A unifying approach to the construction of circulant preconditioners

Ivan Oseledets, Eugene Tyrtyshnikov [1]

Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow 119991

Abstract

The main result is the "black dot algorithm" and its fast version for the construction of a new circulant preconditioner for Toeplitz matrices. This new preconditioner C is sought directly as a solution to one of possible settings of the approximation problem $A \approx C + R$, where A is a given matrix and R should be a "low-rank" matrix. This very problem is a key to the analysis of superlinear convergence properties of already established circulant and other matrix-algebra preconditioners. In this regard, our new preconditioner is likely to be the best of all possible circulant preconditioners. Moreover, in contrast to several "function-based" circulant preconditioners used for "bad" symbols, it is constructed entirely from the entries of a given matrix and performs equally as the best of the known or better than those for the same symbols.

AMS classification: 15A12; 65F10; 65F15

Key words: Matrix approximations; Preconditioners; Superlinear convergence; Circulants; Toeplitz matrices; Low-rank matrices; Skeleton decomposition; Spectral clusters; Spectral distributions.

1 Introduction

The idea of using circulants as preconditioners for Toeplitz matrices was first proposed by Gilbert Strang in 1986 [7]. His idea was to construct a circulant

Email addresses: ivan@bach.inm.ras.ru (Ivan Oseledets), tee@inm.ras.ru (Eugene Tyrtyshnikov).

[1] Supported by the Russian Fund of Basic Research (grant 05-01-00721) and a Priority Research Grant of the Department of Mathematical Sciences of the Russian Academy of Sciences.

taking half of the entries from the first row and column of a Toeplitz matrix. Another popular approach is the optimal preconditioner of T. Chan, which is the Frobenius-norm nearest circulant matrix to the given (Toeplitz) matrix [3]. These preconditioners are easy to construct, but in some cases they fail (the number of iterations may grow considerably as the matrix size n increases). For "bad" cases, several other methods and algorithms were proposed (cf. a survey [9]). However, the most efficient approaches use the symbol (generating function) and, for this reason, can be called "function-based" rather than "matrix-based" (cf. [6]). Moreover, a method which is suitable for symmetric positive definite matrices may not work well for indefinite or nonsymmetric matrices.

In this paper we propose a matrix approach that delivers a new circulant preconditioner which is likely to be the best of the known circulant preconditioners. In contrast to several "function-based" circulant preconditioners used for "bad" symbols, it is constructed entirely from the entries of a given matrix and performs equally as the best of the known or better than those for the same symbols. The main result is the "black dot algorithm" and its fast version for the construction of certain circulant approximations to Toeplitz matrices. In short, if a good circulant preconditioner exists then it can be easily found by our algorithm.

Let us start from the beginning. When a linear system

$$Ax = b$$

is solved by some iterative method (like CG or GMRES) and the convergence is slow (which occurs frequently), then a well-known remedy is to get to a preconditioned system

$$AP^{-1}x = b,$$

where P is called a preconditioner. A rigorous analysis of the preconditioner quality typically starts with embedding a particular system into a sequence of systems (coefficient matrices, right-hand-side vectors, and preconditioners) parameterized by the matrix size n. Then, in order to have a "good preconditioner", we usually take care of the following properties:

(a) AP^{-1} is *boundedly conditioned* (there is a bound on the condition number which is uniform in n);
(b) AP^{-1} has an *eigenvalue cluster* at unity.

At least for Hermitian positive definite matrices and under some additional assumptions in the general case, property (a) indicates the *linear convergence* while (b) underlies the so called *superlinear convergence* (cf. [11]). The existence of cluster is directly related to decompositions of the form [10]

$$A = P + R + E, \tag{1}$$

where rank$R = r \ll n$ and $||E|| \le \varepsilon$. The matrices in the right-hand side of (1) depend on n and ε.

We suggest to construct preconditioners P capitalizing directly on (1). When restricting the choice of P to a suitable matrix class, we consider (1) as a sort of approximation problem. Informally as yet, it reads as follows.

C+R approximation problem: *given a matrix A, approximate it by the sum of two matrices*

$$A \approx C + R,$$

where $C = P$ is a circulant and R is a "low rank" matrix.

For example, consider Toeplitz matrices $A = [a_{i-j}]$ of sizes n = 128, 256, 512 generated by the symbol $f = x^4$ considered on the interval $-\pi < x < \pi$ (it means that a_k are the Fourier coefficients for f). Let $P = C$ in (1) be either the Strang or T. Chan preconditioner. Then, setting the accuracy to $\varepsilon = 10^{-2}$, we find R by the truncation of the singular values of $A - C = R + E$ at the level of ε so that $||E||_2 \le \varepsilon$. In this case we obtain the following ranks for R:

n	Strang	T. Chan
128	8	20
256	8	24
512	8	24

Table 1.1 Dependence of rank$R$ upon n ($\varepsilon = 10^{-2}$).

