

DISPLACEMENT STRUCTURE

THOMAS KAILATH AND ALI H. SAYED

Many problems in engineering and applied mathematics ultimately require the solution of $n \times n$ linear systems of equations. For small-size problems, there is often not much else to do except to use one of the already standard methods of solution such as Gaussian elimination. However, in many applications, n can be very large ($n \sim 1000$, $n \sim 1,000,000$) and, moreover, the linear equations may have to be solved over and over again, with different problem/model parameters, till a satisfactory solution to the original physical problem is obtained. In such cases, the $O(n^3)$ burden, *i.e.*, the number of flops required to solve an $n \times n$ linear system of equations, can become prohibitively large. This is one reason why one seeks in various classes of applications to identify special/characteristic matrix structures that may be assumed in order to reduce the computational burden.

The most obvious structures are those that involve explicit patterns among the matrix entries such as Toeplitz, Hankel, Vandermonde, Cauchy, and Pick matrices. Several fast algorithms have been devised over the years to exploit these special structures. However, even more common than these explicit matrix structures, are matrices in which the structure is implicit. For example, in certain least-squares problems one often encounters products of Toeplitz matrices; these products are not generally Toeplitz, but on the other hand, they are not “unstructured”. Similarly, in probabilistic calculations the matrix of interest is often not a Toeplitz matrix, but rather its inverse, which is rarely Toeplitz itself, but of course is not unstructured: its inverse is Toeplitz. It is well known that $O(n^2)$ flops suffice to solve linear systems of equations with an $n \times n$ Toeplitz coefficient matrix; a question is whether one will need $O(n^3)$ flops to invert a non-Toeplitz coefficient matrix whose inverse is known to be Toeplitz? When pressed, one’s response clearly must be that it is conceivable that $O(n^2)$ flops will suffice, and this is in fact true.

Such problems suggest the need for a quantitative way of defining and identifying structure in matrices. Over the years, starting with [A1], it was found that an elegant and useful way to do so is the concept of *displacement structure*. This concept has also been useful for a host of problems apparently far removed from the solution of linear equations, such as the study of constrained and unconstrained rational interpolation, maximum entropy extension, signal detection, digital filter design, nonlinear Riccati differential equations, inverse scattering, certain Fredholm integral equations, etc (see [A2]-[A3] and the many references therein).

For motivation, consider an $n \times n$ Hermitian Toeplitz matrix, $T = [c_{i-j}]_{i,j=0}^{n-1}$, $c_i = c_{-i}^*$. Since such matrices are completely specified by n entries rather than n^2 , one would

of course expect a reduction in computational effort for handling problems involving such matrices. However, exploiting the Toeplitz structure is apparently more difficult than it may at first seem. To see this, consider the simple case of a real symmetric 3×3 Toeplitz matrix and apply the first step of the Gaussian elimination procedure to it, namely

$$\begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_0 & c_1 \\ c_2 & c_1 & c_0 \end{bmatrix} - \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} c_0^{-1} [c_0 \quad c_1 \quad c_2] = \begin{bmatrix} 0 & & \\ & \Delta & \\ & & \end{bmatrix}$$

where the so-called Schur complement matrix Δ is seen to be

$$\Delta = \frac{1}{c_0} \begin{bmatrix} c_0^2 - c_1^2 & c_1 c_0 - c_1 c_2 \\ c_1 c_0 - c_1 c_2 & c_0^2 - c_2^2 \end{bmatrix}.$$

However, Δ is no longer Toeplitz, so the special structure is lost in the very first step of the procedure. The fact is that what is preserved is not the Toeplitz structure, but a deeper notion called “displacement structure”.

There are several forms of displacement structure, the earliest of which is the following [A1]. Consider an $n \times n$ Hermitian matrix R and the $n \times n$ lower-triangular shift matrix Z with ones on the first subdiagonal and zeros elsewhere (*i.e.*, a lower triangular Jordan block with eigenvalue 0). The displacement of R with respect to Z is denoted by $\nabla_Z R$ and defined as the difference

$$\nabla_Z R \triangleq R - ZRZ^*.$$

The matrix R is said to have displacement structure (or low displacement rank) with respect to Z if the rank of $\nabla_Z R$ is considerably lower than (and independent of) n . For example, a Hermitian Toeplitz matrix has displacement rank 2 with respect to Z .

