Fast computation of the matrix exponential for a Toeplitz matrix

Daniel Kressner, Robert Luce
Fast computation of the matrix exponential for a Toeplitz matrix

Daniel Kressner∗ Robert Luce†

Abstract

The computation of the matrix exponential is a ubiquitous operation in numerical mathematics, and for a general, unstructured $n \times n$ matrix it can be computed in $O(n^3)$ operations. An interesting problem arises if the input matrix is a Toeplitz matrix, for example as the result of discretizing integral equations with a time invariant kernel. In this case it is not obvious how to take advantage of the Toeplitz structure, as the exponential of a Toeplitz matrix is, in general, not a Toeplitz matrix itself. The main contribution of this work is an algorithm of quadratic complexity for the computation of the Toeplitz matrix exponential. It is based on the scaling and squaring framework, and connects classical results from rational approximation theory to matrices of low displacement rank. As an example, the developed methods are applied to Merton’s jump-diffusion model for option pricing.

1 Introduction

Let us consider an $n \times n$ Toeplitz matrix

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & t_0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}.$$  \hspace{1cm} (1)

In this work, we propose a new class of fast algorithms for computing a highly accurate approximation of the matrix exponential $\exp(T)$. An important source of applications for $\exp(T)$ arises from the discretization of integro-differential equations with a shift-invariant kernel. Such equations play a central role in, e.g., the pricing of single-asset options modelled by jump-diffusion processes [3, 28]. The related problem of computing the exponential of a block Toeplitz matrix appears in the Erlangian approximation of Markovian fluid queues [3].

It is well known that the multiplication of a Toeplitz matrix with a vector can be implemented in $O(n \log n)$ operations, using the FFT. This suggests the use of a Krylov subspace method, such as the Lanczos method, for computing the product of $\exp(T)$ with a vector $b$; see, e.g., [23]. For a matrix $T$ of large norm, the Krylov subspace method can be expected to converge slowly [15]. In this case, the use of rational Krylov subspace methods is advisable. For example, Lee, Pang, and Sun [20] have suggested a shift-and-invert Arnoldi method for approximating $\exp(T)b$. Every step of this method requires the solution of a linear system with a Toeplitz matrix. The fast and superfast solution of such linear systems has received broad attention in the literature; we refer to [12, 19, 24, 25] for overviews. Recent work in

∗Ecole Polytechnique Federale de Lausanne, Station 8, 1015 Lausanne, Switzerland (daniel.kressner@epfl.ch, http://anchp.epfl.ch).
†Ecole Polytechnique Federale de Lausanne, Station 8, 1015 Lausanne, Switzerland (robert.luce@epfl.ch, http://people.epfl.ch/robert.luce).
this direction includes an algorithm based on a combination of rank structured matrices and randomized sampling \cite{31}.

If, additionally, $T$ is upper triangular then $T^2$ and, more generally, any matrix function of $T$ is again an upper triangular Toeplitz matrix. This very desirable property allows for the design of efficient algorithms that directly aim at the computation of generators for exp($T$); see \cite{3} and the references therein. It is important to note that this property does not extend to general Toeplitz matrices.

The approach proposed in this work is different from existing approaches, because it aims at approximating the full matrix exponential exp($T$), instead of exp($T$)$^b$, and it does not impose additional structure on $T$. Our approach is based on a combination of the scaling and squaring method for the matrix exponential of unstructured matrices \cite{10,13,14} with approximations of low displacement rank \cite{19}. Specifically, we show that the displacement rank of a rational function of $T$ is bounded by the degree of the rational function. In turn, we obtain an approximate representation of exp($T$), which requires $O(n)$ storage under suitable assumptions and allows to conveniently multiply exp($T$) with a vector in $O(n \log n)$ operations. The latter property is particularly interesting in option pricing; it allows for quickly evaluating prices for times to maturity that are integer multiplies of a fixed time period. The availability of an approximation to the full matrix exponential also allows us to quickly access parts of that matrix. For example, the diagonal can be computed in $O(n)$ operations, which would be significantly more expensive using Krylov subspace methods.

2 Toeplitz matrices

In this section, we recall and establish basic properties of Toeplitz matrices needed for our developments. Following \cite{18}, we define the displacement $\nabla_F(A)$ of $A \in \mathbb{C}^{n \times n}$ with respect to $F \in \mathbb{C}^{n \times n}$ as

$$\nabla_F(A) := A - F A F^*.$$  

We will mostly use the downward shift matrix for $F$, in which case we omit the subindex:

$$\nabla(A) = A - Z A Z^*,$$

where

$$Z = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 1 \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \cdots & \vdots \\ 1 & 0 & \cdots & \cdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}.$$

The rank of $\nabla(A)$ is called the displacement rank of $A$. Toeplitz matrices have displacement rank at most two. Matrices of “small” displacement rank are often called Toeplitz-like matrices. Given a matrix $A$ with rank($\nabla(A)$) $\leq r$, it follows that rank($\nabla(A^{-1})$) $\leq r$. More generally, any Schur complement of $A$ enjoys this property \cite[Thm. 2.2]{18}.

It follows that the inverse of a Toeplitz matrix $T$ has displacement rank at most 2. This property does not extend to general matrix functions of $T$. In particular, exp($T$) usually has full displacement rank. However, as we will see below in Section 3.2, it turns out that matrix functions of $T$ can often be well approximated by a matrix of low displacement rank.

2.1 Generators, reconstruction, and fast matrix-vector products

For $r \geq \text{rank}(\nabla(A))$, there are matrices $G, B \in \mathbb{C}^{n \times r}$ such that

$$\nabla(A) = A - Z AZ^* = GB^*.$$  

We call such a pair $(G, B)$ a generator for $A$. Note that $A$ admits many generators and $(G, B)$ is called a minimal generator for $A$ if $r = \text{rank}(\nabla(A))$. A generator for the Toeplitz
matrix (1) is given by
\[
G = \begin{bmatrix}
t_0 & 1 \\
t_1 & 0 \\
\vdots & \vdots \\
t_{n-1} & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
1 & 0 \\
0 & i_{-1} \\
\vdots & \vdots \\
0 & i_{-n+1}
\end{bmatrix}.
\] (3)

Fast algorithms for Toeplitz-like matrices operate directly on the generator of \( A \) instead of \( A \) itself. When needed, the full matrix can be reconstructed from the generators by noting that (2) is a matrix Stein equation admitting the unique solution
\[
A = \mathcal{T}(G, B) := \sum_{k=0}^{n-1} Z^k G B^* (Z^*)^k.
\] (4)

Letting \( g_j, b_j \in \mathbb{C}^n \) for \( j = 1, \ldots, r \) denote the columns of \( G, B \), we can rewrite (4) as
\[
A = \mathcal{T}(G, B) = L(g_1) U(b_1^*) + L(g_2) U(b_2^*) + \cdots + L(g_r) U(b_r^*),
\] (5)
with the triangular Toeplitz matrices
\[
L(x) := \begin{bmatrix}
x_1 & 0 & \cdots & 0 \\
x_2 & x_1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
x_n & \cdots & x_2 & x_1
\end{bmatrix}, \quad U(x) := \begin{bmatrix}
x_1 & x_2 & \cdots & x_n \\
0 & x_1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & x_2 \\
0 & \cdots & 0 & x_1
\end{bmatrix}.
\]

Using the Fast Fourier Transform (FFT), the matrix-vector product with a Toeplitz matrix requires \( O(n \log n) \) operations; see, e.g., [2, Sec. 10.2.3]. In turn, (5) shows that the multiplication of \( A \) with a vector requires \( O(rn \log n) \) operations; see [19, Chap. 1] for more details.

### 2.2 Generator Truncation

Operations like matrix addition or multiplication typically increase the displacement rank of Toeplitz-like matrices. Even worse, the result of such operations may lead to non-minimal, that is, rank deficient generators with many more columns than necessary. To limit this increase in generator size, we will truncate the singular values of the generators. The following results justify our procedure.

**Lemma 2.1.** The displacement \( \nabla(A) \) for \( A \in \mathbb{C}^{n \times n} \) satisfies
\[
\frac{1}{2} \| \nabla(A) \|_* \leq \| A \|_* \leq n \| \nabla(A) \|_*,
\]
where \( \| \cdot \|_* \) denotes any unitarily invariant norm.

