MAX-PLANCK-GESELLSCHAFT

Peter Benner Thomas Mach

The LR Cholesky Algorithm for Symmetric Hierarchical Matrices

MAX–PLANCK–INSTITUT
FÜR DYNAMIK KOMPLEXER
TECHNISCHER SYSTEME
MAGDEBURG

Max Planck Institute Magdeburg Preprints

MPIMD/12-05 February 28, 2012

Impressum:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for Dynamics of Complex Technical Systems

Address:
Max Planck Institute for Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

www.mpi-magdeburg.mpg.de/preprints

THE LR CHOLESKY ALGORITHM FOR SYMMETRIC HIERARCHICAL MATRICES

PETER BENNER AND THOMAS MACH

ABSTRACT. We investigate the application of the LR Cholesky algorithm to symmetric hierarchical matrices, symmetric simple structured hierarchical matrices and symmetric hierarchically semiseparable (HSS) matrices. The data-sparsity of these matrices make the otherwise expensive LR Cholesky algorithm applicable, as long as the data-sparsity is preserved. We will see in an example that the data-sparsity of hierarchical matrices is not well preserved.

We will explain this behavior by applying a theorem on the structure preservation of diagonal plus semiseparable matrices under LR Cholesky transformations. Therefore we have to give a new more constructive proof for the theorem. We will show that the structure of $\mathcal{H}_\ell$-matrices is almost preserved and so the LR Cholesky algorithm is of almost quadratic complexity for $\mathcal{H}_\ell$-matrices.

1. INTRODUCTION

The LR algorithm and its symmetric version, the LR Cholesky algorithm, were invented by Rutishauser in the 1950s [Rut55, Rut58, RS63]. They are the predecessors of the QR algorithm, one of the most often used algorithms for the computation of the eigenvalues of a matrix M, see, e.g. [GV96, BDD$^+$00, Wat07]. In the last years, QR-like algorithms have been developed for computing the eigenvalues of semiseparable matrices [Fas05, DV05, DV06, VVM05a, VVM05b, BBD11]. All these algorithms are used to solve the eigenvalue problem, the computation of eigenpairs (λ, v) solving

$$Mv = \lambda v.$$

We focus on the symmetric eigenvalue problem, there M is symmetric, $M = M^T$. Further we assume $M \in \mathbb{R}^{n\times n}$. These assumptions ensure that all eigenpairs are real, $(\lambda, v) \in \mathbb{R} \times \mathbb{R}^n$, see, e.g., [Par80].

Here we will investigate the application of the LR Cholesky algorithm to symmetric hierarchical matrices. Hierarchical matrices are an important class of data-sparse matrices. Data-sparse means that a dense matrix is represented with an almost linear amount of data. The semiseparable matrices are data-sparse, too. The further relationship between semiseparable and hierarchical matrices will be explained in Subsection 3.3.

As the operations required by an LR transformation can be performed efficiently in the arithmetic for hierarchical matrices, this gives rise to the hope of the existence of an LR algorithm for computing all eingevalues of hierarchical matrices with almost quadratic complexity. This would require that the block-ranks in in the hierarchical data format remain bounded. The aim of this paper is to investigate this hypothesis and to show eventually that it is not true.

Hierarchical matrices are an efficient way to handle a large class of dense matrices, e.g discretizations of partial differential operators and their inverses. Recently different eigenvalue algorithms for symmetric hierarchical matrices have been investigated, see [BM12a, BM12b]. In the next

---

Date: March 2, 2012.

Peter Benner, Max Planck Institute for Dynamics of Complex Technical Systems, Computational Methods in Systems and Control Theory, Sandtorstr. 1, 39106 Magdeburg, Germany, and Technische Universität Chemnitz, Fakultät für Mathematik, Reichenhainer Str. 41, 09126 Chemnitz, Germany, benner@mpi-magdeburg.mpg.de.

Thomas Mach (corresponding author), Max Planck Institute for Dynamics of Complex Technical Systems, Computational Methods in Systems and Control Theory, Sandtorstr. 1, 39106 Magdeburg, Germany, thomas.mach@googlemail.com.

subsections we will briefly introduce the LR Cholesky algorithm and the concept of hierarchical matrices.

