NUMERICAL METHOD TO A CLASS OF BOUNDARY VALUE PROBLEMS

by

Yu-Yang QIU *

College of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou, China

Original scientific paper
https://doi.org/10.2298/TSCI1804877Q

A class of boundary value problems can be transformed uniformly to a least square problem with Toeplitz constraint. Conjugate gradient least square, a matrix iteration method, is adopted to solve this problem, and the solution process is elucidated step by step so that the example can be used as a paradigm for other applications.

Key words: *boundary value problems, the least square problem, Toeplitz constraint, conjugate gradient least square, matrix iteration*

Introduction

In this paper, we consider a numerical solution to a class of boundary value problems which have the following form:

$$\begin{cases} Lu(x, y) = f(x, y), (x, y) \in \Omega \\ Bu(x, y) = g(x, y), (x, y) \in \partial\Omega \end{cases} \tag{1}$$

where L is a linear differential operator, B – a boundary operator, $f(x, y)$ and $g(x, y)$ are two known functions, and $\Omega \in R^2$ is an open bounded domain with boundary $\partial\Omega$. The problem (1) arises in many fields such as fluid mechanics and thermal science, and has attracted more and more researchers in recent decades [1-6]. By the method of fundamental solutions [1], the problem (1) can be transformed uniformly to a set of linear equations $\mathrm{C}_n\vec{\mathrm{z}} = \vec{\mathrm{b}}$ with unknown vector $\vec{\mathrm{z}}$, the circulant or block-circulant matrix C_n, and the given vector $\vec{\mathrm{b}}$. So the key to the problem (1) is to solve the inverse of the circulant matrix C_n, that is we want to find a matrix X, such that $\mathrm{C}_n\mathrm{X} = \mathrm{I}_n$, or $\mathrm{XC}_n = \mathrm{I}_n$. Note that the previous equations can be regarded as the special cases of the corresponding least square problem (with zero residual), together with the inverse of the circulant matrix C_n is also the Toeplitz matrix [7-11], so it can be generalized to solve the following constrained least square problem:

$$\min_{\mathrm{X}\in \mathrm{S}_{\mathrm{T}_n}} \left\| \mathrm{AXB}^T - \mathrm{F} \right\|_\mathrm{F} \tag{2}$$

where A, B $\in \mathrm{R}^{n\times n}$, $\mathrm{S}_{\mathrm{T}_n}$ is the Toeplitz matrix set with order n. Obviously, if we choose A = C_n, B = I_n, and F = I_n, the problem (2) is equivalent to solve the inverse of the circulant matrix C_n where S_{T_n} is the circulant matrix space.

* Author's, e-mail: yuyangqiu77@163.com

We adopt an iteration algorithm called conjugate gradient least square (CGLS) to solve the problem (2) [7, 12-15], our ideas stem from the following facts.

- The least square problem (2) is equivalent to its norm equation whose coefficient matrix is positive [7].
- The numerical solutions to the norm equation can be solved by Krylov subspace methods, we choose CGLS as an example in this paper.
- The main products in CGLS are matrix-vector and matrix-matrix, however the involved Kronecker product will increase the computational complexity [12], so we release Kronecker product to get the corresponding matrix form iteration. The matrix iteration reveal our methods are effective and feasible.

Notation. In the rest of paper, $R^{m\times n}$ denotes the space of real $m\times n$ matrix. The notation $\otimes$ is Kronecker product, and I_n is the identity matrix with order n. For any matrix $X = [x_1, x_2,\ldots, x_n] \in R^{m\times n}$, X^T stands for its transpose, and $\mathrm{vec}(X) = [x_1^T, x_2^T,\ldots, x_n^T]^T$ is its long vector expanded by columns. For any vector $v \in R^n$, $v(i)$ is its i$^{\text{th}}$ component. The norm $\|\ \|_F$ is the Frobenius norm of matrix, while $\|\ \|_2$ is 2-norm of vector or matrix.