**Proof.** Note that \( \| Z^k \|_2 = 1 \) for \( 0 \leq k < n \). The first inequality follows directly from the definition of the displacement operator, viz.
\[
\| \nabla(A) \|_* = \| A - ZAZ^* \|_* \leq \| A \|_* + \| Z \|_2 \| A \|_* \| Z \|_2 \leq 2 \| A \|_*.
\]
For the second inequality we compute from (4) that
\[
\| A \|_* \leq \sum_{k=0}^{n-1} \| Z^k \|_2 \| \nabla(A) (Z^*)^k \|_2 \leq \sum_{k=0}^{n-1} \| Z^k \|_2 \| \nabla(A) \|_* \| (Z^*)^k \|_2 = n \| \nabla(A) \|_*.
\]

\[\square\]
The bounds of Lemma 2.1 may not be sharp. In particular, one may question whether the factor $n$ of the upper bound is necessary. The following example shows that this linear dependence on $n$ can, in general, not be removed.

**Example 2.2.** Let $g = [1, 1, \ldots, 1]^* \in \mathbb{R}^n$ and $A = T(gg^*)$. Then the $k$th entry of $f = Ag$ is given by $f(k) = kn - k(k - 1)/2$. Since $f$ is monotonically increasing, this allows us to estimate

$$||f||_2^2 = \sum_{k=1}^{n} f(k)^2 \geq \int_0^n f(k)^2 \, dk \geq \frac{2}{15} n^5.$$  

In turn,

$$\|A\|_2 \geq \|Ag\|_2 \|g\|_2 \geq \sqrt{\frac{2}{15} n^2} = \sqrt{\frac{2}{15} n\|\nabla(A)\|_2},$$

which shows that $\|A\|_2/\|\nabla(A)\|_2$ grows linearly with $n$.

Lemma 2.1 allows us to analyze the effect of generator truncation in terms of the approximation error.

**Theorem 2.3.** Let $A \in \mathbb{C}^{n \times n}$ be a Toeplitz-like matrix of displacement rank $r$ and consider the singular value decomposition (SVD)

$$\nabla(A) = USV^* = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix},$$

where $\Sigma_1 \in \text{diag}(\sigma_1, \ldots, \sigma_r)$ and $\Sigma_2 = \text{diag}(\sigma_{r+1}, \ldots, \sigma_r)$. Letting $\tilde{A} = T(U_1 \Sigma_1, V_1)$, it holds that

$$\|A - \tilde{A}\|_2 \leq n\sigma_{r+1} \quad \text{and} \quad \|A - \tilde{A}\|_F \leq n\sqrt{\sigma_{r+1}^2 + \cdots + \sigma_r^2}. \quad (6)$$

**Proof.** By linearity of the displacement operator $\nabla$ we find

$$\nabla(A - \tilde{A}) = \nabla(A) - \nabla(\tilde{A}) = U \Sigma V^* - U_1 \Sigma_1 V_1^* = U_2 \Sigma_2 V_2^*.$$  

Hence, the claimed bounds follow from applying Lemma 2.1 to $A - \tilde{A}$.  

Note that (6) improves upon a result by Pan [26, Eq. (3.5)].

We will use the construction of Theorem 2.3 to compress a generator $(G, B)$ with $G, B \in \mathbb{C}^{n \times r}$ to a generator $(\tilde{G}, \tilde{B})$ with $\tilde{G}, \tilde{B} \in \mathbb{C}^{n \times \tilde{r}}$ and $\tilde{r} < r$. By (6), the singular values of $GB^*$ allow us to quantify the compression error and choose $\tilde{r}$ adaptively. In the particular case of a non-minimal (rank deficient) generator $(G, B)$ with $r > \tilde{r} = \text{rank}(GB^*)$, the construction returns an exact minimal generator. Typically we have $r \ll n$, in which case the cost greatly reduces from $O(n^3)$ FLOPs for computing the SVD of $GB^*$ to $O(r^2 n + r^3)$ FLOPs by first computing thin QR decompositions of $G$ and $B$. This well-known procedure is summarized in Algorithm 1.

**Algorithm 1 Generator Compression**

**Input:** Generator matrices $G, B \in \mathbb{C}^{n \times r}$, integer $\tilde{r} < r$.

**Output:** Generator matrices $\tilde{G}, \tilde{B} \in \mathbb{C}^{n \times \tilde{r}}$ such that $T(G, B) \approx T(\tilde{G}, \tilde{B})$.

1. Compute thin QR factorizations $QR_G = G$ and $QR_B = B$.
2. Compute $S = \tilde{R}G\tilde{R}_n \in \mathbb{C}^{r \times \tilde{r}}$.
3. Compute truncated SVD $U_1 \Sigma_1 V_1^* \approx S$ with $\Sigma_1 \in \mathbb{R}^{r \times \tilde{r}}$.
4. Set $\tilde{G} = Q_G X_1 \Sigma_1^{1/2}$ and $\tilde{B} = Q_B Y_1 \Sigma_1^{1/2}$.

Generators are not uniquely determined. For every $Z \in \text{GL}_r(\mathbb{C})$, the generators $(G, B)$ and $(GZ, BZ^{-*})$ correspond to the same Toeplitz-like matrix. The following lemma shows that this relation in fact characterizes the set of all minimal generators.
Lemma 2.4. Let $A \in \mathbb{C}^{n \times n}$ be a Toeplitz-like matrix of displacement rank $r$, and let $(G_1, B_1), (G_2, B_2)$ be two minimal generators for $A$. Then there exists a matrix $Z \in \text{GL}_r(\mathbb{C})$ such that $G_1 = G_2Z$ and $B_1 = B_2Z^{-*}$.

Proof. We denote the Moore-Penrose pseudoinverse of a matrix $A$ by $A^+$. Since both generators are minimal, all matrices $G_1, G_2, B_1, B_2 \in \mathbb{C}^{n \times r}$ have rank $r$, and so we compute

\[
G_1 = G_1B_1^*(B_1^*)^+ = \nabla(T)(B_1^*)^+ = G_2B_2^*(B_1^*)^+ = G_2Z,
\]

and

\[
B_1^* = G_1^*G_1B_1^* = G_1^*\nabla(T) = \underbrace{G_1^*G_2B_2^*}_{=W} = WB_2^*.
\]

Moreover,

\[
I_r = G_2^+G_2B_2^*(B_2^*)^+ = G_2^+G_1B_1^*(B_2^*)^+ = G_2^+G_2ZWB_2^*(B_2^*)^+ = ZW,
\]

from which it follows that $W = Z^{-1}$.

\[\square\]

3 Bounds on the displacement rank of functions of Toeplitz matrices

The scaling and squaring method [13 chap. 10] for the evaluation of $\exp(T)$ takes three phases: First, the matrix $T$ is scaled by a power of two, then a Padé approximant of the scaled matrix is computed, and in a third step, the approximant is repeatedly squared in order to undo the initial scaling. In the context of Toeplitz matrices, the main challenge is to control the growth of the displacement rank in the second and third phase.

3.1 Polynomial and rational functions of Toeplitz matrices

In the following, we analyze the impact of various operations on the displacement rank of Toeplitz-like matrices and provide explicit expressions for the resulting generators. The following result is a variation of the well-known result that Schur complements do not increase the displacement rank.

Lemma 3.1 (Generator block update). Consider $M = \begin{bmatrix} D & U \\ L & M_1 \end{bmatrix} \in \mathbb{C}^{n \times n}$ with $D \in \mathbb{C}^{k \times k}$ invertible and

\[
\nabla_F(M) = GB^*, \quad G, B \in \mathbb{C}^{n \times r},
\]

for a strictly lower triangular matrix $F$. Let us partition $F = \begin{bmatrix} \hat{F} & 0 \\ F_1 \end{bmatrix}$ with $\hat{F} \in \mathbb{C}^{k \times k}$, $F_1 \in \mathbb{C}^{(n-k) \times (n-k)}$, and $G = \begin{bmatrix} \hat{G} \\ \hat{G}^* \end{bmatrix}$, $B = \begin{bmatrix} \hat{B} \\ \hat{B}^* \end{bmatrix}$ with $\hat{G}, \hat{B} \in \mathbb{C}^{k \times r}$. Then the Schur complement of $D$ in $M$ satisfies

\[
\nabla_{F_1}(M_1 - LD^{-1}U) = G_1B_1^*,
\]

where the generator matrices $G_1, B_1 \in \mathbb{R}^{(n-k) \times r}$ are defined by the relations

\[
\begin{bmatrix} 0 \\ G_1 \end{bmatrix} = G + (F - I_n) \begin{bmatrix} D \\ L \end{bmatrix} D^{-1}(I_k - \hat{F})^{-1}\hat{G}, \quad (7)
\]

\[
\begin{bmatrix} 0 \\ B_1 \end{bmatrix} = B + (F - I_n) \begin{bmatrix} D^* \\ U^* \end{bmatrix} D^{-*}(I_k - \hat{F})^{-1}\hat{B}. \quad (8)
\]