1.1. **LR Cholesky Algorithm.** The QR algorithm consists of QR iterations basically of the form

$$\begin{aligned} Q_{i+1}R_{i+1} &= f_i(M_i), \\ M_{i+1} &= Q_{i+1}^{-1}M_iQ_{i+1}. \end{aligned}$$

The iteration converges towards a (block-) diagonal matrix with the eigenvalues on the (block-) diagonal. This works for many other decompositions $M = GR$, too [WE95, BFW97]. The matrix G has to be non-singular and R upper-triangular. The algorithm of the form

$$\begin{aligned} G_{i+1}R_{i+1} &= f_i(M_i), \\ M_{i+1} &= G_{i+1}^{-1}M_iG_{i+1}, \end{aligned}$$

is called GR algorithm driven by f [Wat00]. The functions f_i are used to accelerate the convergence. For instance $f_i(M) = M - \mu_i I$ is used in the single shift iteration and $f_i(M) = (M - \mu_{i,1}I)(M - \mu_{i,2}I)\cdots(M - \mu_{i,d}I)$ yields a multiple-shift strategy.

We will assume that M is positive definite, so that the Cholesky decomposition can be used. The LR Cholesky algorithm consists of LR Cholesky transformations:

$$\begin{aligned} L_{i+1}L_{i+1}^T &= M_i - \mu_i I \\ M_{i+1} &= L_{i+1}^{-1}M_iL_{i+1} = L_{i+1}^{-1}\left(L_{i+1}L_{i+1}^T + \mu_i I\right)L_{i+1} = L_{i+1}^T L_{i+1} + \mu_i. \end{aligned} \tag{1}$$

The LR (Cholesky) algorithm is the historically first invented GR algorithm. In order to shorten the notation, we will call the operator, which gives us the next iterate LRCH,

$$M_{i+1} =: \mathrm{LRCH}(M_i).$$

In the next subsection we will briefly explain the structure of hierarchical matrices.

1.2. **Hierarchical Matrices.** The concept of hierarchical, short $\mathcal{H}$-, matrices was introduced by Hackbusch in 1998 [Hac99]. Hierarchical matrices enable us to compute data-sparse approximations of linear-polylogarithmic storage complexity to a wide range of dense matrices, see e.g. [Hac09] for examples. We give here only a really brief review of the properties of hierarchical matrices, for more details, exact definitions and theorems see e.g. [GH03, BGH03].

The basic idea of the $\mathcal{H}$-matrix format is to use a hierarchical structure to find and access submatrices with good low-rank approximations and to use them to reduce their storage amount and the computation time involving these submatrices. These low-rank approximations make the $\mathcal{H}$-matrix format data-sparse. The need for truncation in order to close the class of $\mathcal{H}$-matrices under addition, multiplication and inversion renders formal $\mathcal{H}$-arithmetic an approximative arithmetic.

The blocks of low rank are called admissible and the smaller, dense ones, inadmissible. There is an admissibility condition telling us which blocks are admissible, and which blocks need further subdivision. Blocks which are in one dimension smaller than $n_{\min}$ will be stored as dense matrices, since a further subdivision would not increase the computational efficiency.

The basic assumption is that there is a hierarchical structure in the background, which enables us to find and access the submatrices. This hierarchical structure is important for the storage efficiency, a hierarchical matrix M typically requires a storage of $\mathcal{O}(kn\log n)$, where n is the size of the matrix and k the maximal block-wise rank. Further there is a lot of formatted arithmetic requiring only linear-polylogarithmic complexity (M_1, M_2 are $\mathcal{H}$-matrices, $v \in \mathbb{R}^n$):

$$\begin{aligned} &M_1 *_{\mathcal{H}} v : && N_{\mathcal{H}*v}(T_{I\times I}, k) \in \mathcal{O}(kn\log n), \\ &M_1 +_{\mathcal{H}} M_2, M_1 -_{\mathcal{H}} M_2 : && N_{\mathcal{H}+\mathcal{H}}(T_{I\times I}, k) \in \mathcal{O}(k^2 n\log n), \\ &M_1 *_{\mathcal{H}} M_2, (M_1)_{\mathcal{H}}^{-1}, \mathcal{H}\mathrm{LU}(M_1) : && N_{\mathcal{H}*\mathcal{H}}/N_{\mathcal{H}^{-1}}/N_{LU(\mathcal{H})} \in \mathcal{O}(k^2 n\log^2 n). \end{aligned}$$

These arithmetic operations (and a lot more) are implemented in the $\mathcal{H}$lib [HLi09], which we use for the numerical examples.

