Matrix Structures and Matrix Functions

Paola Boito
Department of Mathematics, University of Pisa
Pisa, Italy
paola.boito@unipi.it

ABSTRACT

Structured matrices play a relevant role in symbolic and numerical computations. In the literature and in applications we encounter several types of structure, which are typically related to the properties of the problems they stem from: banded structure is often associated with locality of functions or operators, Toeplitz structure arises from shift invariance properties, off-diagonal low-rank structure appears in inverses of banded matrices.

A common trait of most matrix structures is the availability of fast algorithms that perform fundamental operations, such as matrix-vector or matrix-matrix multiplication, solution of linear systems, or eigenvalue computation. For problems of large size, such algorithms are extremely useful both from a symbolic and a numeric point of view, although of course in a numerical setting one needs to pay attention to possible stability issues.

The tutorial will start with a brief overview of matrix structures and of their properties: we are especially interested here in rank structures and in certain forms of sparsity. We will then focus on selected topics concerning the analysis and computation of functions of structured matrices. Functions of matrices have a wide range of applications, for which we will give several examples, from the solution of differential equations to network analysis. Think for instance of the matrix exponential exp(A): it appears in the solution of a multidimensional Cauchy problem with coefficient matrix A, but it also has a combinatorial meaning when A is the adjacency matrix of a graph. If the quantity of interest is the action of a matrix function on a vector, moreover, one can often bypass the explicit construction of the matrix function itself and devise a more efficient approach.

Most methods for the computation of matrix functions are ultimately based on polynomial or rational approximation, which are either applied in explicit form, or through iterative methods such as Lanczos or Arnoldi, often in combination with suitable pre- and post-processing steps. If matrix A is structured, it is natural to ask how to tailor such methods to the specific structure of A. We will present a few possible answers to this question and demonstrate their advantages through several practical examples.

ISSAC 2023, July 24–27, 2023, Tromsø, Norway

ACM ISBN 979-8-4007-0039-2/23/07…$15.00
https://doi.org/10.1145/3597066.3597148

CCS CONCEPTS

• **Mathematics of computing** → **Computations on matrices**; *Mathematical analysis.*

KEYWORDS

structured matrices, matrix functions

ACM Reference Format:
Paola Boito. 2023. Matrix Structures and Matrix Functions. In *International Symposium on Symbolic and Algebraic Computation 2023 (ISSAC 2023), July 24–27, 2023, Tromsø, Norway*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3597066.3597148

1 INTRODUCTION

Matrices that arise from a large range of problems in mathematics, physics or engineering typically display a characteristic structure, such as sparsity patterns or a rank structure. Exploiting this structure is the key to the design of more efficient algorithms and is especially useful for medium-to-large scale problems, where algorithms with low computational cost play a crucial role.

Matrix structures that have been investigated in the literature include for instance:

- rank structure (e.g., quasiseparable matrices),
- displacement structure (e.g., Toeplitz or Hankel matrices)
- sparsity (e.g., banded matrices).

Here we are interested in *data-sparse* structure, that is, structure that allows us to represent an $n \times n$ matrix using only $O(n)$ parameters. This is because our ultimate goal is the design of linear-time algorithms, for which a necessary condition is, obviously, having to deal with at most a linear number of variables. So, while, say, symmetric or unitary matrices certainly display a kind of structure, we will not focus on such structures (unless they arise in combination with a data-sparse form).

Sparsity – i.e., $n \times n$ matrices having at most $O(n)$ nonzero entries – is an obvious example of data-sparse structure. In some cases, sparsity follows a predetermined pattern, as in banded matrices. In other cases, there is no or little *a priori* information on the position of the nonzero entries: an example is given by adjacency matrices of real-life networks. Even without knowing where the nonzeros are, though, we can assume that matrix-vector products can be computed cheaply, which is why Krylov-type methods are very popular for sparse matrices.

However, data sparsity may very well appear in dense matrices. One example is provided by quasiseparable matrices, a class of (generally dense) rank-structured matrices where off-diagonal blocks have low rank. This rank structure is related to sparsity, because banded matrices and their inverses happen to be quasiseparable.

Further desirable properties in a class of structured matrices include being closed with respect to basic operations such as sum,

product or inversion. The class of quasiseparable matrices, for instance, is closed under inversion, and "almost closed" under sums and products (i.e., sums and products of quasiseparable matrices are still quasiseparable, but may have a higher quasiseparable rank). Similar properties hold for displacement-structured matrices.