Proof. The result is a direct extension of [29 Alg. 3.3] from the Hermitian to the non-Hermitian case.

\[\square\]

It is well known that the displacement rank of the product $T_1T_2$ of two Toeplitz matrices $T_1, T_2$ is at most $4$ [17 Example 2]. The following theorem extends this result to Toeplitz-like matrices.
Theorem 3.2. Let $A_1, A_2 \in \mathbb{C}^{n,n}$ be two Toeplitz-like matrices of displacement ranks $r_1, r_2$ with generators $(G_1, B_1)$ and $(G_2, B_2)$, respectively. Then $A_1A_2$ is a Toeplitz-like matrix of displacement rank at most $r_1 + r_2 + 1$, and a generator $(G, B)$ for $A_1A_2$ is given by

$$G = \begin{bmatrix} (Z-I)A_1(Z-I)^{-1}G_2 & G_1 - (Z-I)A_1(Z-I)^{-1}e_1 \end{bmatrix},$$

$$B = \begin{bmatrix} B_2 & (Z-I)A_2(Z-I)^{-1}B_1 \end{bmatrix},$$

where $e_1 \in \mathbb{R}^n$ denotes the first unit vector. If, additionally, $e_1 \in \text{ran}(G_2) \cup \text{ran}(B_1)$ then $A_1A_2$ has displacement rank at most $r_1 + r_2$.

Proof. Consider the matrix

$$M = \begin{bmatrix} -I & A_2 \\ A_1 & 0 \end{bmatrix},$$

and set $F = Z \oplus Z$. One computes that

$$\nabla_F (M) = M - FMF^* = \begin{bmatrix} -e_1^*e_1^T & G_2B_2^* \\ G_1B_1^* & 0 \end{bmatrix} = \begin{bmatrix} G_2 & 0 & -e_1 \\ 0 & G_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & B_2 \\ B_1^* & 0 \\ e_1^* & 0 \end{bmatrix}.$$

Since $A_1A_2$ is the Schur complement of $-I$ in $M$, Lemma 3.1 implies that $A_1A_2$ has displacement rank at most $r_1 + r_2 + 1$. Moreover, by Theorem 3.2, a generator $(G, B)$ for $A_1A_2$ is given by

$$G = \begin{bmatrix} 0 & G_1 & 0 \\ (Z-I)A_1(Z-I)^{-1}G_2 & G_1 - (Z-I)A_1(Z-I)^{-1}e_1 \end{bmatrix},$$

$$B = \begin{bmatrix} B_2 & 0 & 0 \\ (Z-I)A_2(Z-I)^{-1}B_1 & (Z-I)A_2(Z-I)^{-1}e_1 \end{bmatrix}.$$

Note that at least one of these matrices becomes rank deficient if $e_1 \in \text{ran}(G_2) \cup \text{ran}(B_1)$, which shows the second part of the theorem.

Because of (3), the additional condition of Theorem 3.2 is satisfied for Toeplitz matrices. An analogous condition plays a role in controlling the displacement rank for the inverse of a Toeplitz-like matrix.

Theorem 3.3. Let $A$ be an invertible Toeplitz-like matrix of displacement rank $r$ with generator $(G, B)$. Then $A^{-1}$ is a Toeplitz-like matrix of displacement rank at most $r + 2$, and a generator $(G, B)$ is given by

$$\hat{G} = \begin{bmatrix} (Z-I)A^{-1}(Z-I)^{-1}G & (Z-I)A^{-1}(Z-I)^{-1}e_1 \\ e_1^* & e_1 \end{bmatrix},$$

$$\hat{B} = \begin{bmatrix} (Z-I)A^{-*}(Z-I)B & e_1^* & (Z-I)A^{-*}(Z-I)e_1 \end{bmatrix}.$$

If, additionally, $e_1 \in \text{ran}(G) \cup \text{ran}(B)$ then $A^{-1}$ has displacement rank at most $r + 1$.

Proof. The result follows from applying the technique from the proof of Theorem 3.2 to the embedding $M = \begin{bmatrix} A & \hat{I} \end{bmatrix}$, which has the generator $\hat{G} = \begin{bmatrix} G & 0 \\ 0 & e_1 \end{bmatrix}$, $\hat{B} = \begin{bmatrix} B & 0 \end{bmatrix}$. The second part follows from the observation that at least one of $\hat{G}, \hat{B}$ has at most rank $r + 1$ if $e_1 \in \text{ran}(G) \cup \text{ran}(B)$.

Remark 3.4. We remark that the special shape (3) of $B, G$ for a Toeplitz matrix $T$ imply that the matrix $GB^*$ in the proof of Theorem 3.3 has rank $\leq 2$ and, in turn, the displacement rank of $T^{-1}$ is $\leq 2$. In fact, letting $(G, B) = ([g e_1], [e_1 b])$ denote the generator (3) for $T$, the matrices

$$\hat{G} = \begin{bmatrix} e_1 & 0 \end{bmatrix} + (Z-I)T^{-1}(Z-I)^{-1}[-g \ e_1],$$

$$\hat{B} = \begin{bmatrix} 0 & e_1 \end{bmatrix} + (Z-I)T^{-*}(Z-I)^{-1}[e_1 \ -b].$$

constitute a generator for $T^{-1}$. This result is well known and a corollary of Lemma 3.1.
While Theorems 3.2 and 3.3 are variations of well-known results, we are not aware of existing results on the displacement ranks of powers and polynomials of Toeplitz matrices analyzed in the following.

**Lemma 3.5.** Let \( T \) be a Toeplitz matrix. Then \( T^s \) is a Toeplitz-like matrix of displacement rank at most \( 2s \) for any integer \( s \geq 1 \). Letting \( (G, B) \) denote a generator for \( T \), a sequence of (non-minimal) generators \((G_1, B_1), \ldots, (G_s, B_s) \) for \( T, T^2, \ldots, T^s \) is given by

\[
G_1 = G, \quad G_{i+1} = \left[ P_G^n G \quad P_G^{i-1} G \quad \cdots \quad G \quad -P_G e_1 \quad \cdots \quad -P_G e_1 \right] \quad (9)
\]

\[
B_1 = B, \quad B_{i+1} = \left[ B \quad P_B B \quad \cdots \quad P_B^i B \quad P_B^i e_1 \quad \cdots \quad P_B^i e_1 \right], \quad (10)
\]

for \( i = 1, \ldots, s-1 \), where \( P_G := (Z - I) T (Z - I)^{-1} \) and \( P_B := (Z - I) T^*(Z - I)^{-1} \). Moreover,

\[
e_1 \in \text{ran}(G_1) \subset \cdots \subset \text{ran}(G_s) \quad \text{and} \quad e_1 \in \text{ran}(B_1) \subset \cdots \subset \text{ran}(B_s). \quad (11)
\]

**Proof.** The proof is by induction on \( i \). For \( G_1, B_1 \), the claim (11) follows directly from the expression (3) for the generator of \( T \).

Now assume that (9), (11) hold for all \( G_i, B_i \), with \( i < s \). Invoking Theorem 3.2 with \( T_1 = T^i \) and \( T_2 = T \) yields the formulas (9)–(10). Further, since all columns of \( G_i \) and \( B_i \) are also columns of \( G_{i+1} \) and \( B_{i+1} \), respectively, we directly obtain \( \text{ran}(G_i) \subset \text{ran}(G_{i+1}) \) and \( \text{ran}(B_i) \subset \text{ran}(B_{i+1}) \).