More generally, let $r \ll n$ denote the rank of $\nabla_Z R$. Then one can write $\nabla_Z R$ as $\nabla_Z R = GJG^*$, where G is an $n \times r$ matrix and J is a “signature” matrix of the form $J = (I_p \oplus -I_q)$ with $p + q = r$. This representation is highly nonunique since G can be replaced by $G\Theta$ for any J -unitary matrix Θ , *i.e.*, for any Θ such that $\Theta J \Theta^* = J$; this flexibility is actually very useful. The matrix G is said to be a generator matrix for R since, along with $\{Z, J\}$, it completely identifies R . If one labels the columns of G as

$$G = [x_0 \quad \dots \quad x_{p-1} \quad y_0 \quad \dots \quad y_{q-1}],$$

and lets $L(x)$ denote a lower triangular Toeplitz matrix whose first column is x , then it can be seen that the unique R that solves the displacement equation $\nabla_Z R = GJG^*$ is

given by

$$\begin{aligned} R &= \sum_{i=0}^{n-1} Z^i G J G^* Z^{*i} , \\ &= \sum_{i=0}^{p-1} L(x_i) L^*(x_i) - \sum_{i=0}^{q-1} L(y_i) L^*(y_i) . \end{aligned}$$

Such displacement representations of R as a combination of products of lower- and upper-triangular Toeplitz matrices allow, for example, bilinear forms such as $x^* R y$ to be rapidly evaluated via convolutions (and hence FFTs).

As mentioned above, a general Toeplitz matrix has displacement rank 2, with in fact $p = 1$ and $q = 1$, as interested readers may wish to check. But there are interesting non-Toeplitz matrices with $p = 1 = q$, *e.g.*, the inverse of a Toeplitz matrix. In fact, this is a special case of the following fundamental result: if R is invertible and satisfies $R - Z R Z^* = G J G^*$, then there exists \tilde{H} such that $R^{-1} - Z^* R^{-1} Z = \tilde{H}^* J \tilde{H}$. The fact that Z and Z^* are interchanged in the latter formula suggests that we define a so-called “natural” inverse, $R^{-\#} = \tilde{I} R^{-1} \tilde{I}$, where \tilde{I} is the “reverse” identity matrix (with ones on the antidiagonal). For then one sees (with $H = \tilde{I} \tilde{H}^*$) that

$$R^{-\#} - Z R^{-\#} Z^* = H J H^* .$$

Therefore R and $R^{-\#}$ have the same displacement rank and (when R is Hermitian) the same displacement inertia (since J is the same). [A real symmetric Toeplitz matrix, T , has $T^{-1} = T^{-\#}$, so that T^{-1} has a representation of the form $T^{-1} = L(x) L^*(x) - L(y) L^*(y)$, which after suitably identifying $\{x, y\}$ is a special so-called Gohberg-Semencul formula.] The proof of the above fundamental result is very simple (see, *e.g.*, [A3]) and in fact holds with Z replaced by any matrix F .

An interesting example is obtained by choosing $F = \text{diag}\{f_0, \dots, f_{n-1}\}$, $|f_i| < 1$, when the solution R of the displacement equation $R - F R F^* = G J G^*$ becomes a “Pick” matrix of the form

$$P = \left[\frac{u_i u_j^* - v_i v_j^*}{1 - f_i f_j^*} \right]_{i,j=0}^{n-1} ,$$

where $\{u_i, v_i\}$ are $1 \times p$ and $1 \times q$ vectors. Pick matrices arise in solving analytic interpolation problems and the displacement theory gives new and efficient computational algorithms for solving a variety of such problems (see, *e.g.*, [A2]).

One can handle non-Hermitian structured matrices by using displacement operators of the form $\nabla_{F,A} R = R - F R A^*$. When F is diagonal with distinct entries and $A = Z$, R becomes a Vandermonde matrix. A closely related displacement operator, first introduced in [A4], has the form $F R - R A^*$. Choosing $F = \text{diag}\{f_i\}$, $A = \text{diag}\{a_i\}$ leads to Cauchy-like solutions of the form

$$C_l = \left[\frac{u_i v_j^*}{f_i - a_j^*} \right] , \quad u_i, v_i \in \mathcal{C}^{1 \times r} .$$

The name comes from the fact that when $r = 1$ and $u_0 = 1 = v_0$, C_l is a Cauchy matrix. When $r = 2$, and $u_i = [\beta_i \ 1]$ and $v_i = [1 \ -k_i]$, one gets the so-called Loewner matrix.