The co-ordinate and constrained matrix

The Toeplitz matrix T_n with n order has the following form:

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

where t_i, $i = 1 - n, 2 - n,\ldots, n - 1$ are parameters. Denote the Toeplitz matrix set S_{T_n} by $S_{T_n} = \{T_n | T_n \text{ is Toeplitz matrix with order } n\}$.

Suppose

$$G = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{4}$$

we have

$$T_n = t_{1-n}(G^{n-1})^T + \ldots + t_{-1}G^T + t_0G^0 + t_1G^1 + \ldots + t_{n-1}G^{n-1} \tag{5}$$

Let

$$G^i = (G^{|i|})^T, \quad i = 1 - n, 2 - n, \ldots, -1$$

then eq. (5) is equivalent to:

$$T_n = t_{1-n}G^{1-n} + t_{2-n}G^{2-n} + \ldots + t_{n-1}G^{n-1} = \sum_{i=1-n}^{n-1} t_iG^i$$

Denote the co-ordinate map by:

$$g_c(T_n) = (t_{1-n}, t_{2-n}, \ldots, t_{n-1})^T$$

and constrained matrix by:

$$C_T = [\mathrm{vec}(G^{1-n})^T, \mathrm{vec}(G^{2-n})^T, \ldots, \mathrm{vec}(G^{n-1})]$$

we have

$$tr[(G^i)^T G^j] = \begin{cases} n-|i| & i = j \\ 0 & i \neq j \end{cases}$$

and

$$\mathrm{vec}(T_n) = C_T g_c(T_n)$$

Obviously, $\{G^i\}_{i=1-n}^{n-1}$ are the basis of the Toeplitz matrix space S_{T_n}, and $\{t_i\}_{i=1-n}^{n-1}$ are the corresponding co-ordinate.

For any matrix $Z \in R^{n\times n}$, we begin to represent its co-ordinate $g_c(Z)$. It is not difficult to verify:

$$\mathrm{trace}(Z^T G^i) = \begin{cases} \sum_{k=1}^{n-i} z_{k,k+i}, & i = 0, 1, 2, \cdots n-1 \\ \sum_{k=|i|+1}^{n} z_{k,k+i}, & i = 1-n, 2-n, \cdots, -1 \end{cases}$$

Denote by:

$$tr(Z,i) = \sum_{\beta-\alpha=i} Z_{\alpha,\beta}, \qquad i = 1-n, \quad 2-n, \cdots, n-1$$

then

$$\mathrm{tr}(Z,i) = \mathrm{trace}(Z^T G^i)$$

So

$$[g_c(Z)]_i = \mathrm{trace}\left(\frac{Z^T G^i}{\|G^i\|_F^2}\right) = \frac{\mathrm{tr}(Z,i)}{n-|i|}, \qquad i = 1-n, \quad 2-n, \cdots, n-1$$

Moreover:

$$[\mathrm{vec}(G^i)]^T \mathrm{vec}(Z) = \mathrm{trace}[(G^i)^T Z], \qquad i = 1-n, \quad 2-n, \cdots, n-1$$

Hence, we have the following theorem.

Theorem 1. Let Toeplitz matrix G be defined by eq. (4), for any $Z \in R^{n\times n}$, we have

$$[g_c(Z)]_i = \frac{\mathrm{tr}(Z,i)}{n-|i|}, \qquad i = 1-n, \quad 2-n, \cdots, n-1$$

and

$$[\mathrm{vec}(G^i)]^T \mathrm{vec}(Z) = \mathrm{trace}[(G^i)^T Z], \qquad i = 1-n, \quad 2-n, \cdots, n-1$$

The matrix iteration CGLS

Denote by:

$$\mathrm{M} = (\mathrm{B} \otimes \mathrm{A})\mathrm{C}_\mathrm{T} \quad \text{and} \quad f = \mathrm{vec}(\mathrm{F})$$

then eq. (2) is equivalent to:

$$\min_x \|\mathrm{M}x - f\|_2 \tag{6}$$

whose norm equation is:

$$\mathrm{M}^T \mathrm{M}x = \mathrm{M}^T f \tag{7}$$

Equation (7) can be solved by the following iteration CGLS [7, 13].