One may also ask whether such matrix classes are closed not only under inversion, but with respect to more general matrix functions. When the matrix structure under study is not preserved by the application of a certain function, it may nevertheless happen that the structure is preserved in an approximate sense. A precise characterization of this occurrence offers more insight into the structure and at the same time provides a useful tool for numerical approximation.

When dealing with functions of matrices, one is typically interested in one or more among the following tasks:

- compute/approximate the whole matrix function $f(A)$,
- compute/approximate the action of the matrix function on a vector, that is, $f(A)\,v$,
- compute/approximate a bilinear function of $f(A)$, that is, the quantity $w^H f(A)\,v$; for suitable choices of vectors v and w, this could be for instance a matrix element of $f(A)$, or a row/column sum of elements of $f(A)$,
- if A is structured, give an *a priori* characterization of the (exact or approximate) structure of $f(A)$. Results in this direction are available in particular for banded/sparse [4, 5, 12], Toeplitz [1, 6, 7, 24] and quasiseparable/hierarchical matrices [18, 27].

Computing matrix functions can be computationally expensive; it is therefore interesting and often crucial to design fast algorithms that are tailored to the structure of A, if present. The literature on this topic is very rich and giving a comprehensive overview is outside the scope of this tutorial. We will choose instead to focus on a few specific examples, which can nevertheless offer insight into more general and versatile techniques.

2 BACKGROUND

2.1 Matrix structure

We recall here two main classes of structured matrices: banded and quasiseparable matrices.

Definition 2.1. A matrix $A \in \mathbb{C}^{n\times n}$ is said to be banded with bandwidth r if for any pair of indices i, j with $|i-j| > r$ it holds $A_{ij} = 0$.

Broadly speaking, banded matrices typically arise from locality properties, where each object in the system under study interacts only with its close neighbors. Well-known examples include for instance the finite-difference discretization of a one-dimensional Laplace operator with zero boundary conditions, or matrices arising from polynomial recurrence relations (e.g., Jacobi matrices for orthogonal polynomials).

For the definition of quasiseparable matrices we follow [14]. A comprehensive analysis can be found in [15] and references therein.

Definition 2.2. Let $r_L, r_U \in \mathbb{N}$. A matrix $M \in \mathbb{C}^{n\times n}$ is (r_L, r_U)-quasiseparable if

- $\max_{1\le k\le n-1} \operatorname{rank} M(k+1:n, 1:k) \le r_L$,
- $\max_{1\le k\le n-1} \operatorname{rank} M(1:k, k+1:n) \le r_U$.

The numbers r_L and r_U are known as the lower (resp. upper) quasiseparability order of M. Obviously, this kind of structure is especially interesting when r_L and/or r_U are much smaller than n, which is what is meant here when saying that a matrix is quasiseparable.

A crucial property of the quasiseparable class is closure w.r.t. inversion, sum and product.

Proposition 2.3. *Let $M \in \mathbb{C}^{n\times n}$ be an invertible quasiseparable matrix of order (r_L, r_U). Then M^{-1} is also quasiseparable of order (r_L, r_U).*

Let $M_1, M_2 \in \mathbb{C}^{n\times n}$ be quasiseparable matrices of orders $(r_L^{(1)}, r_U^{(1)})$ and $(r_L^{(2)}, r_U^{(2)})$. Then the matrices $M_1 + M_2$ and $M_1 \cdot M_2$ are quasiseparable of orders at most $(r_L^{(1)} + r_L^{(2)}, r_U^{(1)} + r_U^{(2)})$.

Clearly, r-banded matrices are (r, r)-quasiseparable; from Proposition 2.3 it also follows that their inverses, if they exist, are (r, r)-quasiseparable.

Different representations of quasiseparable structure are available in the literature. We recall a representation in scalar form from [15].

Let $M \in \mathbb{C}^{n\times n}$ be (r_L, r_U)-quasiseparable. Then its entries can be represented in the following way:

$$M_{ij} = \begin{cases} p(i)a_{ij}^{>}q(j), & 1 \le j < i \le n, \\ d(i), & 1 \le i = j \le n, \\ g(i)b_{ij}^{<}h(j), & 1 \le i < j \le n \end{cases} \tag{1}$$

where:

- $p(2), \dots, p(N)$ are row vectors of length r^L, $q(1), \dots, q(n-1)$ are column vectors of length r^L, and $a(2), \dots, a(n-1)$ are matrices of size $r^L \times r^L$; these are called *lower quasiseparable generators* of order r^L;
- $d(1), \dots, d(n)$ are numbers (the diagonal entries),
- $g(2), \dots, g(n)$ are row vectors of length r^U, $h(1), \dots, h(n-1)$ are column vectors of length r^U, and $b(2), \dots, b(n-1)$ are matrices of size $r^U \times r^U$; these are called *upper quasiseparable generators* of order r^U;
- the matrices $a_{ij}^{>}$ and $b_{ij}^{<}$ are defined as

$$\begin{cases} a_{ij}^{>} = a(i-1)\cdots a(j+1) \text{ for } i > j+1; \\ a_{j+1,j}^{>} = 1 \end{cases}$$

and

$$\begin{cases} b_{ij}^{<} = b(i+1)\cdots b(j-1) \text{ for } j > i+1; \\ b_{i,i+1}^{<} = 1. \end{cases}$$

In other words, we represent each off-diagonal entry of M as a product of "building blocks" picked from a set of $O(n)$ items (generators): a row vector times the product of a few matrices, times a column vector. Block versions of this representation are available as well.

Note that quasiseparable structure has also been investigated and used by Pernet and Storjohann in relation to the rank profile matrix [29].

Generalized notions of matrix rank structure have been explored by various authors; we mention the study of hierarchical matrices

proposed by Hackbusch [20] and the analysis and implementation of HSS and HODLR formats presented in [28].

For our purposes, let us recall that in the quasiseparable case both matrix-vector multiplication and matrix inversion (i.e., solution of a linear system) can be performed in $O(n)$ arithmetic operations. In particular, structured matrix inversion relies on LU or QR factorizations, since the triangular and orthogonal factors are rank-structured as well. Typically, algorithms for solving $Ax = b$ with quasiseparable A implicitly compute $x = U^{-1}L^{-1}b$ or $x = R^{-1}U^Tb$ in terms of quasiseparable generators.

2.2 Functions of matrices

The matrix inverse $A \to A^{-1}$ is likely the first example of matrix function that everyone has encountered in their linear algebra courses. Another well-known example is the matrix exponential, usually introduced through its Taylor series expansion

$$e^A = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \ldots = \sum_{k=0}^{\infty} \frac{1}{k!}A^k.$$

More generally, if A is a square matrix and $f(z)$ a sufficiently regular scalar function, we can define the matrix function $f(A)$. Several different (but essentially equivalent) definitions are possible; see [22] for a classical reference.

Definition 2.4. Let $A \in \mathbb{C}^{n\times n}$ be a diagonalizable matrix with eigenvalue decomposition

$$A = M\Lambda M^{-1}, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of A. Suppose that the function $f(z)$ is well defined on the spectrum of A. Then the matrix function $f(A)$ is defined as

$$f(A) = Mf(\Lambda)M^{-1},$$

where $f(\Lambda) = \mathrm{diag}(f(\lambda_1), f(\lambda_2), \ldots, f(\lambda_n))$.

This definition can be extended to non-diagonalizable matrices by using the Jordan canonical form. Because of computational cost and possible stability issues, it is not often used for the actual computation of $f(A)$. It is however useful for theoretical analysis, and it has the merit of highlighting how the definition of $f(A)$ essentially hinges on the values of f – and possibly of its derivatives – on the spectrum of A. This same property is clear from the following definition, based on the interpolating polynomial.

Definition 2.5. Suppose that $A \in \mathbb{C}^{n\times n}$ has distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and let $p(z)$ be the polynomial of degree at most $n-1$ that interpolates $f(z)$ on these eigenvalues. Then $f(A)$ is defined as

$$f(A) = p(A).$$

This idea generalizes immediately to the case of multiple eigenvalues via Hermite interpolation.

Yet another definition of $f(A)$ can be given through a matrix analogue of the Cauchy integral.

Definition 2.6 (Dunford-Cauchy integral). Let $A \in \mathbb{C}^{n\times n}$ and let $f(z)$ be analytic on and inside a closed contour Γ that encloses the spectrum of A. Then $f(A)$ is defined as

$$f(A) = \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A)^{-1}dz. \tag{2}$$

3 PROBLEMS AND APPLICATIONS

3.1 Network analysis

Matrix functions – particularly the resolvent and the exponential function – have applications to the analysis of complex networks [2, 16]. Networks are typically modeled by graphs, and functions of the associated adjacency or Laplacian matrices yield useful information on centrality and communicability properties of the corresponding graph.