For that fixed ε, rank$R$ does not depend on the matrix size n any pronouncedly. However, let us inquire into how it depends on ε for a fixed n.

ε	Strang	T. Chan
10^{-3}	10	244
10^{-4}	18	254
10^{-5}	50	256

Table 1.2 Dependence of rank$R$ upon ε ($n = 256$).

As we see, both preconditioners are not satisfactory with respect to producing a cluster: in this role the T. Chan preconditioner fails completely whereas the Strang one is not good enough because the ε-ranks seem to grow as $\varepsilon^{-\alpha}$, $\alpha \sim 1$. The matrix A is ill-conditioned; therefore, we have to approximate it with high accuracy and none of these preconditioners yields a proper cluster. In this case, however, it turns out that A can be *very accurately* approximated by the sum of a circulant and a matrix of *sufficiently low* rank. But, the corresponding circulant has nothing to do with neither the Strang, nor T. Chan preconditioner. Moreover, it can be proved that quite a general class of Toeplitz

matrices (including all examples in papers on superlinear preconditioners) admits approximations by the sum of a circulant and a low-rank matrix with the estimate [13]

$$r = \mathcal{O}(\log \varepsilon^{-1}(\log \varepsilon^{-1} + \log n)).$$

Hence, we are aware of the existence of a "good circulant" and may be interested to figure out how it can be computed.

We organize the paper in the following way.

In section 2 we expose several settings for the $C + R$ approximation problem and reformulate it as a completion problem for a low-rank matrix with some missing elements ("black dots" in our terminology).

In section 3 we expound a basic idea for the solution of the low-rank completion problem ("black dot algorithm") and prove that it works in a "noise-free" case.

In section 4 we give a practical version of the black dot algorithm which allows us to adaptively determine the rank depending on the desired approximation accuracy.

In section 5 we develop a fast algorithm for the case of Toeplitz matrices and report on the theorems about $C + R$ approximations which are proved in [13].

Finally, in section 6 we present some numerical experiments which confirm the theory.

2 The C+R and D+R approximation problems

Since every circulant matrix is diagonalized by the Discrete Fourier Transform (DFT)

$$C = \frac{1}{n} F^* D F,$$

where F is the DFT matrix and D is a diagonal matrix, the $C + R$ approximation problem can be recast as follows:

$$\widehat{A} = \frac{1}{n} F A F^* \approx D + R. \tag{2}$$

Thus, the general $C + R$ approximation problem easily reduces to the $D + R$ approximation problem, where D is a diagonal matrix.

Now, let us specify what "approximately" and "low rank" mean. When fixing a bound on the rank of the low rank part, we obtain the following optimization problem.

D+R problem I: *Given a matrix A and an integer $r > 0$, find a matrix $B = D + R$ where* $\mathrm{rank}R \leq r$ *and D is a diagonal matrix that minimizes $||A - B||_F$.*

We can exclude from this formulation either R or D. When excluding R, we obtain an optimization problem in the terms of singular values.

D+R problem II: *Given a matrix A and an integer $r > 0$, find a diagonal matrix D that minimizes*

$$\sigma_{r+1}(A - D).$$

It is worth noting that this is a non-smooth, non-convex optimization problem that seems to have many local minima. (We are not aware of any convenient way to solve it.)

When excluding D, we arrive at the following formulation.

D+R problem III: *Given a matrix $A = [A_{ij}]$ and an integer $r > 0$, find a matrix $R = [R_{ij}]$ of rank not greater than r so that it minimizes*

$$\sum_{i,j=1,\, i \neq j}^{n} (A_{ij} - R_{ij})^2.$$

The $D + R$ problem was first considered in [1] with an iterative method proposed therein to solve it. It was a variant of the alternating least squares approach called ADR (Alternating Diagonal Rank) with a two-step iteration of the following form.

Given some guesses for D and R, find new approximations to the solution $\widehat{D}$ and $\widehat{R}$ as follows:

(1) $\widehat{D} = \arg\min_{D} ||A - D - R||_F$;

(2) $\widehat{R} = \arg \min_{R,\, \mathrm{rank}R \leq r} ||A - \widehat{D} - R||_F$.

It is easy to see that at each step the residue $||A - D - R||_F$ decreases. Unfortunately, it seems to be the only advantage of ADR. It often requires a huge amount of iterations. It is sometimes stuck into a local minimum. And it requires $\mathcal{O}(n^3)$ operations at each iteration, which makes this method (in this form) unacceptable for practical purposes. There is a way to modify it (not so trivially) so that it converges to the global minimum, but the computational cost still remains extremely high. Nevertheless, if a good approximation is obtained by some other method (e.g. by the algorithm proposed in the next section), we can get and try some fast modification of ADR to refine a given approximation to the solution.