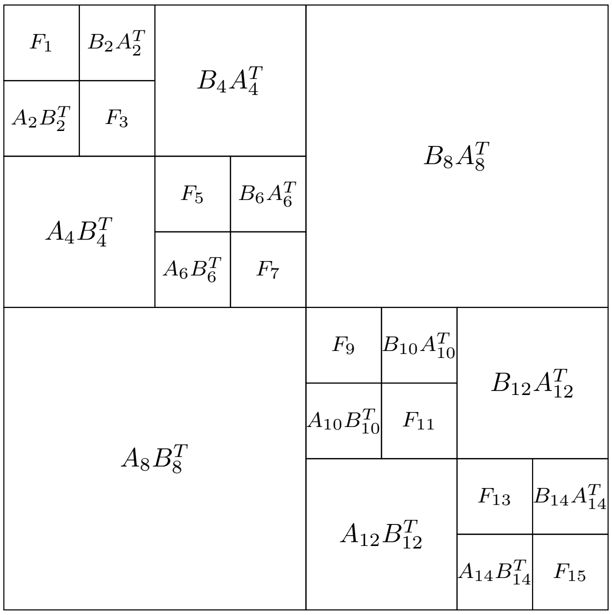

FIGURE 1. Structure of an $\mathcal{H}_3(k)$-matrix.

We will use fixed accuracy $\mathcal{H}$-arithmetic, so we choose the block-wise rank in each truncation with respect to the given accuracy ϵ. The costs of the $\mathcal{H}$-arithmetic depends on the maximal block-wise rank k. Thus in fixed accuracy $\mathcal{H}$-arithmetic the costs depend on $\log \epsilon$.

The $\mathcal{H}$-matrices with a simple structure analogous to Figure 1 are called *$\mathcal{H}_\ell$-matrices*, with ℓ the depth of the hierarchical structure.

Here we will use the Cholesky decomposition [Gra01, Beb08], the QR decomposition [BM10] and the matrix-matrix product for $\mathcal{H}$-matrices. Since the arithmetic operations for hierarchical matrices are of almost linear complexity and in general $\mathcal{O}(n)$ iterations are required to find all n eigenvalues, we expect to get an algorithm of almost quadratic complexity.

Like in the dense case, where one uses matrices of Hessenberg form, we require that the structure of M is preserved under LR Cholesky transformations. We will see that this is not the case for $\mathcal{H}$-matrices. So we will not present an algorithm of almost quadratic complexity, but an argument why such an algorithm does not exist for general $\mathcal{H}$-matrices. In the case of $\mathcal{H}_\ell$-matrices we will show that the structure is almost preserved, so that we finally get an eigenvalue algorithm of almost quadratic complexity for a subset of $\mathcal{H}$-matrices

In Section 2 we describe a numerical experiment, demonstrating the performance of the LR Cholesky algorithm for $\mathcal{H}$-matrices. We will see that the ranks of the admissible blocks are not preserved under LR Cholesky transformation. Afterwards, we give a new proof for the fact that diagonal plus semiseparable matrices are invariant under LR Cholesky transformation and use this proof to explain why the algorithm works for tridiagonal, band and $\mathcal{H}_\ell$-matrices, but not for general $\mathcal{H}$-matrices. The result for $\mathcal{H}_\ell$-matrices is substantiated in Section 4 by numerical experiments. In Section 5 we extend our new proof to the unsymmetric case.

2. LR ALGORITHMS FOR SYMMETRIC $\mathcal{H}$-MATRICES

2.1. **LR Cholesky Transformations.** We will use the LR Cholesky transformation [Rut58], since M is symmetric, see Equation (1). The matrix $f_i(M_i)$ has to be symmetric positive definite, since otherwise the Cholesky factorization does not exist. Special shift strategies ensure this, see [Rut60, Wil65]. But it is not necessary to be too rigorous, because the Cholesky factorization will detect negative eigenvalues and this will give a new upper bound for the shifts and maybe a new

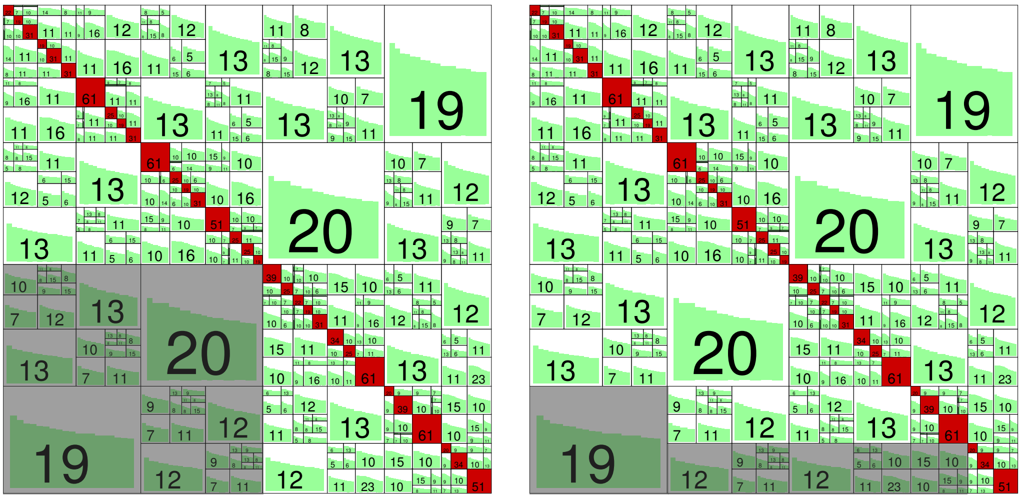

FIGURE 2. Examples for deflation.