It is often convenient to use generating function language, and to define

$$R(z, w) \triangleq \sum_{i,j=0}^{\infty} R_{ij} z^i w^{*j} ,$$

[Finite structured matrices can be extended to be semi-infinite in many natural ways, *e.g.*, by extending the generator matrix by adding additional rows.] In this language, one can introduce general displacement equations of the form

$$d(z, w) R(z, w) = G(z) J G^*(w) ,$$

with

$$d(z, w) = \sum_{i,j=0}^{\infty} d_{ij} z^i w^{*j} .$$

Choosing $d(z, w) = 1 - zw^*$ corresponds to matrix displacement equations of the form (in an obvious notation), $R - Z R Z^* = G J G^*$, while $d(z, w) = (z - w^*)$ corresponds to $Z R - R Z^* = G J G^*$. There are many other useful choices, but to enable recursive matrix factorizations it is necessary to restrict $d(z, w)$ to the form $d(z, w) = \alpha(z) \alpha^*(w) - \beta(z) \beta^*(w)$. Note in particular that when $R = I$, one gets $R(z, w) = 1/(1 - zw^*)$, the Szegő kernel. Other choices lead to the various so-called deBranges and Bergman kernels. Interested readers can find more on this in [A5] and the references therein.

A central feature in many of the applications mentioned earlier is the ability to efficiently obtain the so-called triangular LDU factorization of a matrix and of its inverse. Among purely matrix computations, one may mention that this enables fast determination of the so-called QR decomposition and the factorization of composite matrices such as $T_1 T_2^{-1} T_3$, T_i being Toeplitz. The LDU factorization of structured matrices is facilitated by the fact that Schur complements inherit displacement structure. For example, writing

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} , \quad F = \begin{bmatrix} F_1 & 0 \\ F_2 & F_3 \end{bmatrix} ,$$

it turns out that

$$\text{rank}[S - F_3 S F_3^*] \leq \text{rank}[R - F R F^*] ,$$

where $S = R_{22} - R_{21} R_{11}^{-1} R_{12}$. By properly defining the $\{R_{ij}\}$ and $\{F_i\}$, this allows one to find the displacement structure of various composite matrices. For example, choosing $R_{11} = -T$, $R_{12} = I = R_{21}$, $R_{22} = 0$, and $F = Z \oplus Z$, gives the previously mentioned result on inverses of structured matrices.

Computations on structured matrices are made efficient by working not with the n^2 entries of R , but with the nr entries of the generator matrix G . The basic triangular factorization algorithm is Gaussian elimination, which

as noted in the first calculation, amounts to finding a sequence of Schur complements. Incorporating structure into the Gaussian elimination procedure was, in retrospect, first done (in the special case $r = 2$) by Schur himself in a remarkable 1917 paper dealing with the apparently very different problem of checking when a power series is bounded in the unit disc.

The algorithm below is one generalization of Schur's original recursion. It provides an efficient $O(rn^2)$ procedure for the computation of the triangular factors of a Hermitian positive-definite matrix R satisfying

$$R - ZRZ^* = GJG^*, \quad G \in \mathbb{C}^{n \times r},$$

with $J = (I_p \oplus -I_q)$. So let $R = LD^{-1}L^*$ denote the triangular decomposition of R , where $D = \text{diag}\{d_0, d_1, \dots, d_{n-1}\}$, and the lower triangular factor L is normalized in such a way that the $\{d_i\}$ appear on its main diagonal. The nonzero part of the consecutive columns of L will be further denoted by l_i . Then it holds that the successive Schur complements of R with respect to its leading $i \times i$ submatrices, denoted by R_i , are also structured and satisfy

$$R_i - Z_i R_i Z_i^* = G_i J G_i^*,$$

with $(n-i) \times (n-i)$ lower triangular shift matrices Z_i , and where the generator matrices G_i are obtained by the following recursive construction: start with $G_0 = G$ and repeat for $i \geq 0$:

$$\begin{bmatrix} 0 \\ G_{i+1} \end{bmatrix} = \left\{ G_i + Z_i G_i \frac{J g_i^* g_i}{g_i J g_i^*} \right\} \Theta_i,$$

where Θ_i is an arbitrary J -unitary matrix, and g_i denotes the top row of G_i . Then

$$l_i = G_i J g_i^*, \quad d_i = g_i J g_i^*.$$

The degree of freedom in choosing Θ_i is often very useful. One particular choice leads to the so-called proper form of the generator recursion. Let Θ_i reduce g_i to the form

$$g_i \Theta_i = [\delta_i \quad 0 \quad \dots \quad 0],$$

with a nonzero scalar entry in the leading position. Then

$$\begin{bmatrix} 0 \\ G_{i+1} \end{bmatrix} = Z_i G_i \Theta_i \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + G_i \Theta_i \begin{bmatrix} 0 & 0 \\ 0 & I_{p+q-1} \end{bmatrix},$$

with

$$l_i = \delta_i^* G_i \Theta_i \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad d_i = |\delta_i|^2.$$