Let us look more closely at the D+R problem III. Recall that we are interested in the case where a matrix A is well approximated by the sum of a diagonal and a rank-r matrix. Let us begin with the assumption that A is *exactly* the sum of a diagonal and a rank-r matrix. How can we reveal D and R when given only their sum? The answer is in the next section.

3 Black dots, low rank and skeletons

The problem is formulated in the following way. Suppose that a matrix A can be *exactly* represented as the sum of a diagonal and a rank-r matrix. Given $A = D + R$, how to recover D and R from A?

It is obvious that in the matrix R we know all off-diagonal elements. Therefore, all what is left is to find the diagonal elements of R. Before describing the general algorithm, consider the following simple example of a 6×6 matrix of rank 2:

$$A = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 7 & 8 & 9 & 10 & 11 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{pmatrix}.$$

(It is really a rank-2 matrix because $a_{ij} = i + j$).

Suppose now that we do not know the diagonal elements of the matrix A:

$$A = \begin{pmatrix} \bullet & 3 & 4 & 5 & 6 & 7 \\ 3 & \bullet & 5 & 6 & 7 & 8 \\ 4 & 5 & \bullet & 7 & 8 & 9 \\ 5 & 6 & 7 & \bullet & 9 & 10 \\ 6 & 7 & 8 & 9 & \bullet & 11 \\ 7 & 8 & 9 & 10 & 11 & \bullet \end{pmatrix}.$$

The diagonal elements are marked by black dots. And the question is how to complete the off-diagonal part by filling in the black dots so that the resulting matrix has rank 2? A simple idea can be adopted. Take up the submatrix

housed by the columns 4,5,6 and rows 2,3,4:

$$\widehat{A} = \begin{pmatrix} 6 & 7 & 8 \\ 7 & 8 & 9 \\ \bullet & 9 & 10 \end{pmatrix}.$$

We want to get a rank-2 matrix, so these three columns have to be linearly dependent; hence, the first column has to be a linear combination of the second and the third column. The coefficients of this linear combination are easily determined by solving the following system:

$$\begin{pmatrix} 7 & 8 \\ 8 & 9 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 7 \end{pmatrix}.$$

As is readily seen, the selection of the rows and columns can be done in several ways. In this example, all these ways will lead to the same result. In practice, however, the choice of rows and columns used for the reconstruction is an important issue and may (and often does) go wrong and cause instability.

Let us describe the above procedure in a general setting and prove that it really reconstructs the black dots.

Consider an arbitrary rank-r matrix B, take r linear independent rows and r linear independent columns from B and form matrices $L \in \mathbb{R}^{n\times r}$ (from the columns) and $U \in \mathbb{R}^{r\times n}$ (from the rows). Let $\hat{B}$ denote the $r \times r$ submatrix on the intersection of these selected rows and columns. Then, the submatrix $\hat{B}$ is nonsingular and the matrix B can be decomposed as

$$B = L\hat{B}^{-1}U,$$

which is sometimes called the *skeleton decomposition*.

The main point here is that the rank-r matrix is uniquely defined by its r linear independent columns and r linear independent rows. Let us construct the skeleton decomposition for the matrix with black dots on the diagonal. For our example, the rows 3,4 and columns 1,2 provide us with the nonsingular intersection submatrix, and hence, we can write

$$A = \begin{pmatrix} \bullet & 3 \\ 3 & \bullet \\ 4 & 5 \\ 5 & 6 \\ 6 & 7 \\ 7 & 8 \end{pmatrix} \begin{pmatrix} 4 & 5 \\ 5 & 6 \end{pmatrix}^{-1} \begin{pmatrix} 4 & 5 & \bullet & 7 & 8 & 9 \\ 5 & 6 & 7 & \bullet & 9 & 10 \end{pmatrix}. \tag{3}$$

In order to specify how the black dots are handled, introduce the following "black dot arithmetic":

$$\bullet\bullet = \bullet,$$

$$\bullet x = x\bullet = \bullet,$$

$$\bullet + x = x + \bullet = \bullet,$$

where x is an ordinary number. Thus, the multiplication of matrices in (3) shows that

$$A = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 4 & 5 & \bullet & \bullet & 8 & 9 \\ 5 & 6 & \bullet & \bullet & 9 & 10 \\ 6 & 7 & \bullet & \bullet & \underline{10} & 11 \\ 7 & 8 & \bullet & \bullet & 11 & \underline{12} \end{pmatrix}.$$

That means that we have found the (underlined) diagonal elements (5,5) and (6,6). Since all off-diagonal elements in A are given, now we know two full rows 5,6 and two full columns 5,6 of A and can use the skeleton decomposition again to obtain the full matrix.

In the general case we do literally the same thing.