Iteration CGLS

(1) *Initialization.*

Set

$$x_0 = x_{\mathrm{int}}, \qquad r_0 = \mathrm{M}^T f - \mathrm{M}^T M x_{\mathrm{int}}$$

and
$$p_0 = r_0, \qquad \rho_0 = \| r_0 \|_2^2, \qquad k = k+1$$

(2) *Iteration.* For $i = 1, 2, \cdots$ until convergence:

$$\alpha_k = \frac{\rho_{k-1}}{p_{k-1}^T \mathrm{M}^T \mathrm{M} p_{k-1}}, \qquad x_k = x_{k-1} + \alpha_k p_{k-1}, \quad r_k = r_{k-1} - \alpha_k \mathrm{M}^T \mathrm{M} p_{k-1}$$

$$\rho_k = r_k^T r_k, \qquad \beta_k = \frac{\rho_k}{\rho_{k-1}}, \qquad p_k = r_k + \beta_k p_{k-1}$$

Applying iteration CGLS on eq. (7), we get the corresponding algorithm CGLS_v (one can turn to [7, 13, 14] for details). There are two basic operations Mv and $\mathrm{M}^T u$ in iteration CGLS_v with:

$$v \in R^{2n-1}, \qquad u \in R^{n^2}, \qquad \mathrm{M} \in R^{n^2 \times n^2}$$

When n increases, they will be very big since the matrix M has Kronecker product, which will increase the computational complexity.

In this section, we want to re-write iteration CGLS_v to CGLS_M whose product is only matrix-matrix by releasing Kronecker product. For this end, we should represent the long vector Mv and $\mathrm{M}^T u$ by suitable matrices.

For the vector $v \in \mathrm{R}^{2n-1}$ the Toeplitz matrix $\mathrm{V} \in \mathrm{S}_{\mathrm{T}_n}$ can be set by:

$$\mathrm{V} = \sum_{i=1-n}^{n-1} v(i)\mathrm{G}^{i-1}$$

then we have:

$$\mathrm{vec}(\mathrm{V}) = \mathrm{C}_\mathrm{T} v$$

So

$$\mathrm{M}v = (\mathrm{B} \otimes \mathrm{A})\mathrm{C}_\mathrm{T} v = (\mathrm{B} \otimes \mathrm{A})\mathrm{vec}(\mathrm{V}) = \mathrm{vec}(\mathrm{AVB}^T)$$

That is, the matrix form of vector Mv is AVB^T.

Note that:

$$\mathrm{M}^T u = \mathrm{C}_\mathrm{T}^T(\mathrm{B}^T \otimes \mathrm{A}^T)\mathrm{vec}(\mathrm{U}) = \mathrm{C}_\mathrm{T}^T \mathrm{vec}(\mathrm{A}^T \mathrm{UB})$$

Denote by:

$$\bar{\mathrm{U}} = \mathrm{A}^T \mathrm{UB}$$

we should compute the vector $C_T^T \mathrm{vec}(\bar{\mathrm{U}})$ and get its matrix form to represent $\mathrm{M}^T u$.

With *Theorem 1*, we have:

$$C_T^T vec(\bar{\mathrm{U}}) = \{\mathrm{trace}[(\mathrm{G}^{1-n})^T \bar{\mathrm{U}}], \quad \mathrm{trace}[(\mathrm{G}^{2-n})^T \bar{\mathrm{U}}], \cdots, \quad \mathrm{trace}[(\mathrm{G}^{n-1})^T \bar{\mathrm{U}}]\}^T$$

Hence, the matrix form of vector $\mathrm{M}^T u$ can be chosen by:

$$P_C(M^T u) = \sum_{i=1-n}^{n-1} \mathrm{trace}[(\mathrm{G}^i)^T \bar{\mathrm{U}}]$$

So algorithm CGLS_v can be re-written as its matrix form iteration CGLS _M.