In words, this shows that G_{i+1} can be obtained from G_i as follows:

1. Reduce g_i to proper form.
2. Multiply G_i by Θ_i and keep the last columns of $G_i \Theta_i$ unaltered;
3. Shift down the first column of $G_i \Theta_i$ by one position;

Extensions of the algorithm to more general structured matrices are possible (see, e.g., [A2]-[A3]). In addition, the

algorithm can be extended to provide simultaneous factorizations of both a matrix R and its inverse, R^{-1} – see, e.g., [A3].

An issue that arises in the study of such fast factorization algorithms is their numerical stability in finite-precision implementations. It was mentioned earlier that the generalized Schur algorithm amounts to combining Gaussian elimination with structure, and it is well-known that Gaussian elimination in its purest form is numerically unstable (meaning that the error in the factorization $\hat{L}\hat{D}^{-1}\hat{L}^*$ can be quite large, where \hat{L} and \hat{D} denote the computed L and D , respectively). The instability can often be controlled by resorting to pivoting techniques, *viz.*, by permuting the order of the rows, and perhaps columns, of the matrices before the Gaussian elimination steps. However, pivoting can destroy matrix structure and can thus lead to a loss in computational efficiency. It was observed in [A6] though that for diagonal displacement operators F , and more specifically for Cauchy-like matrices, partial pivoting does not destroy matrix structure. This fact was exploited in [A7] in the context of the generalized Schur algorithm and applied to other structured matrices. This was achieved by showing how to transform different kinds of matrix structure into Cauchy-like structure.

Partial pivoting by itself is not sufficient to guarantee numerical stability even for slow algorithms. Moreover, the above transformation-and-pivoting technique is only efficient for fixed-order problems since the transformations have to be repeated afresh whenever the order changes. Another approach to the numerical stability of the generalized Schur algorithm is to examine the steps of the algorithm directly and to stabilize them without resorting to transformations among matrix structures. This was done in [A8] where it was shown, in particular, how to implement the hyperbolic rotations in a reliable manner. For all practical purposes, the main conclusion of [A8] is that the generalized Schur algorithm, with certain modifications, is backward stable for a large class of structured matrices.

References

- [A1] KAILATH, T., KUNG, S. Y., and MORF, M., Displacement ranks of a matrix, *Bulletin of the American Mathematical Society*, vol. 1, no. 5, pp. 769–773, Sep. 1979.
- [A2] KAILATH, T. and SAYED, A. H., Displacement structure: Theory and applications, *SIAM Review*, 37, pp. 297–386, 1995.
- [A3] KAILATH, T., Displacement structure and array algorithms, in *Fast Reliable Algorithms for Matrices with Structure*, T. Kailath and A. H. Sayed, editors, SIAM, PA, 1999.
- [A4] HEINIG, G. and ROST, K., *Algebraic Methods for Toeplitz-like Matrices and Operators*, Akademie-Verlag, Berlin, 1984.
- [A5] LEV-ARI, H. and KAILATH, T., Triangular factorization of structured Hermitian matrices, *Operator Theory: Advances and Applications*, Birkhäuser, Boston, vol. 18, pp. 301–324, 1986.
- [A6] HEINIG, G., Inversion of generalized Cauchy matrices and other classes of structured matrices, in *Linear Algebra for Signal Processing*, IMA volumes in Mathematics and its Applications, A. Bojanczyk and G. Cybenko, editors, vol. 69, pp. 95–114, 1994.
- [A7] GOHBERG, I., KAILATH, T. and OLSHEVSKY, V., Fast Gaussian elimination with partial pivoting for matrices with displacement structure, *Math. of Computation*, vol. 64, pp. 1557–1576, 1995.
- [A8] CHANDRASEKARAN, S. and SAYED, A. H., Stabilizing the gen-

eralized Schur algorithm, *SIAM J. Matrix Anal. Appl.* vol. 17,
pp. 950–983, 1996.