Black dot algorithm. *Given a matrix A and granted that it admits a splitting into $A = D + R$ with a diagonal D and a rank-r matrix R, find at least $n - 2r$ full columns and rows of the unknown R proceeding as follows:*

(1) *Pick up in A a nonsingular $r \times r$ submatrix $\hat{A}$ whose elements do not lie on the main diagonal. Suppose that the rows and columns of this submatrix have indices $i_1, ..., i_r$ and $j_1, ..., j_r$, respectively, and let matrices L and U of sizes $n \times r$ and $r \times n$ be composed of these columns and rows.*

(2) *Form the matrix* $Q = L\hat{A}^{-1}U$ *(still with some black dots) and observe that the elements*

$$Q_{ij} = R_{ij}, \qquad i \neq j_1, ..., j_r, \quad j \neq i_1, ..., i_r, \tag{4}$$

are no longer the black dots. Consequently, at this moment we come to know at least $n - 2r$ *diagonal elements of the matrix* R*.*

The algorithm is based on the following simple

Theorem 3.1 *The elements of the above defined matrix* Q *satisfy (4).*

Proof. Using the definition of L and U, we obtain

$$Q_{ij} = \sum_{k=1}^{r} \sum_{l=1}^{r} R_{ij_k} (\hat{A}^{-1})_{kl} R_{i_l,j}.$$

If $i \neq j_1, ..., j_r$ and $j \neq i_1, ..., i_r$, then none of the elements R_{i,j_k}, R_{j,i_l} is located on the main diagonal. Thus, all these elements are known and the corresponding elements of Q must coincide with those of R. $\square$

In our applications $r \ll n$, so the two steps of the black dot algorithm allow us to find the bulk of diagonal elements. To reveal the remaining entries of R we need one more step as follows:

(3) *If* n *is large enough (let* $n - 2r \geq r$*, or, equivalently,* $n \geq 3r$*), then at least* r *full rows and* r *full columns of the matrix* R *are acquired. Assume that these* r *columns and rows are linear independent. Then use them to build up the skeleton decomposition and decipher the remaining black dots of* R*.*

Remark that this third step is based on the assumption that the first two steps have produced r linear independent columns and rows with already known elements. It is sufficient to assume that A possesses two nonsingular $r \times r$ submatrices which occupy neither common column nor common row of A and both contain only off-diagonal elements of A.

4 Adaptive version of the black dot algorithm

Several problems still remain which were not discussed above. First of all, A may be not exactly the sum of a diagonal and a low rank matrix. Instead,

$$A = D + R + E,$$

where $||E|| \leq \varepsilon$ can be viewed as kind of "noise". Moreover, the bound r on the rank of R (depending on ε) may be not known beforehand. Thus, we are faced with a rank-revealing problem wherein r is to be found, given some desired accuracy ε.

In the "noise-present" case, the choice of the columns and rows (equivalently, the intersection submatrix) on which the skeleton decomposition is based is crucial. Which submatrix is the best? If there were no black dots, a good choice would reside in the *maximal volume principle* [5]: if the intersection submatrix $\widehat{A}$ has maximal volume (determinant in modulus) among all $r \times r$ submatrices then the element-wise error estimate for the skeleten decomposition reads

$$|(A - L\hat{A}^{-1}U)_{ij}| \;\leq\; (r+1)\,\sigma_{r+1}(A), \tag{5}$$

where $\sigma_{r+1}(A)$ is the singular value of A (assumed in the non-inreasing order) on position $r+1$. We conjecture that in the case of black dots the same good choice should be a submatrix of maximal volume among all $r \times r$ submatrices with fully defined elements (having no black dots).

Since finding the maximal volume submatrix is not an easy task, we can do with some submatrix of a sufficiently large volume. Such a submatrix (and its size, above all) can be obtained by a variant of the *incomplete cross approximation algorithm* [4,12] (cf. [2] for boundary element applications). It proceeds as follows(here R is a nonzero matrix to be approximated by a low-rank matrix):

(1) Initialization: $k = 0,\;\; R^k = R$.

(2) Find the pivot position: $(i_0, j_0) = \arg\max_{i,j} |R^k_{ij}|$.

(3) Calculate the cross-based skeleton:

$$C_k = \frac{u_k v_k^\top}{R^k_{i_0 j_0}},$$

where u_k is the i_0-th row of matrix R_k and v_k is the j_0-th column of R_k.

(4) Calculate the new residue: $R_{k+1} = R_k - C_k$.

(5) If the residue norm $||R_{k+1}||$ is small enough, then stop and return $\sum_{i=0}^{k} C_i$ as a rank-r skeleton approximation to the initial matrix R. Otherwise, increase k by 1 and proceed with step 2.

At each step we substract from the matrix a single rank-one matrix, called *skeleton* and determined from the pivot column and row comprising a cross. The pivots are chosen to eliminate "large" elements in the residue matrix.