Finally, since \( e_1 \in \text{ran}(G) \), each of the last \( i \) columns of \( G_{i+1} \) and \( B_{i+1} \) is a linear combination of one of the first \( i + 1 \) block columns. In turn,

\[
\text{ran}(G_{i+1}) = \text{ran}\left( \left[ P_G^n G \quad P_G^{i-1} G \quad \cdots \quad G \right] \right),
\]

\[
\text{ran}(B_{i+1}) = \text{ran}\left( \left[ B \quad \cdots \quad B^i B \right] \right),
\]

which implies \( \text{rank}(G_{i+1}) \leq 2(i + 1) \) and \( \text{rank}(B_{i+1}) \leq 2(i + 1) \). In particular, the displacement rank of \( T^s \) is at most \( 2(i + 1) \). \( \square \)

**Theorem 3.6.** Let \( T \) be a Toeplitz matrix and \( p \in \mathcal{P}_s \), where \( \mathcal{P}_s \) denotes the set of polynomials of degree at most \( s \). Then \( p(T) \) is a Toeplitz-like matrix of displacement rank at most \( 2s \).

**Proof.** Let \( p = \sum_{k=0}^{s} a_k z^k \) and consider the generators \((G_i, B_i)\), \( 1 \leq i \leq s \), for the monomials \( T^i \) constructed in (9)–(10). Setting \( G_0 := B_0 := e_1 \) and using the linearity of the displacement operator \( \nabla \) we obtain

\[
\nabla(p(T)) = \sum_{i=0}^{s} a_i \nabla(T^i) = \left[ a_0 e_1 \quad a_1 G_1 \quad \cdots \quad a_s G_s \right] =: G_p B_p^*.
\]

It follows from (11) that \( \text{rank}(G_p) = \text{rank}(G_s) \leq 2s \), \( \text{rank}(B_p) = \text{rank}(B_s) \leq 2s \), and hence \( \text{rank}(\nabla(p(T))) \leq 2s \). \( \square \)

The following theorem is the main result of this section and quantifies the effect of a rational function on the displacement rank. It shows that the displacement rank grows at most linearly with the degree of the rational function, defined as the maximal degree of the numerator and denominator.

**Theorem 3.7.** Let \( T \) be a Toeplitz matrix, and let \( p \in \mathcal{P}_{s_p}, q \in \mathcal{P}_{s_q} \) be such that \( q(T) \) is invertible. Let \( (G_p, B_p) \) and \( (G_q, B_q) \) denote generators of \( p(T) \) and \( q(T) \), respectively. Then \( r(T) = p(T) q(T)^{-1} \) is a Toeplitz-like matrix of displacement rank at most \( 2 \max\{s_p, s_q\} + 1 \), and a generator is given by

\[
G = \left[ -(Z - I)^{-1} q(T)^{-1} G_q \quad (Z - I) q(T)^{-1} (Z - I)^{-1} G_p \quad e_1, \right] \quad (12)
\]

\[
B = \left[ (Z - I) p(T)^* q(T)^{-1}(Z - I)^{-1} B_q \quad B_q \quad (Z - I) p(T)^* q(T)^{-1}(Z - I)^{-1} e_1 \right]. \quad (13)
\]
Proof. The Schur complement of the leading diagonal block in the embedding \( M = \begin{bmatrix} -q(T) & p(T) \\ \end{bmatrix} \), is \( q(T) \begin{bmatrix} -1 \\ I \end{bmatrix} \), and setting \( F = \begin{bmatrix} Z^T \\ Z^T \end{bmatrix} \) one computes

\[
M - F M F^* = \begin{bmatrix} -G_q B_q^* & G_p B_p^* \\ e^T_1 e^T_1 \end{bmatrix} = \begin{bmatrix} -G_q & G_p \\ 0 & e^T_1 \end{bmatrix} \begin{bmatrix} B_q^* & 0 \\ 0 & B_p^* \end{bmatrix}.
\]

(14)

The formulas (12) and (13) are obtained by applying Lemma 3.1.

To see that the matrix (14) has rank at most \( 2 \max\{s_p, s_q\} + 1 \), we recall from (11) that the ranges of the generator matrices for monomials are nested and thus Theorem 3.6 implies \( \text{rank} \left[ -G_q G_p \right] \leq 2 \max\{s_p, s_q\} \).

3.2 Low displacement-rank approximation of matrix exponential

Theorem 3.6 allows us to derive a priori bounds on the numerical displacement rank of \( \exp(T) \), using rational approximations of the exponential function. To see this, let us first recall a seminal result by Gonchar and Rakhmanov [7].

Theorem 3.8. There is a constant \( C \) such that

\[
\inf_{p_1, p_2 \in P_s} \max_{\lambda \in (-\infty, 0]} |e^\lambda - p_1(\lambda)/p_2(\lambda)| \leq CV^{-s}
\]

holds for all \( s \geq 1 \) with \( V = 9.28903 \ldots \).

Corollary 3.9. Let \( T \in \mathbb{C}^{n,n} \) be a diagonalizable Toeplitz matrix with all eigenvalues real and contained in \((-\infty, \mu]\) for some \( \mu \in \mathbb{R} \). Then

\[
\min\{\|\exp(T) - A\|_2 : \text{rank}(\nabla(A)) \leq 2s + 1\} \leq \tilde{C}V^{-s},
\]

for \( \tilde{C} = \kappa(X)e^{\mu}C \), where \( C, V \) are as in Theorem 3.8 and \( \kappa(X) \) is the condition number of a matrix \( X \) such that \( X^{-1}TX \) is diagonal.

Proof. According to Theorem 3.7 the matrix \( A = e^\mu p_2(T)^{-1}p_1(T) \) with \( p_1, p_2 \in P_s \) has displacement rank at most \( 2s + 1 \). From

\[
\|\exp(T) - A\|_2 = \|\exp(\mu I)\exp(T - \mu I) - A\|_2 \\
\leq \kappa(X)e^{\mu} \max_{\lambda \in (-\infty, 0]} |e^\lambda - p_1(\lambda)/p_2(\lambda)|,
\]

the result follows using Theorem 3.8.

Corollary 3.9 implies that the singular values of \( \nabla(\exp(T)) \) decay at least exponentially to zero, with a decay rate that does not deteriorate even if \( T \) has very small eigenvalues. This property is retained by approximations to the matrix exponential.

Corollary 3.10. Under the assumptions of Corollary 3.9 let \( B \in \mathbb{C}^{n \times n} \) satisfy \( \|B - \exp(T)\|_2 \leq \tau \) for \( \tau \geq 0 \). If \( s \) is an integer such that

\[
\tilde{C}V^{-s} \leq \tau,
\]

then

\[
\min\{\|B - A\|_2 : \text{rank}(\nabla(A)) \leq 2s + 1\} \leq 2\tau
\]

Proof. The result follows from the triangular inequality

\[
\|B - A\|_2 \leq \|B - \exp(T)\|_2 + \|\exp(T) - A\|_2 \leq 2\tau,
\]

and Corollary 3.9.
The case of complex spectra is more difficult. A common approach to obtain rational approximations is to consider the contour integral representation

$$\exp(T) = \frac{1}{2\pi i} \int_{\Gamma} e^{z(I - T)^{-1}} \, dz,$$

where $\Gamma$ is a contour enclosing the spectrum of $T$. Applying numerical quadrature with $s$ points to (16) yields an approximation $r(T) \approx \exp(T)$, where $r$ is a rational function of degree $s$ and hence $r(T)$ has displacement rank at most $2s + 1$ by Theorem 3.7. In the absence of information on the spectrum of $T$, one might choose $\Gamma$ to be a circle of radius larger than $\|T\|_2$. Applying the composite trapezoidal rule yields exponential convergence but the convergence rate deteriorates as $\|T\|_2$ grows; see, e.g., [30]. Sometimes, much better results can be obtained if more information on the spectrum is available. For example, if $A$ is sectorial (that is, its eigenvalues are contained in a sector strictly contained in the left half plane), López-Fernández et al. [21, Thm. 1] establish a bound of the form

$$\| \exp(T) - r(T) \|_2 \leq C \gamma^s$$

where the rate $0 < \gamma < 1$ depends on the opening angle of the sector but not on the norm of $A$. The rational function $r$ has degree $s$ and is obtained by applying quadrature to (16) with $\Gamma$ chosen to be the left branch of a hyperbola. Analogous results hold for the case that the numerical range of $A$ is contained in the open left half complex plane; see [9, Sec. 4.2] for an overview.

If $T$ has large norm and eigenvalues on or close to the imaginary axis then it cannot be expected that $\exp(T)$ admits a good approximation of low displacement rank. In turn, the methods developed in this paper are not efficient for this type of matrices. The following example illustrates such a situation.