lower bound, see [Rut60]. We prefer the computation of the shift by five steps of inverse iteration

$$y_{i+1} := L^{-T}L^{-1}\mu_i y_i, \qquad \mu_{i+1} = \frac{1}{\|y_{i+1}\|_2}, \qquad i = 1, \dots, 5,$$

and since this would lead to an approximation of the smallest eigenvalue from above we subtract a multiple of the error estimation

$$\left\| M^{-1}y_5 - \frac{1}{\mu_5}y_5 \right\|_2.$$

If the smallest eigenvalue is found, we have to deflate in order to continue with a matrix, whose smallest eigenvalue is larger. We will deflate if the norm of the last line except the diagonal element is lower a given $\epsilon_{\text{deflation}}$. A second type of deflation is detected if all $\mathcal{H}$-matrix blocks in $M_{i:n,1:i-1}$ have block-rank 0. For example, the matrix will be deflated if the gray marked blocks in Figure 2 are of block-wise rank 0 (w.r.t. a prescribed deflation tolerance). If there is only one inadmissible block left after deflation, then we use the LAPACK [ABB+99] dense eigenproblem solvers for the remaining matrix.

We implement all these steps using the $\mathcal{H}$lib1.3 [HLi09]. We test our code with an example series out of the $\mathcal{H}$lib1.3, the finite element discretization matrices of the 2D Laplacian on the unit square. We will call these matrices FEMX, where X is the number of discretization points in each direction. Since the eigenvalues of these matrices are known, we can compare the computed eigenvalues with the exact ones. The algorithm computes approximations to the eigenvalues within the expected error tolerance of $\mathcal{H}$-arithmetic accuracy $\epsilon = 10^{-5}$ times the number of steps, see Table 1.

But our matrix has too many dense blocks after 10 steps, see Figure 3. Each square represents a leave of the hierarchical product tree resp. a submatrix. Inadmissible leaves are red (dark gray). Admissible leaves are green (light gray). The dark green (dark gray) leaves are of full rank but stored in the rank-k format since the block was admissible in the original matrix. The number inside the square is the rank of the submatrix. The vertical bars inside the square show the singular values of the submatrix on a logarithmic scale, where the lower edge is 10^{-5}, the chosen $\mathcal{H}$-arithmetic accuracy.

The arithmetic for the dense blocks has cubic complexity. The dense blocks have together a size of $\mathcal{O}(n) \times \mathcal{O}(n)$, so that the complexity of the whole algorithm is not almost quadratic but cubic, see again Table 1.

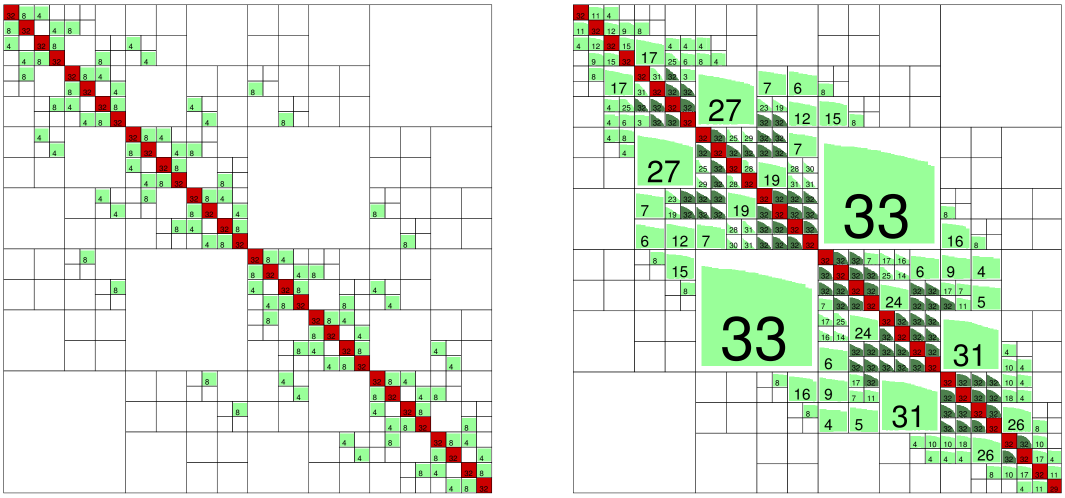

FIGURE 3. Structure of FEM32 (left) and FEM32 after 10 steps of LR Cholesky transformation (right).