When the algorithm is finished, it gives an instance of the skeleton decomposition with a hopefully good submatrix (the pivot strategy does not pursue exact maximization of the volume, but is capable of making it reasonably large).

The overall cost of the above incomplete cross approximation variant is $\mathcal{O}(n^2 r)$ operations due to the complete pivoting step. However, the incomplete cross approximation approach was originally motivated by the hope that it can approximate a near-to-low-rank matrix using just a small amount of its elements (the cross) [4]. If the matrix is exactly of rank r, then the Gaussian elimination with pivoting gives zero pivot exactly after r steps. As a matter of fact, the same approach is adapted to the "noise-present" case. And it may be implemented with various pivoting strategies (e.g. row or column pivoting) and with especial gain from those that leave most elements of the matrix out of play. With a partial pivoting, we may get a larger coefficient at $\sigma_{r+1}(A)$ in the skeleton decomposition estimate (5); it depends in effect on how close the resulting intersection matrix volume gets to the maximal volume [5]. For some strategies and under certain assumptions, it can be 2^r instead of $r+1$. Anyway, even by the price of accuracy deterioration, a significant reduction of computational complexity makes partial pivoting the only practical choice for the cross approximation algorithm.

The incomplete cross approximation algorithm can be easily tailored to the case of matrix with some unknown entries (black dots). We only have to trace how the black dots spread at each step. The rows and columns containing the black dots will comprize our "black lists" (revised at each step).

Adaptive black dot algorithm.

(1) *Initialization:*

$$k = 0, \quad R_{ii} = 0, \quad R_{ij} = A_{ij}, \quad i \neq j, \quad R^k = R,$$

$$\mathcal{L}_r = \emptyset, \quad \mathcal{L}_c = \emptyset \quad (\textit{"black lists"}).$$

(2) *Find the pivot position:*

$$(i_0, j_0) = \arg \max_{i \neq j,\, i \notin \mathcal{L}_c,\, j \notin \mathcal{L}_r} |R^k_{ij}|.$$

(3) *Calculate the cross-based skeleton:*

$$C_k = \frac{u_k v_k^\top}{R^k_{i_0 j_0}},$$

where u_k is the i_0-th row and v_k is the j_0-th column of R_k.

(4) *Calculate the new residue:* $R_{k+1} = R_k - C_k$.

(5) *Add the element* i_0 *to* $\mathcal{L}_c$ *and the element* j_0 *to* $\mathcal{L}_r$.

(6) *Calculate the error:*

$$\delta^k = (\sum_{i,j \in S} (R^k_{ij})^2)^{1/2}, \quad S = \{i, j : 1 \le i, j \le n,\ i \ne j,\ i \notin \mathcal{L}_r,\ j \notin \mathcal{L}_c\}.$$

(7) *If* δ_k *is small enough, then quit and return*

$$d_i = (\sum_{m=0}^{k} C_m)_{ii}, \quad i \notin \mathcal{L}_r, \quad i \notin \mathcal{L}_c,$$

as approximations to the corresponding diagonal elements R_{ii}. *Otherwise, increase* k *by* 1 *and proceed with step 2.*

This algorithm creates an instance of the skeleton decomposition with some adaptively chosen submatrix $\hat{R}$, whose size was not given beforehand and which is expected to be "good enough" for the skeleton approximation purposes. The error is measured on the elements in the known part of the matrix.

If n is large enough, then the black dot algorithm returns the approximations to all but $2r$ diagonal elements. Then we are to run it the second time with a constraint that pivots are selected only from fully known rows and columns. In the end we obtain the skeleton approximation for R:

$$R \approx \sum_{k=1}^{r} x_k y_k^\top = XY^\top.$$

For the diagonal matrix part of the $D + R$ approximation, we apparently have

$$D \approx \mathrm{diag}(\mathrm{A} - \mathrm{XY}^\top).$$

As it is put above, the adaptive black dot algorithm caters for the $D + R$ approximation problem. However, it can be easily adapted to many problems with other prescribed patterns for the black dots, rather than the main diagonal entries.

The above-presented algorithms require $\mathcal{O}(n^2(\log n + r))$ operations to construct a $C + R$ approximation to an unstructured matrix A. This comes from the use of FFT to compute the elements of FAF^* and is also due to the complete pivoting strategy. In the case of Toeplitz matrices this is not acceptable. In the next section we will show how the above techniques can work in the Toeplitz case with the $\mathcal{O}(n(\log n + r^2))$ complexity.

5 Toeplitz case

5.1 Fast evaluation of the elements of the Fourier image of a Toeplitz matrix

Now assume that the input matrix T is Toeplitz and A is its Fourier image:

$$T = [t_{i-j}], \qquad A = \frac{1}{n} FTF^*.$$

The off-diagonal elements of $A = [A_{kl}]$ admit a simple and gainful parametrization as follows.

