297.5	435.8	602.1	804.5	1021

表 5.提案 1 と提案 2 の速さ(二重 Toeplitz)

	1000	2000	3000	4000	5000	6000
提案 1	0.011	0.048	0.12	0.29	0.57	1.03
提案 2	1.05	4.8	19.3	79.5	521	1490.3

表 6.提案 1 と提案 2 のノルム(二重 Toeplitz)

	1000	2000	3000	4000	5000	6000
提案 1	89.7	174.2	263.1	372.5	507.8	687.5
提案 2	107.8	215.7	368.6	580.1	793.8	1027.4

これらを図で表すと以下ようになる.

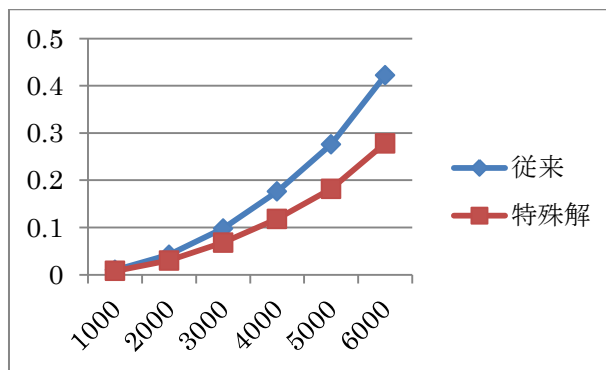


図 1.従来手法と特殊解との速さの比較

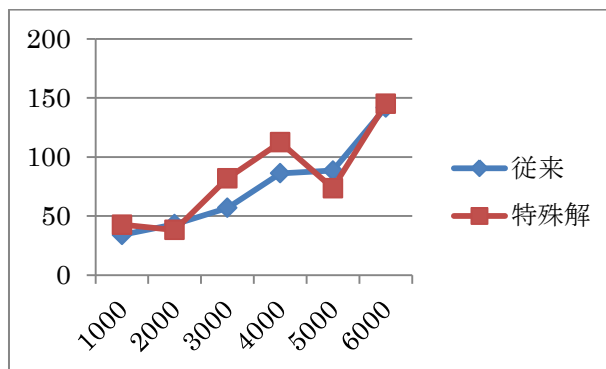


図 2.従来手法と特殊解とのノルムとの比較

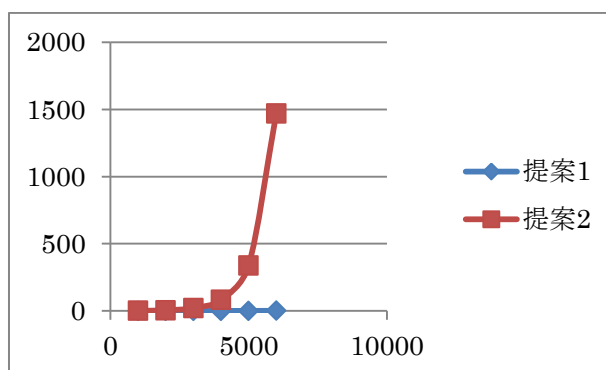


図 3.提案手法 1 と提案手法 2 の速さ

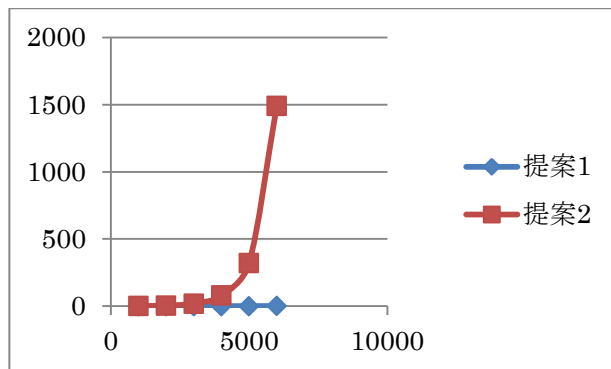


図 4.提案手法 1 と提案手法 2 の二重 Toeplitz 時の速さ

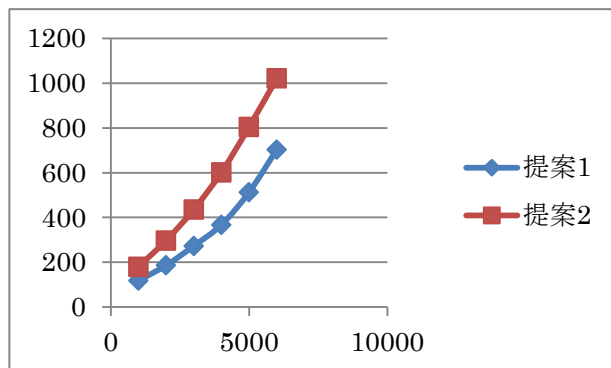


図 5.提案手法 1 と提案手法 2 のノルム

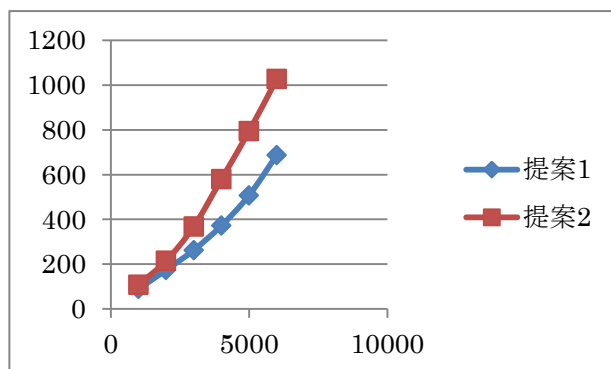


図 6.提案手法 1 と提案手法 2 の二重 Toeplitz 時のノルム

10. まとめ

実験結果より,Trench-Zohar のアルゴリズムを拡張したことで様々な行列に対し適応させることが出来ると分かった.しかしながら,提案手法の結果を見るとノルムが高くなってしまい,余り精度の高いものとは言えない結果となってしまった.また結果内容は平均をとったことより,行列によっては適正できないものもあることが分かった.精度を上げるためにはよりアルゴリズムの細かな点や逆行列を求めることを考慮したうえで改良するべきだと考える.

11. 謝辞

本研究を遂行するにあたり,多大なるご指導を頂きました法政大学教授李磊教授に厚くお礼を申し上げます.また,本研究室の李研究室の行列班にも感謝いたします.さらに,様々な助言を頂きました本研究室の大学院生の諸先輩方,並びに 4 年生諸氏にも感謝いたします.

参考文献

- [1] Tadao Nakamura,Parallel Processing For Solving Toeplitz. Systems on NEC SX-3/44R Super computer. Computational Mechanics'95.
- [2] E.H.Bareiss,Numerical Solution of Linear Equations with Toeplitz and vector Toeplitz matrices,Numer.math,13:5(1969),pp404-424
- [3] W.F.Trench,An algorithm for the Inversion of Finite Toeplitz Matrics, J.SIAM pp.515-522.