Name	n	t_i in s	t_i/t_{i-1}	rel. Err.	Iterations
FEM8	64	0.05		1.5534 e−08	101
FEM16	256	4.09	82	5.2130 e−07	556
FEM32	1 024	393.73	96	5.2716 e−06	2 333
FEM64	4 096	45 757.18	116	1.6598 e−03	10 320

TABLE 1. Numerical results for the LR Cholesky algorithm applied to FEM-series.

2.2. **QR Algorithm.** The QR algorithm has some advantages compared with the LR Cholesky transformation. The QR algorithm can be used for non-symmetric and indefinite matrices, too. This allows the usage of any shift. Further one can show that one QR iteration is equivalent to two LR Cholesky iterations [Xu98].

Since both algorithms are equivalent in a sense it is not surprising that the QR algorithm also leads to too many blocks of full rank. For the FEM32 example, this state lasts a few hundred iterations before the convergence towards a diagonal matrix starts to reduce the ranks of most off-diagonal blocks, like in the LR Cholesky transformation. This effect of the convergence is too late, since the decomposition of an $\mathcal{H}$-matrix with submatrices of full rank covering $n/4 \times n/4$ leads to a complexity of $\mathcal{O}(n^3)$.

Obviously the algorithm does not have an almost quadratic complexity. In the next section we will explain this behavior.

3. Structure Preservation under LR Cholesky Transformation

In this section we will prove that the structure of symmetric diagonal plus semiseparable matrices of rank r is preserved under LR Cholesky transformations.

Definition 3.1. *(diagonal plus semiseparable matrix)*
Let $M = M^T \in \mathbb{R}^{n\times n}$. If M can be written (using MATLAB® notation) in the form

$$M = \operatorname{diag}(d) + \sum_{i=1}^{r} \left(\operatorname{tril}\left(u^i v^{iT}\right) + \operatorname{triu}\left(v^i u^{iT}\right)\right), \tag{2}$$

with $d, u^i, v^i \in \mathbb{R}^n$, then M is a symmetric (generator representable) diagonal plus semiseparable matrix of rank r. *We say M is a dpss matrix for short.*

Obviously this representation of M is storage efficient only for $r < n$. We will also use this notation if r is larger than n, so that $\mathcal{H}$-matrices fit in this definition, too, see Section 3.3.

The following theorem will be applied to matrices of various structure. We will give an explanation why the LR Cholesky algorithm is efficient for tridiagonal and band matrices as well as for rank structured matrices and is inefficient for general hierarchical matrices. Further we will show that a small increase in the block-wise rank is sufficient to use the LR Cholesky transformation for $\mathcal{H}_\ell$-matrices.

A similar theorem on the structure preservation of dpss matrices under QR transformation is proven in [Fas05], but we need a more constructive proof here. Also Theorem 3.1 in [PVV08] is not suitable for our argumentation, since the theorem deals with dpss matrices in Givens-vector representation, but we will use ideas of their proof here.

Theorem 3.2. *Let M be a symmetric positive definite diagonal plus semiseparable matrix, with a decomposition like in Equation (2). The Cholesky factor L of $M = LL^T$ can be written in the form*

$$L = \operatorname{diag}\left(\tilde{d}\right) + \sum_{i=1}^{r} \operatorname{tril}\left(u^i \tilde{v}^{iT}\right). \tag{3}$$

Multiplying the Cholesky factors in reverse order gives the next iterate, $N = L^T L$, of the Cholesky LR algorithm. The matrix N has the same form as M,

$$N = \operatorname{diag}\left(\hat{d}\right) + \sum_{i=1}^{r} \left(\operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right) + \operatorname{triu}\left(\tilde{v}^i \hat{u}^{iT}\right)\right). \tag{4}$$

Proof. The diagonal entries of L fulfill:

$$\begin{aligned} L_{pp} &= \sqrt{M_{pp} - L_{p,1:p-1} L_{p,1:p-1}^T}, \\ \tilde{d}_p + \sum_i u_p^i \tilde{v}_p^i &= \sqrt{d_p + \sum_i 2 u_p^i v_p^i - L_{p,1:p-1} L_{p,1:p-1}^T}. \end{aligned} \tag{5}$$

This condition can easily be fulfilled when $p = 1$. Furthermore there is still some freedom for choosing $\tilde{v}_1^i$ if we choose $\tilde{d}_p$ adequately.