Iteration CGLS_M

(1) *Initialization.*

$$\text{Set } \mathrm{X}_0 = \mathrm{X}_{\mathrm{int}}, \qquad R_0 = P_C(\mathrm{M}^T f - \mathrm{M}^T \mathrm{AX}_0 \mathrm{B}^T)$$

and
$$P_0 = R_0, \quad \rho_0 = \| g_c(R_0) \|_2^2, \quad k = k + 1$$

(2) *Iteration*. For $i = 1, 2, \cdots$ until convergence:

$$\tilde{P} = P_C[\mathrm{M}^T \mathrm{M}\mathrm{vec}(\mathrm{P})], \qquad W = P_C[\mathrm{C}_\mathrm{T}^T \mathrm{vec}(\tilde{P})]$$

$$\alpha = \frac{\rho_0}{g_c(\mathrm{P})}, \qquad \mathrm{X} = \mathrm{X} + \alpha \mathrm{P}, \qquad \mathrm{R} = \mathrm{R} - \alpha \mathrm{W}$$

$$\rho_1 = \|g_c(\mathrm{R})\|_2^2, \qquad \beta = \frac{\rho_1}{\rho_0}, \qquad \rho_0 = \rho_1, \qquad \mathrm{P} = \mathrm{R} + \beta \mathrm{P}$$

Check the convergence by $\|g_c(\mathrm{R})\|_2 < \tau$ with given number τ.

Numerical examples

In this section, we present two numerical examples to illustrate the effectiveness of our proposed iteration. For the test matrices A, B, the right-hand side matrix F and the residual error ε are set by:

$$\mathrm{F} = \mathrm{AXB}^T, \qquad \varepsilon = \left\|\mathrm{F} - \mathrm{AXB}^T\right\|_F$$

with $\mathrm{X} \in \mathrm{S}_{T_n}$, so the expected error ε should be zero.

We report the numerical results by iteration CGLS_M. All examples are performed by mathematical software on a personal computer of the Intel Core CPU i5 5300U with 4G memory.

Example 1. In this example, we test the residual error of least square problem (2), the matrices A, B are set by:

$$\mathrm{A} = \mathrm{U}_A \mathrm{diag}(\sigma_A^1, \sigma_A^2, \cdots, \sigma_A^n)\mathrm{V}_A, \qquad \mathrm{B} = \mathrm{U}_B \mathrm{diag}(\sigma_B^1, \sigma_B^2, \cdots, \sigma_B^n)\mathrm{V}_B$$

where singular values $\sigma_A^i, \sigma_B^i, \cdots i = 1, 2, \cdots, n$ are randomly chosen and the orthogonal matrix: U_A, U_B, V_A, and V_B are set by:

$$[\mathrm{U}_A, \mathrm{temp}] = qr(1 - 2\mathrm{rand}(n)), \qquad [\mathrm{V}_A, \mathrm{temp}] = qr(1 - 2\mathrm{rand}(n))$$

$$[\mathrm{U}_B, \mathrm{temp}] = qr(1 - 2\mathrm{rand}(n)), \qquad [\mathrm{V}_B, \mathrm{temp}] = qr(1 - 2\mathrm{rand}(n))$$

The Toeplitz matrix $\mathrm{X} \in \mathrm{S}_{T_n}$ is set by (5).

For the given stopping criteria $\tau = 10^{-11}$, the iteration numbers and the CPU time seem to depend on the matrix size *n*. As *n* increases, the CPU time grows quickly, but ε changes a little. In tab. 1, we list the CPU time, ε, and iteration numbers for different values of *n*, respectively.

Example 2. In this example, we consider the inverse of circulant matrix. The circulant matrix $\mathrm{X} \in \mathrm{S}_{T_n}$ is set by eq. (5) with suitable parameters. The matrix size varies from n = 20 to n = 500 and the stop stopping criteria $\tau = 10^{-11}$, respectively. In tab. 2, we list the CPU time and error residual for different values of *n*, respectively.