The general gist is as follows. Recall that the adjacency matrix of a simple graph with nodes labeled from 1 to n is the $n \times n$ matrix A such that $A_{i,j} = 1$ if there is an edge from node i to node j, and $A_{i,j} = 0$ otherwise. Also recall that, for any positive integer k, the (i, j) entry of A^k counts the number of walks of length k that start from node i and arrive at node j. If a function $f(z)$ has a Taylor series expansion $f(z) = \sum_{k=0}^{\infty} a_k(z - z_0)^k$ that converges in a disc containing the spectrum of A, then each entry

$$[f(A)]_{ij} = \sum_{k=0}^{\infty} a_k[(A - z_0I_n)^k]_{ij}$$

is a weighted sum of the number of walks from node i to node j, where the weights for each walk length are given by the Taylor coefficients of $f(z)$. The (weighted) number of walks that begin and end at the same node can be taken as a measure of the importance (centrality) of that node; similarly, the weighted number of walks from node i to node j tells us how "well-connected" nodes i and j are. This motivates the following definitions [17].

Definition 3.1. With the notation and hypotheses introduced above, define:

- the f-subgraph centrality of node i as $[f(A)]_{ii}$,
- the f-subgraph communicability between nodes i and j as $[f(A)]_{ij}$.

The most popular choices for f are the exponential function $f(A) = e^A$ and the resolvent function $f(A) = (I_n - \alpha A)^{-1}$, where the real parameter α is such that $0 < \alpha < \rho(A)$, with $\rho(A)$ the spectral radius of A.

Adjacency matrices for realistic networks are typically sparse. Note that the sparsity pattern of A also depends on the labeling of the nodes; re-labeling the nodes does not fundamentally change the problem under study, but yields a new adjacency matrix $A' = PAP^T$, where P is a permutation matrix that encodes the permutation of node labels. In some cases it may be useful to choose P so that the sparsity pattern of PAP^T is more amenable to structured computations. For instance, a reordering algorithm such as Cuthill-McKee [10] may help bring A to banded form.

Examples of adjacency matrices for different graphs are shown in Figure 1.

3.2 Differential equations and exponential integrators

The role of the matrix exponential in the solution of linear systems of ODE is well known: the Cauchy problem

$$\frac{d}{dt}y(t) = Ay(t), \qquad y(t_0) = y_0,$$

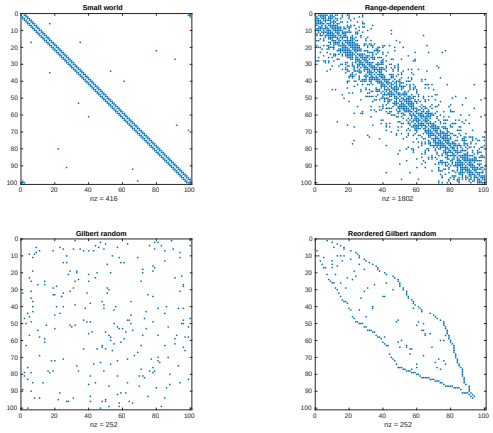

Figure 1: Examples of sparsity patterns for adjacency matrices. Top left: small-world graph; the banded plus low-rank structure is visible. Top right: range-dependent random graph; note the banded structure. Bottom: Gilbert random graph and its reordered form (using reverse Cuthill-McKee).

with A a constant matrix, admits the solution $y(t) = e^{At}y_0$. More generally, exponential integrators [23] require the computation – and possibly the inversion – of ϕ-functions applied to a matrix. The ϕ-functions are defined recursively as

$$\phi_0(z) = e^z,$$
$$\phi_{k+1}(z) = \frac{\phi_k(z) - (1/k!)}{z}, \qquad k = 0, 1, 2, \ldots \tag{3}$$

These functions also play a role in differential problems where the initial conditions are replaced by boundary or integral conditions.

In many cases, the matrix A is structured: for instance, if we start from a heat equation $\frac{\partial}{dt}u(x,t) = \gamma\frac{\partial^2}{\partial x^2}u(x,t)$ and perform a partial discretization of the spatial derivative using, say, finite differences, then we obtain a system of ODE where the matrix A is the classical discretization of the one-dimensional Laplacian, and therefore banded.

As an example of differential equation with nonlocal boundary conditions, consider the following problem in $\mathbb{R}^n$:

$$\frac{dv(t)}{dt} = Av(t), \quad 0 < t < 2\pi, \tag{4}$$
$$\frac{1}{2\pi}\int_0^{2\pi} v(t)\, dt = g, \tag{5}$$

where A is a linear operator in $\mathbb{R}^n$ and $g \in \mathbb{R}^n$ is a given vector. Suppose that $\mu_k = ik$, $k = \pm1, \pm2, \pm3, \ldots$ are regular points of A. Then the (unique) solution is

$$v(t) = q_t(A)g, \quad q_t(z) = \frac{2\pi z e^{zt}}{e^{2\pi z} - 1},$$

that is,

$$v(t) = 2\pi\, A\, e^{At}\, (e^{2\pi A} - I)^{-1} g,$$

and the function $q_t(z)$ is, up to scaling, the reciprocal of the function $\phi_1(z)$ defined in (3). In a similar vein, see also [19] for recent work on inversion of the ϕ_2 function and its application to differential problems.