Lemma 5.1

$$A_{kl} = \frac{v_k - v_l}{n(w^{k-l} - 1)}, \quad 0 \le k, l \le n-1, \quad k \ne l, \tag{6}$$

$$v_k = w^{-k}\widetilde{v}_k, \qquad w = e^{\frac{2\pi i}{n}},$$

$$\widetilde{v} = F\widetilde{t}, \qquad \widetilde{t}_k = t_{k-n} - t_k, \quad 0 \le k \le n-2, \qquad \widetilde{t}_{n-1} = 0.$$

Proof. By the definition of the Fourier image A,

$$nA_{kl} = \sum_{\alpha=0}^{n-1}\sum_{\beta=0}^{n-1} w^{-\alpha k} t_{\alpha-\beta} w^{\beta l} = \sum_{\beta=0}^{n-1} w^{\beta(l-k)} \sum_{\alpha=-\beta}^{n-\beta-1} w^{-\alpha k} t_\alpha.$$

Upon a change of the summation order, we obtain

$$\begin{aligned}
nA_{kl} &= \sum_{\alpha=-n+1}^{-1} w^{-\alpha k} t_\alpha \sum_{\beta=-\alpha}^{n-1} w^{\beta(l-k)} + \sum_{\alpha=1}^{n-1} w^{-\alpha k} t_\alpha \sum_{\beta=0}^{n-\alpha-1} w^{\beta(l-k)} \\
&= \sum_{\alpha=1}^{n-1} w^{\alpha l} t_{-\alpha} \frac{w^{(n-\alpha)(l-k)} - 1}{w^{l-k} - 1} + \sum_{\alpha=1}^{n-1} w^{-\alpha k} t_\alpha \frac{w^{(n-\alpha)(l-k)} - 1}{w^{l-k} - 1} \\
&= \frac{1}{w^{k-l} - 1} \Big(\sum_{\alpha=1}^{n-1} (w^{\alpha k} t_{-\alpha} - w^{\alpha l} t_{-\alpha}) + \sum_{\alpha=1}^{n-1} (w^{-\alpha l} t_\alpha - w^{-\alpha k} t_\alpha) \Big) \\
&= \frac{v_k - v_l}{w^{k-l} - 1},
\end{aligned}$$

where

$$v_k = \sum_{\alpha=1}^{n-1} (w^{\alpha k} t_{-\alpha} - w^{-\alpha k} t_\alpha) = \sum_{\alpha=1}^{n-1} w^{-\alpha k} (t_{\alpha-n} - t_\alpha). \quad \square$$

The total cost of computing v is the cost of one FFT and of the multiplication by a diagonal matrix. The main diagonal of A is also computed in one FFT. As soon as the preprocessing step is done, each element of A can be computed via (6) very fast.

Instead of the complete pivoting step, we have to use some partial pivoting. We propose a *rook scheme* as follows:

(1) *At each step, calculate the superdiagonal of the residue matrix R_k:*

$$S = [(R_k)_{12}, ..., (R_k)_{n-1,n}].$$

(2) *Find the maximal in modulus element in S and its position $(i_0, i_0 + 1)$.*
(3) *Find the maximal in modulus element in the i_0-th row of the matrix R_k and use it for the calculation of the next cross.*

The overall complexity of the black dot algorithm with this rook scheme for Toeplitz matrices is now reduced to

$$\mathcal{O}(n(\log n + r^2)).$$

The factor r^2 comes from the fact that to calculate a certain column or row of the residue matrix R_k we have to calculate corresponding elements in previous $k - 1$ crosses, therefore the complexity for rank r is proportional to

$$n(0 + 1 + 2 + ... + (r - 1)) = \mathcal{O}(nr^2).$$

Thus, in the Toeplitz case the C+R approximation can be calculated fast, provided that $r \ll n$. Upper estimates on r, which we call *circulant ranks*, are presented in the next subsection.

5.2 Existence of the C+R approximation for some Toeplitz matrices

As we claimed in the introduction, some wide and practically important classes of symbols lead to such Toeplitz matrices that can be very accurately approximated by the sum of a circulant and a low rank matrix. Below we summarize the results on this issue proved in [13].

Theorem 5.1 *Let T be a Toeplitz matrix with the symbol*

$$f = P(z) + \frac{Q(z)}{L(z)}, \quad z = e^{ix},$$

where P, Q, L are polynomials, L has no roots on the unit circle, the degree of Q is not greater than the degree of L, and L and Q have no common roots.