The entries below the diagonal fulfill:

$$\begin{aligned} L_{1:p-1,1:p-1} L_{p,1:p-1}^T &= M_{1:p-1,p} \\ L_{1:p-1,1:p-1} L_{p,1:p-1}^T &= \sum_i v_{1:p-1}^i u_p^{iT}. \end{aligned}$$

If we define $\tilde{v}_{1:p-1}^i$ by

$$L_{1:p-1,1:p-1} \tilde{v}_{1:p-1}^i = v_{1:p-1}^i, \tag{6}$$

then $L_{p,1:p-1} = \sum_i u_p^i \tilde{v}_{1:p-1}^{iT}$ and the above condition is fulfilled. The diagonal $\operatorname{diag}\left(\tilde{d}\right)$ result from (5). So the Cholesky factor has the form as in Equation (3).

The idea of first choosing $\tilde{v}_p^i$ by computing the next row of the Cholesky decomposition before we compute the diagonal entry of the last row, is borrowed from [PVV08, Proof of Theorem 3.1].

The next iterate N is the product $L^T L$. We have

$$\begin{aligned} N &= L^T L \\ &= \left(\operatorname{diag}\left(\tilde{d}\right) + \sum_i \operatorname{tril}\left(u^i \tilde{v}^{iT}\right)\right)^T \left(\operatorname{diag}\left(\tilde{d}\right) + \sum_i \operatorname{tril}\left(u^i \tilde{v}^{iT}\right)\right) \\ &= \operatorname{diag}\left(\tilde{d}\right) \operatorname{diag}\left(\tilde{d}\right) + \sum_i \operatorname{diag}\left(\tilde{d}\right) \operatorname{tril}\left(u^i \tilde{v}^{iT}\right) + \sum_i \operatorname{tril}\left(u^i \tilde{v}^{iT}\right)^T \operatorname{diag}\left(\tilde{d}\right) + \\ &\quad + \sum_i \sum_j \operatorname{tril}\left(u^j \tilde{v}^{jT}\right)^T \operatorname{tril}\left(u^i \tilde{v}^{iT}\right) \end{aligned}$$

$$u^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \star \\ \vdots \\ \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} \rightsquigarrow \tilde{u}^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \star \\ \vdots \\ \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} \rightsquigarrow \hat{u}^i = \begin{bmatrix} \star \\ \vdots \\ \star \\ \star \\ \vdots \\ \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad v^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \star \\ \vdots \\ \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} \rightsquigarrow \tilde{v}^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \star \\ \vdots \\ \star \\ \star \\ \vdots \\ \star \end{bmatrix}$$

FIGURE 4. Sparsity pattern of $\hat{u}^i$ and $\tilde{v}^i$.

We will now show that $\operatorname{tril}(N,-1) = \sum_i \operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}, -1\right)$:

$$\operatorname{tril}(N,-1) = \sum_i \operatorname{diag}\left(\tilde{d}\right) \operatorname{tril}\left(u^i \tilde{v}^{iT}, -1\right) + \operatorname{tril}\left(\sum_i \sum_j \operatorname{tril}\left(u^j \tilde{v}^{jT}\right)^T \operatorname{tril}\left(u^i \tilde{v}^{iT}\right), -1\right).$$

The other summands are zero in the lower triangular part. Define $\tilde{u}^i_p := \tilde{d}_p u^i_p,\ \forall p = 1,\dots,n$. So we get

$$\operatorname{tril}(N,-1) = \sum_i \operatorname{tril}\left(\tilde{u}^i \tilde{v}^{iT}, -1\right) + \operatorname{tril}\left(\sum_i \sum_j \underbrace{\operatorname{tril}\left(u^j \tilde{v}^{jT}\right)^T \operatorname{tril}\left(u^i \tilde{v}^{iT}\right)}_{:=T^{ji}}, -1\right).$$

We have $T^{ji}_{pq} = \tilde{v}^j_p u^{jT}_{p:n} u^i_{p:n} \tilde{v}^{iT}_q$, if $p > q$. It holds that

$$u^{jT}_{p:n} u^i_{p:n} = \begin{bmatrix} 0 & \cdots & 0 & u^j_p & u^j_{p+1} & \cdots & u^j_n \end{bmatrix} u^i.$$