3.3 Electronic structure computation

Linear scaling methods in electronic structure computations rely on efficient approximation of spectral projectors associated with sparse (often banded) matrices arising from a discretization of the Hamiltonian operator associated with the physical system. In the language of matrix functions, this means computing $h(H)$, where $h(x)$ is a Heaviside step function and H is a Hermitian banded matrix [3]. One can choose to compute instead $f_{FD}(H)$, where f_{FD} is a Fermi-Dirac function of the form $f_{FD}(z) = (1 + e^{\beta(z-\mu)})^{-1}$ and coincides with $h(z)$ when restricted to the spectrum of H.

In this application, *a priori* bounds on the off-diagonal decay of elements of $h(H)$ as in Theorem 4.1 hold a deep physical meaning, as (possibly approximate) bandedness is related to the physical notion of locality.

Note that this case is especially tricky because the Heaviside function has a point of discontinuity, which raises difficulties for the polynomial approximation techniques on which relies the proof of Theorem 4.1.

4 COMPUTATIONAL TECHNIQUES

The range of techniques for computing or approximating $f(A)$ is very large. General-purpose methods may rely on the definitions given in section 2. For instance, the Maple command `MatrixFunction` uses the interpolating polynomial. The MATLAB function `funm`, on the other hand, is based on the Schur-Parlett algorithm [11], whereas Mathematica's `MatrixFunction` combines Schur-Parlett and diagonalization/Jordan form depending on whether A is inexact or symbolic. When possible, however, it is preferable to employ approaches that are specifically tailored to the function $f(z)$ and to the structure of A, if present. One idea is to adapt a suitable standard method to the structure of A, exploiting fast implementations of basic operations such as addition, multiplication and inversion. Examples include:

- Rational (e.g., Padé) approximation

  $$f(A) \approx p(A)q(A)^{-1}, \qquad p(z), q(z) \text{ polynomials.}$$

  A well-known application is the scaling-and-squaring method for the matrix exponential, which relies on a Padé approximation of e^z. The polynomial evaluation and inversion operations can be implemented in structured fashion; see for instance the `expm` function in the MATLAB toolbox `hm-toolbox` for HODLR matrices.
- Rational iterations. Example: the Newton iteration for the matrix sign function

  $$A_{k+1} = \frac{1}{2}(A_k + A_k^{-1}).$$

- Krylov methods for approximating $f(A)v$. See in particular [9] for recent advances in this direction for banded and rank-structured matrices.

As long as the method relies on matrix inversion and on a small number of matrix sums or products, one can compute $f(A)$ in an efficient way.

Another approach to the approximate computation of $f(A)v$ uses discretization of the Dunford-Cauchy integral (2):

$$f(A)v = \frac{1}{2\pi}\int_\Gamma f(z)(zI-A)^{-1}v\,dz \approx \sum_k w_k f(z_k)(z_kI-A)^{-1}v,$$

with suitably chosen integration nodes $\{z_k\}_k \subset \Gamma$ and weights $\{w_k\}_k$. This approach has the merit of reformulating the problem as the solution of several diagonally shifted linear systems, which are typically structured if A is structured and therefore allow for fast solutions. More precisely, not only each linear system is structured, but in some cases we can also improve on the idea of solving each system separately, and devise algorithms that solve the whole set of shifted linear systems more efficiently than repeatedly appying a structured solver to each system.

Similar considerations hold if $f(z)$ is approximated via partial fraction decomposition.

4.1 Approximate structure preservation

A natural question when defining or computing $f(A)$ with A structured is: will $f(A)$ retain the same structure as A?

This is true in some particular cases, such as computing the inverse of a quasiseparable matrix. But, for more general functions, the answer is no: for instance, the exponential of a banded matrix is typically full (non-banded). However, it may still be true that $f(A)$ is structured in an approximate sense: in the previous example, the exponential of a banded matrix often displays a fast off-diagonal decay behavior, so it is close to being banded. In fact, the following result holds [4]:

Theorem 4.1. *Let A be a real symmetric m-banded or sparse matrix and let $f(x)$ be a smooth function on an interval containing the spectrum of A. Then there are constants $C > 0$ and $0 < \lambda < 1$ such that*

$$|[f(A)]_{ij}| \le C\lambda^{\frac{|i-j|}{m}} \tag{6}$$

(applicable if A is m-banded) and

$$|[f(A)]_{ij}| \le C\lambda^{d(i,j)}, \tag{7}$$

where $d(i,j)$ is the graph distance between nodes i and j.