Then there exists a circulant matrix C and a matrix R such that

$$T = C + R,$$

$$\operatorname{rank}(R) \;\leq\; \deg(P) + \deg(L) + 1.$$

Theorem 5.2 *Let a Toeplitz matrix T be associated with a piecewise-analytic symbol of the form*

$$f = g + \sum_{\alpha=0}^{l} \sum_{k=0}^{m} A_{k\alpha} \, (z - \zeta_k)^{\alpha} \, \log(z - \zeta_k), \quad z = e^{ix}, \quad |\zeta_k| = 1,$$

where g is analytic in a disk containing $|z| = 1$. Then for any $\varepsilon > 0$ there exist a circulant C and a matrix R such that

$$|(T - C - R)_{ij}| \;\leq\; |T_{ij}|\varepsilon,$$

$$\operatorname{rank}(R) \;\leq\; \log \varepsilon^{-1}[c_0 + c_1 \log \varepsilon^{-1} + c_2 \log n] + c_3, \tag{7}$$

and c_0, c_1, c_2, c_3 are independent of n and ε.

Theorem 5.1 states that Toeplitz matrices generated by an arbitrary rational trigonometric symbol are exactly the sums of a circulant and a matrix whose rank is bounded uniformly in the matrix size.

Theorem 5.2 addresses the case when a symbol is an analytic function plus a function with logarithmic singularities. The corresponding Toeplitz matrices can be approximated by $C + R$ matrices with pretty high accuracy. At the first glance, this may seem to be a rather special class of matrices. However, this very prototype for symbols *covers all examples considered in papers on superlinear preconditioners.* Indeed, functions of the form $(z - \zeta_k)^{\alpha} \log(z - \zeta_k)$ possess a jump in the α-th derivative. For example, for $f = x^4$ defined on the interval $-\pi < x < \pi$ and then considered as a 2π-periodic function for all x, we have jumps in the first and third derivatives due to their non-periodicity. Substracting from f functions of the form

$$A(z - \zeta) \log(z - \zeta) + B(z - \zeta)^3 \log(z - \zeta),$$

where ζ is a jump point and A and B are proportional to the magnitude of the jumps, we obtain the analytic function which can be approximated by trigonometric polynomials (leading to band Toeplitz matrices) with exponentially decaying error.

To summarize, all functions with a finite number of jumps of finite order enjoy the $C + R$ approximation with the rank of R estimated by (7).

6 Numerical experiments

The circulants taken from the $C + R$ approximation of Toeplitz matrices are natural to use as preconditioners in PCG (preferred whenever possible) or GMRES (in all other cases). These circulants are obtained by the rook scheme of the black dot algorithm. We have tried them for Toeplitz matrices generated by several typical symbols (defined on the interval $= -\pi < x < \pi$ and then extended by 2π-periodicity to all real x) as follows:

(A) Positive definite Hermitian Toeplitz matrices:

(1) $f_1 = |x|$,
(2) $f_2 = x^2$,
(3) $f_3 = |x|^3$,
(4) $f_4 = x^4$,
(5) $f_5 = x^2(x-\pi)^2$,
(6) $f_6 = (x+\pi)^2$.

(B) Indefinite Hermitian Toeplitz matrices taken from [8]:

(7) $f_7 = x^2(x^2+1)\mathrm{sgn}(x)$,
(8) $f_8 = \mathrm{sgn}(x-\pi+2)\mathrm{sgn}(x+\pi-2)(\cos(x+2)+1)(\cos(x-2)+1)$,
(9) $f_9 = ((\frac{x}{\pi})^2 - 1)^2 - 0.9$.

(C) Non-Hermitian Toeplitz matrices ($z = e^{ix}$):

(10) $f_{10}(z) = \frac{z^4-1}{(z-\frac{3}{2})(z-\frac{1}{2})}$,
(11) $f_{11}(z) = \frac{(z+1)^2(z-1)^2}{(z-\frac{3}{2})(z-\frac{1}{2})}$.

Table 6.1 presents timings (here 1 = time for computing a single Toeplitz matrix-by-vector product) for the rook scheme. The matrices were approximated with relative accuracy $\varepsilon = 10^{-7}$. It is worthy to note that, in practice, different symbols may require different accuracies. If the matrix is not very ill-conditioned, we might opt for a larger ε. But to deal with ill-conditioned matrices we should take ε sufficiently small.

The Toeplitz matrices generated by the above symbols satisfy the hypotheses of Theorems 5.1 and 5.2, and thence are well approximated by the $C + R$ matrices. In the case of symbols f_{10} and f_{11} we observed zero residue, and that led us to the formulation of Theorem 5.1, of which we were not aware before the experiments. Table 6.2 shows the ranks of matrices R computed by the rook scheme.