Example 3.11. Let $T \in \mathbb{R}^{2000 \times 2000}$ be a skew-symmetric Toeplitz matrix $[1]$ with $t_1 = 1$, $t_{-1} = -1$ and all other entries zero. The following table shows the numerical displacement rank of $\exp(\alpha T)$, defined as the number of singular values of $\nabla(\exp(\alpha T))$ larger than $10^{-10}$ times the first singular value:

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>num. displacement rank</td>
<td>11</td>
<td>29</td>
<td>153</td>
<td>1309</td>
</tr>
</tbody>
</table>

Clearly, as $\alpha$ grows it becomes increasingly difficult to approximate $\exp(\alpha T)$ by a Toeplitz-like matrix.

4 Algorithmic tools

In Section 5, we will adapt two variants of the scaling and squaring method for Toeplitz matrices. The algorithmic tools needed for an efficient implementation are the same for both, and we will describe them in this section without making reference to either algorithm.

4.1 Norm estimation and scaling

The first step of the scaling and squaring method consists of determining a scaling parameter $\rho \in \mathbb{N}$ such that $\|2^{-\rho}T\| \approx 1$, which necessitates computing $\|T\|$ or an estimate thereof.

Since matrix-vector products with a Toeplitz matrices can be carried out in $O(n \log n)$ operations, the power method for estimating $\|T\|_2$ can be implemented with the same complexity per iteration. Alternatively, $\|T\|_1$ can be computed at little cost.

Lemma 4.1. Let $T \in \mathbb{C}^{n \times n}$ be a Toeplitz matrix. Then $\|T\|_1$ can be computed in $O(n)$ operations.
Proof. Let us denote the first column and row of $T$ by $c$ and $r$, respectively, and set $\mu_j := \|Te_j\|_1$, $1 \leq j \leq n$. From the structure of $T$ we find that
\[
\mu_1 = \sum_{i=1}^{n} |c_i|, \quad \mu_{j+1} = \mu_j - |c_{n-j+1}| + |r_j| \text{ for } 1 \leq j < n,
\]
and hence $\|T\|_1 = \max_{1 \leq j \leq n} \{\mu_j\}$ can be computed in $O(n)$ operations. \hfill \Box

Once the scaling parameter $\rho$ is determined, the generator of $T$ is scaled accordingly, which obviously requires only $O(n)$ operations.

4.2 Fast solution of Toeplitz and Toeplitz-like systems

In order to compute generators for rational functions of Toeplitz matrices, we need to solve linear systems of equations with Toeplitz and Toeplitz-like matrices; see Theorem 3.7. We will now briefly summarize a well established technique for the solution of such systems in linear time (the “GKO algorithm” \[6\]).

Let $A \in \mathbb{C}^{n \times n}$ be a Toeplitz-like matrix of displacement rank $r \ll n$, so that $A$ satisfies the matrix Stein equation \[2\] with a low-rank right hand side. It is well known (see, e.g., \[8\] sec. 0.2 and the references therein) that $T$ also satisfies numerous other matrix equations, including the Sylvester equation
\[
\Delta Z_{1,1}(T) := Z_1 T - T Z_{-1} = \hat{G} \hat{B}^*,
\]
with low-rank right-hand side and $Z_{\delta} := Z + \delta e_1 e_n^*$.\footnote{The Sylvester operator matrices $D_1, D_2$ are diagonal matrices. The transformation between \[17\] and \[18\] involves only FFTs and diagonal scalings. The fact that the Sylvester operator matrices $D_1$ and $D_2$ are now diagonal allows for pivoting within the generalized Schur algorithm, requiring in total $O(n^2)$ operations. Combined with further safeguarding techniques one obtains an efficient algorithm for the solution of linear systems with $A$ that enjoys similar stability properties as traditional Gaussian elimination with pivoting \[5\].}

One could directly apply the generalized Schur algorithm \[18\] to either representation \[2\] or \[17\] in order to solve linear systems with $A$ in $O(n^2)$ operations, but without further assumptions on $A$, such as well-conditioned leading principal submatrices, or more involved algorithmic techniques \[4\] a numerically stable solution is not guaranteed.

Instead, we propose to use a transformation \[6\] prop. 3.1] (see also \[11\]) of \[17\] to a Cauchy-like Sylvester displacement equation
\[
D_1 C - CD_2 = \hat{G} \hat{B}^*,
\]
with the same displacement rank and where $D_1, D_2$ are diagonal matrices. The transformation between \[17\] and \[18\] involves only FFTs and diagonal scalings.

For our purpose of evaluating rational matrix functions of Toeplitz matrices only one minor technicality needs to be resolved: The generator matrices $G, B$ with respect to the matrix Stein equation \[2\] need to be transformed to generator matrices with respect to the Sylvester equation \[17\]. The following lemma shows that the corresponding displacement rank increases at most by two.

\textbf{Lemma 4.2.} Let $A \in \mathbb{C}^{n \times n}$ be a Toeplitz-like matrix, and let $[\alpha \, \beta ]$ and $[r \, a ]$ denote its last column and row, respectively. If $T$ denotes the Toeplitz matrix with first column $[\alpha \, \beta ]$ and first row $[r \, a ]$ then
\[
\Delta Z_{1,1}(A) = (\nabla(T) - \nabla(A)) Z_{-1}.
\]

\textbf{Proof.} One directly calculates that
\[
\Delta Z_{1,1}(A) Z_{-1} = Z_1 A Z_{-1}^* - A = (Z + e_1 e_n^*) A (Z^* + e_n e_1^*) - A
\]
\[
= Z A Z^* - A + e_1 e_n^* A Z^* - Z A e_n e_1^* - e_1 e_n^* A e_n e_1^*
\]
\[
= -\nabla(A) + \begin{bmatrix} \alpha & r \\ c & 0_{n-1, n-1} \end{bmatrix} = -\nabla(A) + \nabla(T).
\]
To compute a $\Delta z_1, z_{-1}$ generator for $A$ using Lemma 4.2, one needs to reconstruct the last column and row of $A$ from a generator with respect to $\nabla$. According to (9) this requires $2r$ matrix-vector multiplications with triangular Toeplitz matrices and can hence be computed in $O(rn \log n)$ operations. We can summarize the preceding discussion as follows.

**Corollary 4.3.** Let $A \in \mathbb{C}^{n \times n}$ be a Toeplitz-like matrix of displacement rank $r$. Then linear systems with $A$ can be solved in $O(rn^2)$ operations.

### 4.3 Computing generators of Toeplitz matrix polynomials

In Lemma 4.3 we computed explicit expressions for generators of monomials $T, T^2, T^3, \ldots$. Within these expressions, one needs to (repeatedly) apply the matrices

$$(Z - I)T(Z - I)^{-1} \quad \text{and} \quad (Z - I)T^s(Z - I)^{-1}$$

to a given canonical Toeplitz generator $(G, B)$. Note that applying $(Z - I)^{-1}$ to a vector amounts simply to computing the vector of its cumulative sums, and that the application of $Z - I$ to a vector can be evaluated with $n - 1$ subtractions. Hence, both operations require $O(n)$ operations. Finally, as mentioned in Section 2, matrix-vector products with $T$ and $T^s$ can be evaluated in $O(n \log n)$ operations, so that we have the following result.

**Corollary 4.4.** Let $T \in \mathbb{R}^{n \times n}$ be a Toeplitz matrix, then a set of generators for the monomials $T, T^2, \ldots, T^s$ can be computed with $O(sn \log n)$ operations.

Because of the nested structure of the monomial generators (cf. (9)–(10)), only the generator $(G_s, B_s)$ for the leading monomial $T^s$ is actually needed for the evaluation of $p(T) := \sum_{k=0}^s a_k T^k$. A generator for $p(T)$ can be computed by appropriate linear combination of the block columns of $G_s$, i.e., there exists a matrix $X \in \mathbb{R}^{3s-1, 3s-1}$ defined through the coefficients $a_0, \ldots, a_s$, such that $(G_s X, B_s)$ is a generator for $p(T)$. For example, if we set

$$X = \begin{bmatrix} a_2 I_2 & 0 & 0 \\ 0 & a_2 I_2 & 0 \\ 0 & 0 & a_2 \end{bmatrix} + \begin{bmatrix} 0 & a_1 I_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

then $(G_2 X, B_2)$ is a generator for $a_2 T^2 + a_1 T$.