We define a matrix Z by

$$Z_{g,h} := \begin{cases} \sum_j \tilde{v}^j_g u^j_h, & g \le h, \\ 0, & g > h. \end{cases}$$

Thus,

$$Z_{p,:} = \sum_j \tilde{v}^j_p \begin{bmatrix} 0 & \cdots & 0 & u^j_p & u^j_{p+1} & \cdots & u^j_n \end{bmatrix}.$$

Finally we get

$$\operatorname{tril}(N,-1) = \sum_i \operatorname{tril}\left(\left(\tilde{u}^i + Z u^i\right) \tilde{v}^{iT}, -1\right) = \sum_i \operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}, -1\right). \tag{7}$$

Since N is symmetric, the analogue is true for the upper triangle. □

Remark 3.3. *An analog proof, see Section 5, exists for the unsymmetric case, where the semiseparable structure is preserved under LU transformations.*

Theorem 3.2 tells us that $N = \operatorname{LRCH}(M)$ is the sum:

$$N = \operatorname{diag}\left(\hat{d}\right) + \sum_{i=1}^{r} \left(\operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right) + \operatorname{triu}\left(\tilde{v}^i \hat{u}^{iT}\right)\right),$$

with $\tilde{v}^i$ being the solution of

$$L\tilde{v}^i = v^i,$$

where L is a lower triangular matrix and

$$\hat{u}^i := \left(Z + \operatorname{diag}\left(\hat{d}\right)\right) u^i,$$

and Z is an upper triangular matrix. We define the set of non-zero indices of a vector $x \in \mathbb{R}^n$ by

$$SP(x) = \{i \in \{1, \dots, n\} | x_i \neq 0\}$$

Let i_v be the smallest index in $SP(v^i)$. Then in general $SP(\tilde{v}^i) = \{i_v, \dots, n\}$. The sparsity pattern of $\tilde{v}^i$ is

$$SP(\tilde{v}^i) = \left\{p \in \{1, \dots, n\} \middle| \exists q \in SP(v^i) : q \leq p\right\}.$$

Since $\tilde{u}^i_p = \tilde{d}_p u^i_p$, we have $SP(\tilde{u}^i) = SP(u^i)$. It holds that $\hat{u}^i \neq 0$ if either $\tilde{u}^i \neq 0$ or there is a j such that $\beta^{ji}_p \tilde{v}^j_p \neq 0$. The second condition is in general (if r is large enough) fulfilled for all $p \leq \max_{q \in SP(u^i)} q$. The sparsity patterns of the vector $\hat{u}^i$ is

$$SP(\hat{u}^i) = \left\{p \in \{1, \dots, n\} \middle| \exists q \in SP(u^i) : p \leq q\right\}.$$

The sparsity pattern of $\hat{u}^i$ and $\tilde{v}^i$ are visualized in Figure 4.

Theorem 3.2 shows that the structure of diagonal plus semiseparable matrices is preserved under LR Cholesky transformations. In the following subsections we will use the theorem to investigate the behavior of tridiagonal matrices, matrices with rank structures, $\mathcal{H}$-matrices and $\mathcal{H}_\ell$-matrices under LR Cholesky transformations.

We assume in the following subsections that M is symmetric positive definite, so that the Cholesky decomposition of M is unique.

3.1. **Tridiagonal Matrices.** Let M now be a symmetric tridiagonal matrix. This means $M_{ij} = M_{ji}$ and $M_{ji} = 0$ if $|i - j| > 1$. The matrix M can be written in the form of Equation (2). We have $d = \operatorname{diag}(M)$ and

$$\begin{aligned} u^i &= M_{i+1,i} e_{i+1}, \\ v^i &= e_i, \end{aligned}$$

where e_i is the ith column of the identity matrix.

The matrix $N = \mathrm{LRCH}(M)$ is again tridiagonal, since $\operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right)$ has only non-zero entries for the indices (i, i), $(i + 1, i)$ and $(i + 1, i + 1)$.

An analog argumentation for band matrices with bandwidth $2b$ exists, if

$$u^i = \begin{bmatrix} 0 & \cdots & 0 & M^T_{i+1:i+b,i} & 0 & \cdots & 0 \end{bmatrix}^T.$$