Of particular interest is the dependence of "circulant ranks" on ε. Typical behaviour is shown in Table 6.3.

n	128	256	512	1024
f_1	24	23	22	20
f_2	15	18	18	17
f_3	23	23	22	20
f_4	16	17	19	18
f_5	19	21	20	15
f_6	8	9	10	10
f_7	19	23	22	20
f_8	21	24	22	20
f_9	12	12	11	11
f_{10}	4	3	3	3
f_{11}	5	4	4	4

Table 6.1. Timings(in matvecs) for construction of the C+R approximation ($\varepsilon = 10^{-7}$).

n	128	256	512	1024
f_1	36	37	38	41
f_2	19	23	23	26
f_3	28	29	32	32
f_4	20	21	23	24
f_5	27	25	22	18
f_6	17	20	22	22
f_7	27	32	38	35
f_8	28	33	34	30
f_9	15	15	12	10
f_{10}	4	4	4	4
f_{11}	6	6	6	6

Table 6.2. "Circulant ranks" ($\varepsilon = 10^{-7}$).

Once the $C + R$ approximation is found, we use C^{-1} as an explicit preconditioner. For the symmetric positive definite case it is important for C to be also symmetric and positive definite. Numerical experiments show that symmetry is maintained by our algorithm.

ε	10^{-4}	10^{-5}	10^{-6}	10^{-7}
Rank	13	19	21	24

Table 6.3. Dependence of "circulant rank" on ε, $n = 512$, symbol f_4.

However, our circulants sometimes have negative or zero eigenvalues. In this case, we improve them by setting the unwanted eigenvalues to 1. This is a low rank correction that makes the circulants positive definite. Table 6.4 shows the number of negative/zero eigenvalues for symbols $|x|^k$, $k = 1, 2, 3, 4$. We can observe that this number does not depend on n.

$\|x\|$	x^2	$\|x\|^3$	x^4
0	1	1	1

Table 6.4. Number of negative/zero eigenvalues of the constructed circulants.

Finally, Table 6.5 reports on the number of iterations required for the solution of the preconditioned system. The relative error of the solution was 10^{-6}.

n	128	256	512	1024
f_1	8	8	9	8
f_2	6	6	6	6
f_3	13	16	17	20
f_4	15	16	16	20
f_5	3	3	3	3
f_6	5	5	5	5
f_7	12	12	13	14
f_8	10	10	11	11
f_9	3	4	4	4
f_{10}	9	9	9	9
f_{11}	8	9	9	9

Table 6.5. Number of iterations.

References

[1] M. Bebendorf, S. Rjasanow and E. E. Tyrtyshnikov, Approximation using diagonal-plus-skeleton matrices, *Math. asp. of bound. elem. meth.* (Palaiseau, 1998), Chapman Hall/CRC, Boca Raton, FL, 2000, 45–52.

[2] M. Bebendorf, Approximation of boundary element matrices, *Numer. Math.*, 86 (2000), 565-589.

[3] T. F. Chan, An optimal circulant preconditioner for Toeplitz systems, *SIAM J. Sci. Statist. Comput.*, 9(1988), 766–771.

[4] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, A theory of pseudo-skeleton approximations, Linear Algebra Appl. 261: 1–21 (1997).

[5] S. A. Goreinov, E. E. Tyrtyshnikov, The maximal-volume concept in approximation by low-rank matrices, *Contemporary Mathematics*, 208 (2001), 47–51 (Ed. V. Olshevsky).

[6] D. Noutsos, S. Serra Capizzano, P. Vassalos, A preconditioning proposal for ill-conditioned Hermitian two-level Toeplitz systems, *Numer. Linear Algebra Appl.*, 12 (2005), 231–239.

[7] G. Strang, A proposal for Toeplitz matrix calculations, *Studies in Appl. Math.*, 74 (1986), 171–176.

[8] R. H. Chan, D. Potts, G. Steidl, Preconditioners for Nondefinite Hermitian Toeplitz Systems, *SIAM Matrix Anal. Appl.*, Vol. 22, No. 3 (2001), 647–665.

[9] R. Chan, M. K. Ng and A. M. Yip, A Survey of Preconditioners for Ill-Conditioned Toeplitz Systems, Structured Matrices in Mathematics, Computer Science, and Engineering II, *Contemporary Mathematics*, 281 (2001), 175–191 (Ed. V. Olshevsky).

[10] E. E. Tyrtyshnikov, A unifying approach to some old and new theorems on distribution and clustering, *Linear Algebra Appl.* 232 (1996), 1–43.

[11] E. E. Tyrtyshnikov, N. L. Zamarashkin, and A. Yu. Yeremin, Clusters, preconditioners, convergence, *Linear Algebra Appl.* 263 (1997), 25–48.

[12] E. Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method, *Computing* 64, no. 4 (2000), 367–380.

[13] N. L. Zamarashkin, I. V. Oseledets, E. E. Tyrtyshnikov, On approximation of Toeplitz matrices by the sum of a circulant and a low rank matrix, *RAS Reports*, 2005, to appear.