Alternatively, a Horner-like scheme can be used to compute a generator for $p(T)$. Let $T_k, 0 \leq k \leq s$ be the $k$th Horner polynomial, defined via the recursion

$$T_0 := a_0 I, \quad T_k := T T_{k-1} + a_{s-k} I \quad \text{for} \quad 1 \leq k \leq s,$$

then $T_k$ is the Schur complement of $-I$ in the embedding

$$M = \begin{bmatrix} -I & T_{k-1} \\ T & a_k I \end{bmatrix}.$$

(19)

Using similar arguments and techniques as for the evaluation of monomials in $T$, it follows that the evaluation of $p(T)$ based on (19) can be carried out in $O(sn \log n)$ operations. In contrast to the evaluation based on monomials of $T$, the resulting generator has length $2s$.

### 4.4 Evaluating rational approximants by solving Toeplitz-like systems

We now turn to the computation of generators for rational functions of $T$. Let $r(z) = \frac{p(z)}{q(z)}$ be a rational function, and let $(G_p, B_p)$ and $(G_q, B_q)$ be generators for $p(T)$ and $q(T)$,
respectively; see Section 4.3. From equations (12)-(13), we see that a generator for \( r(T) = q(T)^{-1}p(T) \) is found by solving the linear Toeplitz-like systems
\[
q(T)^{-1}(Z - I)^{-1} \begin{bmatrix} G_q & G_p \end{bmatrix} \quad \text{and} \quad q(T)^{-*}(Z - I)^{-1} \begin{bmatrix} B_q & e_1 \end{bmatrix}.
\] (20)
In total there are 2 \( \deg(q) + \deg(p) + 1 \) right hand sides to solve for, and since the displacement rank of \( q(T) \) is at most 2 \( \deg(q) \), the techniques outlined in Section 4.2 yield the following result.

**Corollary 4.5.** Let \( T \in \mathbb{C}^{n \times n} \) be a Toeplitz matrix, and \( r(z) = \frac{p(z)}{q(z)} \) a rational function of degree \( s = \max\{\deg(p), \deg(q)\} \). Then a generator for \( r(T) = q(T)^{-1}p(T) \) can be computed with \( O(s^2n^2) \) operations.

The dependence on \( s^2 \) in the above statement is nearly negligible in our context, since the degrees of the Padé approximants we will be using are typically small, and never larger than thirteen. Note also that (12)–(13) involve matrix-vector multiplications with \( p(T) \) and \( p(T^*) \), but the cost for these are dominated by solving the linear systems (20).

### 4.5 Evaluating rational approximants by partial fraction expansion

Any rational function with simple poles can be expressed as a partial fraction expansion
\[
r(z) = \sum_{i=1}^{m} \frac{\beta_i}{z - \alpha_i} + p(z),
\] where \( m \) is the number of poles \( \alpha_i \) with residues \( \beta_i \), and \( p \) is a polynomial. Let \( T \in \mathbb{C}^{n \times n} \) be a Toeplitz matrix, and assume that none of the poles of \( r \) is an eigenvalue of \( T \). Generators for \( p(T) \) can be computed using the techniques described in Section 4.3, and we now discuss the computation of generators for the Toeplitz-like matrix
\[
\sum_{i=1}^{m} \beta_i (T - \alpha_i I)^{-1}.
\] (21)

First note that for any \( \alpha \in \mathbb{C} \) the matrix \( T_{\alpha} = T + \alpha I \) is a Toeplitz matrix as well, with the generator
\[
G_{\alpha} = \begin{bmatrix} t_0 + \alpha & 1 \\ t_1 & 0 \\ \vdots & \vdots \\ t_{n-1} & 0 \end{bmatrix}, \quad B_{\alpha} = \begin{bmatrix} 1 & 0 \\ 0 & t_{-1} \\ \vdots & \vdots \\ 0 & t_{-n+1} \end{bmatrix}.
\]

Hence, the evaluation of (21) is simply the sum of \( m \) inverse Toeplitz matrices and we can therefore apply the result from Remark 3.4 and the related Gohberg-Semencul formulas (see, e.g., [16] for an overview) to compute a generator for (21) by solving \( O(m) \) linear Toeplitz systems using the technique described in Section 4.2.

In the common case where \( T \) is real, and the expansion (21) involves pairs of complex conjugates shifts \( \alpha, \bar{\alpha} \) and residues \( \beta, \bar{\beta} \), then
\[
\beta(T - \bar{\alpha})^{-1} = \bar{\beta}(T - \alpha)^{-1},
\] so that a real generator for \( \beta(T - \alpha)^{-1} + \bar{\beta}(T - \bar{\alpha})^{-1} \) can be computed by means of solving two linear Toeplitz systems (instead of four). So let \((G, B)\) be a generator for \( \beta(T - \alpha)^{-1} \), then
\[
\nabla (\beta(T - \alpha)^{-1} + \bar{\beta}(T - \bar{\alpha})^{-1}) = \nabla (\beta(T - \alpha)^{-1}) + \overline{\nabla (\beta(T - \alpha)^{-1})} = GB^* + \overline{G\bar{B}^*} = 2 \text{Re}(GB^*) = 2(\text{Re}(G)\text{Re}(B)^* + \text{Im}(G)\text{Im}(B)^*).
\]

In the case of a Padé approximant, this implies that the number of Toeplitz matrix inversions is roughly halved. Of course, the asymptotic cost is unchanged, and we summarize our findings as follows.
Corollary 4.6. Let $T \in \mathbb{C}^{n \times n}$ be a Toeplitz matrix, and $r(z) = \sum_{i=1}^{n} \beta_i(z - \alpha_i)$ a rational function. Then $r(T)$ can be evaluated in $O(mn^2)$ operations involving at most $2m + 2$ solutions of linear Toeplitz systems.

4.6 Iterative squaring

At the final stage of scaling and squaring algorithms we have at hand a rational approximation $r(2^{-\rho}T) \approx \exp(2^{-\rho}T)$, and in order to obtain an approximation for $\exp(T)$, the initial scaling is undone by squaring the matrix $r(2^{-\rho}T)$ $\rho$ times.

Let $(G, B)$ be a generator of length $s$ for the Toeplitz-like matrix $A$. Then a generator of length $2s + 1$ for $A^2$ can be computed using Theorem 3.2. The cost for computing the new generator is dominated by the evaluation of the products $AG$ and $A^*B$. Each of these products can be computed based on the expansion (5), involving $2s^2$ multiplications with triangular Toeplitz matrices, resulting in an operation of complexity $O(s^2 n \log n)$.

Each squaring operation effectively doubles the length of the generator matrices, and hence after $\rho$ squaring operations we would obtain generator matrices of length $O(2^s s)$, which is computationally feasible only for tiny values of $\rho$. However, if the spectrum of $T$ allows for low degree rational approximation of $e^z$ (see Sec. 3), the same is true for each scaled matrix $2^{-k}T$, $1 \leq k \leq \rho$. Consequently, if the rational approximation to $\exp(2^{-\rho}T)$ is such that

$$r(2^{-\rho}T)^2^k \approx \exp(2^{-k}T), \quad 0 \leq k \leq \rho,$$

then Corollary 3.10 shows that each displacement $\nabla(r(2^{-\rho}T)^2^k)$ is close to a low rank matrix, and a generator compression (Alg. 1) applied after every squaring operation will reduce the intermediate generator length back to $O(s)$, without compromising the approximation quality of the final approximation to $\exp(T)$. The computational cost for each of this compressions is dominated by the matrix multiplications $AG$ and $A^*B$. The following Corollary summarizes the discussion.

Corollary 4.7. Generators for the sequence $r(2^{-\rho}T)^2, \ldots, r(2^{-\rho}T)^2^\rho$ can be computed in $O(\rho s^2 n \log n)$, provided that each intermediate displacement $\nabla(r(2^{-\rho}T)^2), \ldots, \nabla(r(2^{-\rho}T)^2^\rho)$ has numerical rank $O(s)$.

4.7 Reconstruction of Toeplitz-like matrices

As mentioned in Section 2 a Toeplitz-like matrix can be reconstructed from a generator based on $[3]$. We note next that this operation can be implemented efficiently.

Lemma 4.8. Let $A \in \mathbb{C}^{n \times n}$ be a Toeplitz-like matrix, and $(G, B)$ a generator for $T$ of length $r$. Then $A$ can be computed from $(G, B)$ in $O(rn^2)$ operations.