Note that this is not merely an existence result: the constants C and λ can be easily computed explicitly. Such results have their basis in work from the 1980s on inverses of banded matrices [12, 13] and have undergone further developments until very recently [5].

Besides being of theoretical interest, these *a priori* bounds have practical applications. For example, we could use them to decide in advance which pairs of nodes in a graph are badly connected, that is, their f-subgraph communicability is below a given threshold.

Similar, although more involved, results are available for quasiseparable structure. In [27], sufficient conditions are provided ensuring that, if $f(z)$ is holomorphic in a disc in the complex plane and A is quasiseparable of order k, it holds

$$\sigma_\ell(C) \le \gamma e^{-\frac{\alpha\ell}{k}}, \tag{8}$$

where C is a generic off-diagonal block in $f(A)$, and σ_ℓ denotes the ℓ-th singular value. The quantities $\gamma, \alpha > 0$ depend on f and on the spectral properties of A. Clearly (8) implies the existence of good quasiseparable approximants for $f(A)$. See also [26] for similar results on the structure of solutions of Lyapunov matrix equations with quasiseparable coefficients.

5 EXAMPLES

Let us present a few examples of the ideas outlined above.

5.1 Bounds on communicability for small-world networks

Our toy model here is a 200 nodes small-world network, that is, a graph where each node j is connected to nodes $j-1$ and $j+1$ (mod 200) and a few extra connections are added randomly; see top-left picture in Figure 1 for a 100-node example. We would like to bound exponential subgraph communicabilities, so as to be able to say in advance which pairs of nodes are surely badly connected. The adjacency matrix A of our graph is sparse, but not banded, although it can be brought to banded form via permutations (i.e., node relabeling).

Let us use the results of Theorem 4.1. We can do that in two ways: either apply (7) directly to A, or reorder A to banded form and apply (6). The first strategy is more accurate, but also more expensive; see Figure 2.

5.2 Computation of matrix square root

The Dunford-Cauchy integral formulation for $f(A)$ has the merit of writing $f(A)$ in terms of a contour integral of an expression involving $(zI-A)^{-1}$. If A is structured, the inverse of $zI-A$ can often be computed cheaply, and the contour integral can be discretized via a suitable quadrature formula. For specific functions, a combination of this idea with a clever choice of a conformal map in the complex plane may yield a discrete approximation of $f(A)$ that converges remarkably quickly.

In particular, the matrix square root of a real matrix A with real positive spectrum can be rewritten as [21]

$$A^{1/2} = \frac{iA}{\pi}\int_{-i\infty}^{i\infty}(w^2I-A)^{-1}dw.$$

Its discretization via a suitable node grid on the imaginary axis converges geometrically; note however that the formula requires the evaluation of elliptic functions.

Suppose we want to approximate $A^{1/2}b$ for a given structured matrix $A \in \mathbb{R}^{n\times n}$ and a given vector $b \in \mathbb{R}^n$. The discussion above implies that the problem reduces to solving a few linear systems of the form $(w^2I-A)v = b$, for different values of the parameter w.

If A is quasiseparable, the following two properties hold:

- any diagonal shift of A is still quasiseparable, therefore solving $(w^2I-A)v = b$ costs $O(n)$ arithmetic operations,
- the QR factorization of w^2I-A takes the form

  $$w^2I - A = VU_wR_w,$$

  with V, U_w orthogonal and R_w upper triangular; note in particular that the factor V is the same for all shifted systems. This remark does not change the asymptotic computional cost, but it allows us nevertheless to halve the constant hidden in the "big O" notation, and therefore the overall computation time.

