

Preface

Usually one avoids numerical algorithms involving operations with large, fully populated matrices. Instead one tries to reduce all algorithms to matrix-vector multiplications involving only sparse matrices. The reason is the large number of floating point operations; e.g., $\mathcal{O}(n^3)$ for multiplying two general $n \times n$ matrices. Starting with Strassen's algorithm [236], one has tried to reduce the work to $\mathcal{O}(n^\gamma)$ with $\gamma < 3$. However, these attempts cannot be satisfactory since $\gamma \geq 2$ is a lower bound, and even quadratic work is unacceptable for large-scale matrices.

The hierarchical matrix (\mathcal{H} -matrix) technique provides tools for performing matrix operations in almost linear work $\mathcal{O}(n \log^* n)$. This is no contradiction to the previously mentioned lower bound, since the former statement holds for *exact* computations, whereas \mathcal{H} -matrix operations yield approximations. The approximation errors are nevertheless acceptable, since large-scale matrices are usually obtained from discretisations which anyway contain a discretisation error. The operations enabled by the \mathcal{H} -matrix technique are not only matrix addition and multiplication, but also matrix inversion and the LU or Cholesky decomposition. The positive statements above do not hold for all matrices, but they are valid for the important class of matrices originating from standard discretisations of elliptic partial differential equations or related integral equations.

The treatment of systems of linear equations by the \mathcal{H} -matrix technique can be classified between a direct method and an iterative method. On the one hand, the approximate inverse or LU decomposition can be determined with freely selectable accuracy. This allows for direct solution of a linear system with the corresponding accuracy. On the other hand, an inverse or LU decomposition with moderate accuracy is sufficient for constructing a fast iteration since a good approximation of the inverse is also a perfect preconditioner. Hence, the hierarchical matrix technique allows us to construct preconditioners in a black-box fashion.

Having all matrix operations available, a much larger class of problems can be treated than by restricting to matrix-vector multiplications. For instance, computing matrix-valued functions is possible (e.g., the matrix exponential function) as well as solving matrix equations (e.g., the Riccati equation).

Approximate performance of the operations can be successful only if the work for reaching an accuracy ε increases only weakly as $\varepsilon \rightarrow 0$. For the inverse of the matrices discretising elliptic boundary value problems and for matrices corresponding to boundary integral equations it will be shown that the work depends only logarithmically on ε . For general large matrices such a statement is wrong; i.e., the \mathcal{H} -matrix technique cannot be applied to *all* matrices. Nevertheless, numerical tests show a very robust behaviour. Good performance is observed in many cases, for which no analytical justification is available. Another important feature for the practical performance is the fact that the \mathcal{H} -matrix technique is of black-box type.

The \mathcal{H} -matrix technique is based on three different components. (a) The *first, analytical component* is the local, separable approximation of the Green function or, respectively, of the kernel function of an integral operator. In the past, different versions of such techniques have been applied to discrete integral operators: panel-clustering [149], multipole expansions [115], or matrix compression by wavelets [76]. Using these techniques, matrix-vector multiplication by a fully populated matrix can be performed in almost linear work. (b) The *second component* belongs to linear algebra. Singular value decompositions and QR decompositions play an important role for organising the local matrix data. (c) The *third component* concerns discrete structures. The tools from (a) and (b) are applied to submatrices. The suitable partition of a matrix into submatrices of optimal size is a crucial step for both data compression and the ability to perform matrix operations. Discrete structures are characterised by two trees, the cluster tree and the block cluster tree.

The aim of this monograph is the comprehensive introduction into the technique of hierarchical matrices. Since this technique is developed in particular for fully populated large-scale matrices from the field of boundary value problems, we shortly discuss the boundary integral method and discretisation of boundary value problems. According to the different components mentioned above, the first chapter is concerned with different items from analysis, linear algebra, and the structures forming the basis of the algorithms. To avoid technical details in the first parts, this book contains five appendices containing detailed information about the required background.

The last chapter shows a connection to the numerical treatment of large-scale tensors. Here we only mention those tensor techniques that are directly connected with hierarchical matrices. On the other hand, the idea of low-rank approximations can be applied very successfully to many tensor-structured problems involving multivariate functions, grid functions in high-dimensional spatial tensor grids, and the corresponding matrices (cf. Hackbusch [132]). We emphasise an essential difference between the low-rank techniques used for \mathcal{H} -matrices and tensors. In the case of \mathcal{H} -matrices, low-rank approximation is applied to suitable submatrices exhibiting a fast decay of their singular values, whereas the tensor approximations are global ones.

The book contains several hints concerning implementational details. A concrete description of the algorithms (in the computer language C) can be found in the Lecture Notes [55]. Concrete numerical examples and comparisons can also be found in Börm [48] and in many of the cited articles.

The author developed the technique of hierarchical matrices at the end of the nineties. The first article [121] appeared in 1999. An essential step was the efficient implementation of the method in the dissertation [102] of L. Grasedyck, which was defended in 2001. The first German version of this work in 2009 is based on manuscripts of lectures at the University of Leipzig (summer semester 2004 and winter semester 2006/7) and at the Christian-Albrechts-Universität Kiel (summer semester 2004). The material has been essentially enriched by the contributions of S. Börm, M. Bebendorf, R. Kriemann, B. Khoromskij, and S. Le Borne. The implementation of the \mathcal{H} -matrix technique has led to the software package \mathcal{H} -Lib^{pro} (see [175, 179, 180]).

The present book is not only a translation of the German edition, but is a revised and extended version. For instance, a new chapter on eigenvalue problems has been added.

Besides the names mentioned above, I like to thank L. Banjai, P. Benner, W. Kreß, M. Löhndorf, J.M. Melenk, S. Sauter, as well as further coworkers and guests of the Max-Planck Institute in Leipzig, who have contributed to the content of this monograph.

The author also wishes to express his gratitude to the publisher Springer for their friendly cooperation. In particular, he thanks Ann Kostant, editorial consultant of Springer, for polishing the English.

Leipzig and Kiel, July 2015

Wolfgang Hackbusch

Hierarchical Matrices: Algorithms and Analysis

Hackbusch, W.

2015, XXV, 511 p. 87 illus., 27 illus. in color., Hardcover

ISBN: 978-3-662-47323-8