Proof. The number of operations for computing $D = GB^*$ is in $O(rn^2)$. Then the expression

$$A = \sum_{k=0}^{n-1} Z^k D(Z^*)^k$$

can be evaluated in $O(n^2)$ operations by noting that the $k$th subdiagonal (superdiagonal) of $A$ is just the vector of cumulative sums of the $k$th subdiagonal (superdiagonal) of $D$. □

4.8 A remark on the use of the FFT

Many of the operations discussed in this section involve or even reduce to matrix-vector multiplication with Toeplitz and Toeplitz-like matrices, cf. Secs. 4.1, 4.3, 4.4, 4.6. The complexity of computing a matrix-vector product $Ax$ for a Toeplitz-like matrix $A \in \mathbb{C}^{n \times n}$ of displacement rank $r$ is $rn \log n$. Although carrying out these multiplications using the FFT
is asymptotically faster than standard matrix-vector multiplication, an actual computational advantage is gained only for sufficiently large matrix dimension $n$.

If that is not the case, it is preferable to resort to standard matrix-vector multiplication. Since $A$ is of displacement rank $r$, the reconstruction of $A$ from a generator can be done in $O(rn^2)$ operations, as described in Sec. 4.7. Consequently, a single matrix-vector product can be computed using standard multiplication in $O(rn^2 + n^2)$ operations, and $\ell$ matrix-vector products with $A$ can be evaluated in $O((r+\ell)n^2)$ operations. Note that in this latter case, the cost for FFT based multiplication is in $O(\ell n \log n)$.

## 5 Scaling and squaring algorithms for Toeplitz matrices

In Section 3 we have shown that rational approximations to the matrix exponential of a Toeplitz matrix $T$ enjoy low displacement rank, provided that $T$ is negative real or sectorial. We will now use scaling and squaring algorithms that take advantage of this property. Based on the techniques presented in Section 4 the resulting algorithms will require $O(n^2)$ operations for computing $\exp(T)$, which is optimal since the output is also of size $n^2$.

Denote by $r_{k,m}(z) = \frac{p_{k,m}(z)}{q_{k,m}(z)}$ the $(k,m)$-Padé approximant to the exponential function, meaning that the numerator polynomial is of degree $k$, and the denominator polynomial of degree $m$. Scaling and squaring algorithms take advantage of the fact that Padé approximations are very accurate close to origin. An input matrix $A$ is thus scaled by a power of two, so that $\|2^{-\rho}A\| \lesssim 1$, and then the Padé approximant $r_{k,m}(2^{-\rho}A)$ is computed. Finally, an approximation to $\exp(A)$ is obtained by squaring the result repeatedly, viz.

$$\exp(A) \approx r_{k,m}(2^{-\rho}A)^{2^p},$$

using the identity $\exp(A) = \exp(\sigma^{-1}A)^{\sigma}$, $\sigma \in \mathbb{C} \setminus \{0\}$.

Different strategies for choosing the scaling parameter $\rho$ and the Padé degree $(k,m)$ yield different methods. We will discuss two recently proposed scaling and squaring methods. The first one, described by Higham [14], is based on a diagonal Padé approximation of degree at most 13 and makes no assumption on the spectrum of the input matrix. The second one by Güttel and Nakatsukasa [10] employs a subdiagonal Padé approximations of much smaller degree, and is particularly useful for the case where $A$ has only negative real eigenvalues.

### 5.1 A diagonal scaling and squaring method

The scaling and squaring method designed by Higham [14] was until recently the default method to compute the matrix exponential in Matlab, available via the command `expm`. It scales the input matrix $A$ so that $\|2^{-\rho}A\| \lesssim 5.4$, and approximates $\exp(A)$ using a diagonal Padé approximation, i.e., $k = m$ in the notation from above. The approximation degree is at most 13, or less for matrices that need not be scaled. This parameter choice is designed such that the approximation error can be interpreted as a backward error $E$ (in any consistent matrix norm)

$$r_{m,m}(2^{-\rho}A)^{2^p} = \exp(A+E), \|E\| \leq u\|A\|,$$

where $u$ denotes the unit roundoff in double precision; see [14, Thm. 2.1]. Note that no assumption on the spectral properties of $A$ have been made. The matrix $E$ can be shown to commute with $A$, and hence one obtains immediately

$$\|r_{m,m}(2^{-\rho}A)^{2^p} - \exp(A)\| \leq \||\exp(\rho)\exp(E) - I\|| \leq \||\exp(A)\||\|E\||\exp(\rho)\| \leq \||\exp(A)\||u\|A\| e^{\rho\|A\|},$$

which bounds the forward error of the approximation. At the same time, Higham shows that the matrix $q_{m,m}(A)$ is well conditioned under this parameter regime.
assume that \( \exp(T) \) is Hermitian negative definite, so that 

\[
\| \exp(T) \| \leq 1.
\]

As the degree \( s \) depends only logarithmically on the norm of \( T \), the approximation of \( \exp(T) \) produced by this scaling and squaring method can be expected to be of low numerical displacement rank.

We now explain why the assumption (22) is satisfied for this scaling and squaring method. As can be seen from [14, Alg. 2.3], the input matrix is not scaled if \( \| A \|_1 \leq 5.4 \). Otherwise, \( \rho \) is chosen as the smallest integer satisfying \( \| 2^{-\rho} A \|_1 \leq 5.4 \), and the Padé approximant is chosen independently of \( \rho \). Consequently all the intermediate exponential approximations obtained during the squaring phase satisfy a bound like (23), implying that all the intermediate displacement ranks are bounded. A numerical example is discussed in Section 6.2; see Figure 4.

A practical algorithm of quadratic complexity is obtained by replacing the unstructured matrix computation operations used in [14] by their structured counterparts explained in Sec. 4. The resulting method is shown in Algorithm 2.

We close the discussion by noting that Algorithm 2 almost achieves our goal of designing a method of quadratic complexity for the Toeplitz matrix exponential. While the approximation degree \( m \) is bounded by 13, the scaling power \( \rho \) still grows logarithmically with \( \| A \|_1 \), and consequently the number of squaring iterations is not bounded independently of the numerical values of \( A \). From a practical point of view Algorithm 2 still behaves like a quadratic algorithm.

### 5.2 A subdiagonal scaling and squaring method

The other scaling and squaring method we adapted for the Toeplitz case is described and analyzed in [10]. The method is designed for matrices whose spectrum is contained in the negative real line, or located close to it. In contrast to Higham’s method, it (typically) employs a subdiagonal Padé approximation (“sexpm”). Compared to \texttt{exmp}, \texttt{sexmp} has several attractive features:

1. For the Padé degree \( (k, m) \) we have \( k, m \leq 5 \), resulting in computational cost savings.
2. The Padé approximant can be evaluated as a partial fraction expansion, thus involving only solves with Toeplitz matrices instead of Toeplitz-like matrices.

3. The number of scaling iterations \( \rho \) is bounded by four, independently of the norm of \( A \). Hence, the potential displacement rank growth as discussed in Section 4.6 is not an issue for this method.

If \( A \in \mathbb{C}^{n,n} \) is normal, the approximation \( B \) of \( \exp(A) \) obtained through \texttt{sexpm} satisfies

\[
\|B - \exp(A)\|_2 \leq u \|A\|_2,
\]

and the authors in fact show that their method is a forward stable method \[10, \text{Thm. 4.1}\]. The adaption to the Toeplitz case, coined \texttt{sexpmt}, is shown in Alg. 3.

6 Numerical experiments

We have implemented Algorithms 2 and 3 in Matlab. For solving the Toeplitz-like systems as described in Section 4.2, we are using the \texttt{drsolve} package \[1\]. All experiments were conducted on standard Linux box using a single computational thread.

6.1 Exact error on small matrices

As a first test we compute the exact normwise error of the approximation of \( \exp(T) \) via Algorithm 2 on a diverse set of small Toeplitz matrices, from the following sources:

- The 16 Toeplitz matrices available via the \texttt{smtgallery} command of the structured matrix toolbox \[27\].

- Seven matrices from \[20, \text{sec. 5}\]. Specifically, we generated matrices according to Examples 1 and 2 from \[20\] for time steps 1, 10 and 100 as well as one instance of Example 3 from \[20\] with time step 1. The last example refers to the Merton model, which is considered further in Section 6.2.

\[\texttt{http://bugs.unica.it/~gppe/soft/#drsolve}\]
\[\texttt{http://bugs.unica.it/~gppe/soft/smt/}\]
Figure 1 shows the normwise relative errors
\[ \frac{\|\exp(T) - \expmt(T)\|_F}{\|\exp(T)\|_F}, \]
where \( \expmt(T) \) denotes the computed approximation to \( \exp(T) \) obtained from Algorithm 2. The “exact” \( \exp(T) \) was computed using Matlab’s variable precision arithmetic with 150 digits. Further, we show for each matrix in the set an approximation to the relative condition number of the exponential condition number \([13\text{, chap. 10}]\) (black line). The errors of a backward stable method would realize errors close to this line, and we see that the errors of \( \expmt \) are roughly bounded by ten times this quantity.