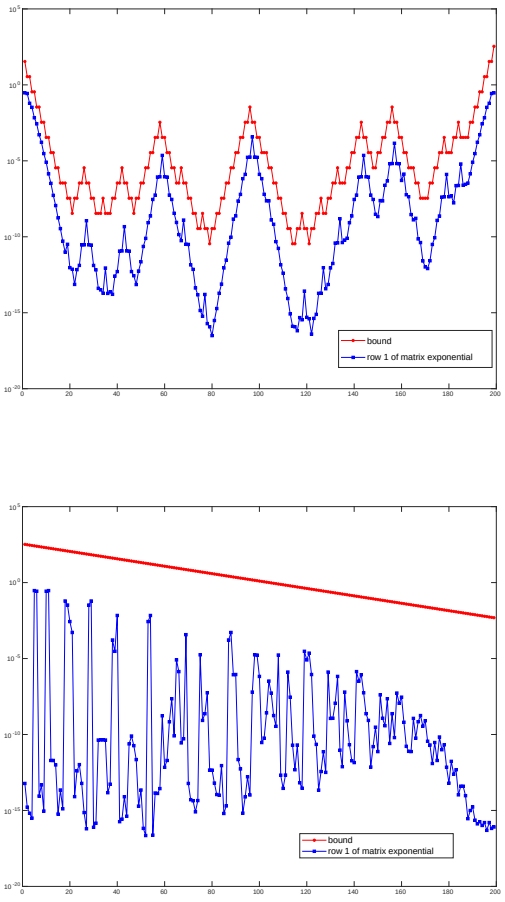

Figure 2: Decay bounds for e^A, with A the adjacency matrix of a small-world graph. Left: plot of first row of e^A and of bounds based on graph distance. Right: first row of the exponential of reordered A and bound based on band structure.

Consider for instance a matrix $A \in \mathbb{R}^{2500\times 2500}$ arising from a finite-difference discretization of the differential operator $-\Delta u(x,y) + 5\frac{\partial u}{\partial x}u(x,y)$ on the unit square with zero boundary conditions. Let us approximate $A^{1/2}b$, where b is taken as the vector of all ones, using the approximation outlined above in combination with a structured solver of quasiseparable linear systems. The convergence of the approximation is shown in Table 1.

ℓ	relative error
6	1.49e−5
7	2.48e−6
8	3.03e−7
9	4.16e−8
10	3.97e−9
11	5.80e−10
12	9.13e−11
13	1.26e−11
14	1.27e−12
15	2.18e−13

Table 1: Relative approximation errors for $A^{1/2}b$ using contour integral discretization and structured solution of linear systems. The parameter ℓ denotes the number of rational terms (i.e., number of quadrature nodes) used in the approximation.

5.3 Solution of nonlocal differential problems

As mentioned in Section 3.2, the solution to the differential problem (4)-(5) can be expressed via the action of $\psi_1(A) = \phi_1^{-1}(A)$ on vector g. The function $\psi_1(A)$ can be expanded as [8]

$$\psi_{n,s}(A) = p_n(A) + 2(-1)^n \left(\sum_{k=1}^{s} \frac{1}{k^{2n}} \left(\left(\frac{A}{2\pi} \right)^2 + k^2 I \right)^{-1} \right) \left(\frac{A}{2\pi} \right)^{2(n+1)},$$

where

$$p_n(A) = I - \frac{1}{2}A + \sum_{i=0}^{n-1} A^{2(i+1)} \frac{B_{2(i+1)}}{(2(i+1))!},$$

and B_k is the k-th Bernoulli number. This is a family of approximation formulas formed by a polynomial term plus a rational part resembling a partial fraction decomposition. Application of such expressions to a vector requires only matrix-vector products and the solution of diagonally shifted linear systems. If A is quasiseparable, the computational remarks of section 5.2 still apply.

6 CONCLUSION AND PERSPECTIVES

We have outlined a few examples of how functions of structured matrices can be analyzed, computed and/or approximated through the properties of matrix structure. This is a very active research area; further research directions include:

- the development of "structure-friendly" formulations for more matrix functions of interest;
- the development of "structure-friendly" formulations for more structured classes (e.g., displacement-structured matrices), and their interplay with band or off-diagonal rank structure;
- extension from finite-size matrices to more general operators and (semi)infinite matrices;
- development of ad-hoc software over various platforms.

Finally, let us mention that the recent surge of interest in randomized methods for numerical linear algebra also involves rank structure; see e.g., [25] and references therein.

REFERENCES

[1] Bernhard Beckermann, Joanna Bisch, and Robert Luce. 2022. On the rational approximation of Markov functions, with applications to the computation of Markov functions of Toeplitz matrices. *Numerical Algorithms* 91, 1 (2022), 109–144.
[2] Michele Benzi and Paola Boito. 2020. Matrix functions in network analysis. *GAMM-Mitteilungen* 43, 3 (2020), e202000012.
[3] Michele Benzi, Paola Boito, and Nader Razouk. 2013. Decay properties of spectral projectors with applications to electronic structure. *SIAM review* 55, 1 (2013), 3–64.

[4] Michele Benzi and Nader Razouk. 2007. Decay bounds and O (n) algorithms for approximating functions of sparse matrices. *Electron. Trans. Numer. Anal* 28, 16-39 (2007), 08.