Table 1. The CGLS_M for least square problem with Toeplitz constraint

n	ε	CPU time	Iteration numbers
20	$7.06\cdot10^{-12}$	0.02	56
50	$1.57\cdot10^{-12}$	0.26	236
100	$7.34\cdot10^{-11}$	4.01	540
200	$3.74\cdot10^{-11}$	35.32	781
300	$1.98\cdot10^{-10}$	102.44	1187

Table 2. The CGLS_M for the inverse of circulant matrix

n	ε	CPU time	Iteration numbers
20	$5.01\cdot10^{-11}$	0.02	15
50	$1.58\cdot10^{-11}$	0.26	40
100	$7.34\cdot10^{-11}$	1.14	70
200	$5.66\cdot10^{-10}$	9.36	145
300	$3.98\cdot10^{-10}$	41.67	187

Conclusion

This paper reports the iteration CGLS for least square problem with Toeplitz matrix constraint, whose special case is the inverse of the circulant matrix. We can use it to solve a class of boundary value problems. Compared with the existing methods, our iteration only involves matrix-matrix product and it is easy to be implemented.

Acknowledgment

The research was supported in part by the National Natural Science Foundation of China, Natural Science Foundation of Zhejiang Province and Foundation for Young Talents of ZJGSU under Grant Nos. Y6110639, 11201422, 11571312, 1020XJ1314019.

References

[1] Liu, X. Y., *et al.*, Circulant Matrix and Conformal Mapping for Solving Partial Differential Equations, *Computers & Mathematics with Applications, 68* (2014), 3, pp. 67-76
[2] He, J.-H., Effect on Temperature on Surface Tension of a Bubble and Hierarchical Ruptured Bubbles for Nanofiber Fabrication, *Thermal Science*, *16* (2012), 1, pp. 327-330
[3] He, J.-H., Variational Iteration Method – a Kind of Non-Linear Analytical Technique: Some Examples, *International Journal of Non-Linear Mechanics, 34* (1999), 4, pp. 699-708
[4] Salimi, S., *et al.*, An Analytical Solution to the Thermal Problems with Varying Boundary Conditions Around a Moving Source, *Applied Mathematical Modelling, 40* (2016), 13-14, pp. 6690-6707
[5] Abassy T. A., *et al.*, Toward a Modified Variational Iteration Method, *Journal of Computational and Applied Mathematics, 207* (2007), 1, pp. 137-147
[6] Lu, J. F., Numerical Analysis of the (2+1)-Dimensional Boiti-Leon-Pempinelli Equation, *Thermal Science, 21* (2017), 4, pp. 1657-1663
[7] Golub, G. H., Van Loan, C. F., Matrix Computations, in: *Special Linear Systems* (3rd ed; Cha. 4), Johns Hopkins University Press, Baltimore, Md., USA, 3rd edition, 1996
[8] Chan, R. H., Jin, X. Q., *An Introduction to Iterative Toeplitz Solvers*, Society for Industrial and Applied Mathematics, Philadelphia, Penn, USA, 2007
[9] Chan, T. F., Hansen, P., A Look-Ahead Levinson Algorithm for Indefinite Toeplitz Systems, *SIAM J. Matrices Anal. Appl. 13* (1992), 2, pp. 490-506
[10] Carmona, A., The Inverses of Some Circulant Matrices, *Applied Mathematics and Computation, 270* (2015), 1, pp. 785-793
[11] Kailath, T., Sayed, A. H., Displacement Structure: Theory and Applications, *SIAM Reviews, 37* (1995), 3, pp. 297-386
[12] Fausett, D. W., Fulton, C. T., Large Least Squares Problems Involving Kronecker Products, *SIAM J. Matrix Anal. Appl. 15* (1994), 1, pp. 219-227
[13] Hestenes, M. R., Stiefel, E., Methods of Conjugate Gradients for Solving Linear System, *J. Res. Nat. Bur. Stand. 49* (1952), 1, pp. 409-436
[14] Stiefel, E., Stabilization without using Gaussian Normal Equations (in German) *Wiss. Z. Tech. Hochsch. Dresden, 2* (1952/53), pp. 441-442
[15] Paige, C. C., Saunders, A., Sparse Linear Equations and Sparse Least Squares Problems, *ACM Transactions on Mathematical Software*, *8* (1982), 2, pp. 195-209

Paper submitted: February 20, 2017
Paper revised: October 10, 2017
Paper accepted: October 18, 2017