### 6.2 Option pricing using the Merton model

We now turn to the evaluation of option prices in the Merton model, for one single underlying asset \([22]\). There, in contrast to the Black-Scholes model, the expected return of the asset evolves as a mixture of continuous and jump processes. The option value \( \omega(\xi, t) \) on \((\mathbb{R}, [0,T]) \) satisfies the partial integro-differential equation (PIDE)

\[
\omega_t = \frac{\nu^2}{2} \omega_{\xi\xi} + \left( r - \lambda \kappa - \frac{\nu^2}{2} \right) \omega_{\xi} - (r + \lambda) \omega + \lambda \int_{-\infty}^{\infty} \omega(\xi + \eta, t) \phi(\eta) d\eta, \tag{24}
\]

where \( T \) denotes time to maturity, \( \nu \geq 0 \) is the volatility, \( r \) is the risk-free interest rate, \( \lambda \geq 0 \) is the arrival intensity of a Poisson process, \( \phi \) is the normal distribution with mean \( \mu \) and standard deviation \( \sigma \), and \( \kappa = e^{\mu + \sigma^2/2} - 1 \).

The discretization of (24) first truncates the infinite domain \((\mathbb{R}, [0,T])\) to \((\xi_{\text{min}}, \xi_{\text{max}}) \times [0,T] \). Then central differences and the rectangle method are used to discretize the differential and integral terms in (24), respectively. Because the coefficients do not depend on \( \xi \) and the integral kernel is shift invariant, the discretization yields a (nonsymmetric) Toeplitz matrix \( T \) having only negative real eigenvalues. We refrain from giving details and refer to the excellent summary given in \([20\text{, Example 3}]\).

In all experiments, we used parameters identical to the ones used in \([20]\): \( \xi_{\text{min}} = -2, \xi_{\text{max}} = 2, K = 100, \nu = 0.25, r = 0.05, \lambda = 0.1, \mu = \text{atL}0.9, \sigma = 0.45 \), as well as a full time step 1.
Figure 2: Experimental comparison of approximations to exp($A_n$) for the Merton model [6.2]

**Left:** Run time comparison for the computation of the full exponential approximation. **Right:** Relative error with respect to expm. The dashed gray line shows $u\|A_n\|_F$, a lower bound for the exponential condition number (times machine precision).

Figure 3: Displacement ranks of the exponentials computed for the Merton model.

Figure 2 (left) shows the run time for the four matrix exponential approximations expm, expmt, sexpm, and sexpmt. As expected the run time of the structure exploiting algorithms grow only quadratically.

We now discuss the accuracy of the obtained approximations. Since the computation of the matrix exponential using variable precision arithmetic is too expensive for the matrix sizes we are considering here, we assess the accuracy of expmt, sexpm, and sexpmt with reference to expm. Figure 2 (right), shows the relative distance

$$\frac{\|B - \text{expm}(T)\|_F}{\|\text{expm}(T)\|_F},$$

where $B$ is the approximation we want to compare. In addition, we show $u\|T\|_F$ (dashed line), which is a lower bound for the exponential condition number (times machine precision). Since the relative error w.r.t. expm is roughly bounded by this quantity, we conclude that our adapted scaling and squaring methods behave forward stable in this example.

In Figure 3 we show the displacement ranks of the approximations to the matrix ex-
Figure 4: Evolution of the displacement ranks in the squaring phase of Alg. 2 for the Merton model discretized with $n = 4096$ points.

For $\text{expmt}$ and $\text{sexpmt}$, this rank corresponds to the length of the generator obtained by the corresponding method after the squaring phase. For $\text{expm}$ and $\text{sexpm}$, the shown rank is the numerical rank of the displacements, as determined by MATLAB’s $\text{rank}$ function. As suggested by the discussion in Section 5, all these ranks are close to each other, and in particular quite small. Finally, Figure 4 shows how the displacement rank evolves during the squaring phase of $\text{expmt}$ ($n = 4096$).

7 Conclusions and future work

We have shown that the full matrix exponential of a Toeplitz matrix can be computed efficiently using scaling and squaring algorithms. A key result that enables this efficient computation is the low displacement rank of rational functions for Toeplitz matrices. Combined with classical results for rational best approximations of the exponential function, it asserts that the Toeplitz matrix exponential itself enjoys low displacement rank if its spectrum is not ill-behaved. By carefully adapting all the matrix computations in the general scaling and squaring framework, we obtain algorithms of quadratic complexity of the input size for computing $\exp(T)$ for a Toeplitz matrix $T$. Since the output size of the matrix exponential is quadratic as well, our algorithms hence achieve optimal complexity.

In this work we have focused on analyzing the displacement rank of polynomials, rational functions and the matrix exponential itself. Two important aspects have received less attention than they probably deserve. One is the design of the scaling and squaring logic itself, for which we relied on the works of Higham [14], as well as G"uttel and Nakatsukasa [10]. An important design goal for these methods is a small number of (unstructured) matrix operations of cubic complexity such as inversion or matrix-matrix multiplication. However, as our careful description in Section 4 shows, minimizing these operations is of much less importance in the Toeplitz case, as long as the overall quadratic complexity is maintained. It would thus be of interest to design a scaling and squaring method specifically for the case of Toeplitz matrices. We also did not attempt to analyze the forward stability in floating point arithmetic for our adapted methods in detail, although such an analysis is certainly of interest.

Finally, our results show that for Toeplitz matrices it is even possible to implement scaling and squaring algorithms of subquadratic complexity, provided that only the generators of $\exp(T)$ are requested, and not the full matrix exponential. Recall that the generators are already sufficient for applying the exponential to a vector. If, for example, the Toeplitz inver-
sions in Alg. 3 are carried out by superfast solvers (e.g., [19, 24, 31]), then these generators can be computed in $O(n \log n)$.

References


Recent publications:

MATHEMATICS INSTITUTE OF COMPUTATIONAL SCIENCE AND ENGINEERING
Section of Mathematics
Ecole Polytechnique Fédérale (EPFL)
CH-1015 Lausanne

11.2016 Andrea Bar.tezzaghi, Luca Dedè, Alfio Quarteroni:
Isogeometric analysis of geometric partial differential equations

12.2016 Erna Begović Kovač, Daniel Kressner:
Structure-preserving low multilinear rank approximation of antisymmetric tensors

13.2016 Diane Guignard, Fabio Nobile, Marco Picasso:
A posteriori error estimation for the steady Navier-Stokes equations in random domains

14.2016 Matthias Boltten, Karsten Kahil, Daniel Kressner, Francisco Macedo, Sonja Sokolovic:
Multigrid methods combined with low-rank approximation for tensor structured Markov chains

15.2016 Nicola Guglielmi, Mutti-Ur Rehman, Daniel Kressner:
A novel iterative method to approximate structured singular values

16.2016 Yvon Maday, Andrea Manzoni, Alfio Quarteroni:
An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated

17.2016 Andrea Manzoni, Luca Ponti:
An adjoint-based method for the numerical approximation of shape optimization problems in presence of fluid-structure interaction

18.2016 Stefano Pagani, Andrea Manzoni, Alfio Quarteroni:
A reduced basis ensemble Kalman filter for state/parameter identification in large-scale nonlinear dynamical systems

19.2016 Andrea Manzoni, Federico Negri:
Automatic reduction of PDEs defined on domains with variable shape

20.2016 Marco Fedele, Elena Faggiano, Luca Dedè, Alfio Quarteroni:
A patient-specific aortic valve model based on moving resistive immersed implicit surfaces

21.2016 Diana Bonomi, Andrea Manzoni, Alfio Quarteroni:
A matrix discrete empirical interpolation method for the efficient model reduction of parametrized nonlinear PDEs: application to nonlinear elasticity problems

22.2016 Jonas Ballani, Daniel Kressner, Michael Peters:
Multilevel tensor approximation of PDEs with random data

23.2016 Daniel Kressner, Robert Luce:
Fast computation of the matrix exponential for a Toeplitz matrix