[5] Michele Benzi and Michele Rinelli. 2022. Refined decay bounds on the entries of spectral projectors associated with sparse Hermitian matrices. *Linear Algebra Appl.* 647 (2022), 1–30.

[6] Dario A. Bini, Stefano Massei, and Beatrice Meini. 2017. On functions of quasi-Toeplitz matrices. *Sbornik: Mathematics* 208, 11 (2017), 1628.

[7] Dario A Bini and Beatrice Meini. 2019. On the exponential of semi-infinite quasi-Toeplitz matrices. *Numer. Math.* 141 (2019), 319–351.

[8] Paola Boito, Yuli Eidelman, and Luca Gemignani. 2022. Computing the reciprocal of a ϕ-function by rational approximation. *Advances in Computational Mathematics* 48 (2022), 1–28.

[9] Alice Cortinovis, Daniel Kressner, and Stefano Massei. 2022. Divide-and-Conquer Methods for Functions of Matrices with Banded or Hierarchical Low-Rank Structure. *SIAM J. Matrix Anal. Appl.* 43, 1 (2022), 151–177.

[10] Elizabeth Cuthill and James McKee. 1969. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*. 157–172.

[11] Philip I Davies and Nicholas J Higham. 2003. A Schur-Parlett algorithm for computing matrix functions. *SIAM J. Matrix Anal. Appl.* 25, 2 (2003), 464–485.

[12] Jean-Pierre Dedieu. 1988. Matrix homographic iterations and bounds for the inverses of certain band matrices. *Linear Algebra Appl.* 111 (1988), 29–42.

[13] Stephen Demko, William F Moss, and Philip W Smith. 1984. Decay rates for inverses of band matrices. *Mathematics of computation* 43, 168 (1984), 491–499.

[14] Yuli Eidelman and Israel Gohberg. 1999. On a new class of structured matrices. *Integral Equations and Operator Theory* 34, 3 (1999), 293–324.

[15] Yuli Eidelman, Israel Gohberg, and Iulian Haimovici. 2014. *Separable type representations of matrices and fast algorithms*. Springer.

[16] Ernesto Estrada and Desmond J Higham. 2010. Network properties revealed through matrix functions. *SIAM review* 52, 4 (2010), 696–714.

[17] Ernesto Estrada and Juan A Rodriguez-Velazquez. 2005. Subgraph centrality in complex networks. *Physical Review E* 71, 5 (2005), 056103.

[18] Ivan Gavrilyuk, Wolfgang Hackbusch, and Boris Khoromskij. 2005. Data-sparse approximation to a class of operator-valued functions. *Mathematics of computation* 74, 250 (2005), 681–708.

[19] Luca Gemignani. 2023. Efficient Inversion of Matrix ϕ-Functions of Low Order. *arXiv preprint arXiv:2302.07565* (2023).

[20] Wolfgang Hackbusch. 2015. *Hierarchical matrices: algorithms and analysis*. Vol. 49. Springer.

[21] Nicholas Hale, Nicholas J Higham, and Lloyd N Trefethen. 2008. Computing A^α,$\log(A)$, and related matrix functions by contour integrals. *SIAM J. Numer. Anal.* 46, 5 (2008), 2505–2523.

[22] Nicholas J Higham. 2008. *Functions of matrices: theory and computation*. SIAM.

[23] Marlis Hochbruck and Alexander Ostermann. 2010. Exponential integrators. *Acta Numerica* 19 (2010), 209–286.

[24] Daniel Kressner and Robert Luce. 2018. Fast computation of the matrix exponential for a Toeplitz matrix. *SIAM J. Matrix Anal. Appl.* 39, 1 (2018), 23–47.

[25] James Levitt and Per-Gunnar Martinsson. 2022. Linear-complexity black-box randomized compression of hierarchically block separable matrices. *arXiv preprint arXiv:2205.02990* (2022).

[26] Stefano Massei, Davide Palitta, and Leonardo Robol. 2018. Solving rank-structured Sylvester and Lyapunov equations. *SIAM journal on matrix analysis and applications* 39, 4 (2018), 1564–1590.

[27] Stefano Massei and Leonardo Robol. 2017. Decay bounds for the numerical quasiseparable preservation in matrix functions. *Linear Algebra Appl.* 516 (2017), 212–242.

[28] Stefano Massei, Leonardo Robol, and Daniel Kressner. 2020. hm-toolbox: Matlab software for HODLR and HSS matrices. *SIAM Journal on Scientific Computing* 42, 2 (2020), C43–C68.

[29] Clément Pernet and Arne Storjohann. 2018. Time and space efficient generators for quasiseparable matrices. *Journal of Symbolic Computation* 85 (2018), 224–246.