3.2. **Matrices with Rank Structure.** Following [DV05] we define a rank structure by a quadruple (i, j, r, λ). A matrix M satisfies this rank structure if

$$\operatorname{rank}\left((M - \lambda I)_{i:n,1:j}\right) \leq r.$$

In [DV05] it is proven that the rank structures of M are preserved under QR transformation. This rank structure is also preserved under LR Cholesky transformations, since M can be written as diagonal-plus-semiseparable matrix. Therefore we use

$$\operatorname{tril}(M - \lambda I) = \operatorname{tril}\left(\begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ AB^T & \tilde{M}_{22} \end{bmatrix}\right)$$

with the low-rank factorization $AB^T = (M - \lambda I)_{i:n,1:j}$. This leads to

$$\operatorname{tril}(M - \lambda I) = \operatorname{tril}\left(\begin{bmatrix} 0 \\ A \end{bmatrix} \begin{bmatrix} B^T & 0 \end{bmatrix}\right) + \operatorname{tril}\left(\begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \end{bmatrix}\right) + \operatorname{tril}\left(\begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} 0 & \tilde{M}_{22} \end{bmatrix}\right).$$

After the LR Cholesky transformation we get:

$$\operatorname{tril}(N) = \operatorname{tril}\left(\begin{bmatrix} \star \\ \star \end{bmatrix} \begin{bmatrix} \star & \star \end{bmatrix}\right) + \operatorname{tril}\left(\begin{bmatrix} \star \\ 0 \end{bmatrix} \begin{bmatrix} \star & \star \end{bmatrix}\right) + \operatorname{tril}\left(\begin{bmatrix} \star \\ \star \end{bmatrix} \begin{bmatrix} 0 & \star \end{bmatrix}\right) + \operatorname{diag}(d).$$

In the first summand, we have still a low-rank factorization with rank r. The other summands are zero in the lower left block, so that the rank structure is preserved.

3.3. **$\mathcal{H}$-Matrices.** Let $M = M^T$ now be a symmetric $\mathcal{H}$-matrix, with a symmetric hierarchical structure. Under a symmetric hierarchical structure, we will understand that the blocks $M|_b$, where $b = s \times t$, and $M|_{b^T}$, where $b^T = t \times s$, are symmetric, so that b^T is admissible if and only if b is admissible. Further, we assume $M|_b = AB^T = \left(BA^T\right)^T = \left(M|_{b^T}\right)^T$, so that the ranks are equal, $k_b = k_{b^T}$. Since the Cholesky decomposition of M leads to a matrix with invertible diagonal blocks of full rank, we assume that the diagonal blocks of M are inadmissible. Further, all other blocks should be admissible. Inadmissible non-diagonal blocks will be treated as admissible blocks with full rank.

The matrix M can now be written in the form of Equation (2). First we rewrite the inadmissible blocks. We choose block $b = s \times s$ and the diagonal of $M|_b$ forms $d|_s$. For the off-diagonal entries M_{pq} and M_{qp}, we have $|p - q| \leq n_{\min}$. We need at most $n_{\min} - 1$ pairs (u^i, v^i) to represent the inadmissible block by the sum

$$\operatorname{diag}(d) + \sum_i \operatorname{tril}\left(u^i v^{iT}\right) + \operatorname{triu}\left(v^i u^{iT}\right).$$

We choose the first pair, so that the first columns of $u^1 v^{1T}$ and $M|_b$ are equal. The next pair $u^2 v^{2T}$ has to be chosen so that it is equal to $M|_b - u^1 v^{1T}$ in the second column, and so on. Like for block/band matrices, these pairs from inadmissible blocks do not cause problems to the $\mathcal{H}$-structure, since $\operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right)$ has non-zero entries only in the $s \times s = b$ block and this block is an inadmissible block anyway.

Further we have admissible blocks. Each admissible block in the lower triangular is a sum of products:

$$M|_b = AB^T = \sum_{j=1}^{k_b} A_{j,\cdot} \left(B_{j,\cdot}\right)^T.$$

For each pair $(A_{j,\cdot}, B_{j,\cdot})$ we introduce a pair (u^i, v^i) with $u^{iT} = \begin{bmatrix} 0 & \cdots & 0 & A_{j,\cdot}^T & 0 & \cdots & 0 \end{bmatrix}$ and $v^{iT} = \begin{bmatrix} 0 & \cdots & 0 & B_{j,\cdot}^T & 0 & \cdots & 0 \end{bmatrix}$, so that $M|_b = \sum u^i v^{iT}|_b$.

After this has been done for all blocks, we are left with M in the form of Equation (2) with, what is most likely to be, an upper summation index $r \gg n$. under LR Cholesky transformation is not sufficient for the preservation of the $\mathcal{H}$-matrix structure. Further, we require the preservation of the sparsity pattern of u^i and v^i, since otherwise the ranks of other blocks are increased. Exactly this is not the case for general $\mathcal{H}$-matrices, since these pairs have a more general structure and cause larger non-zero blocks in N, as shown in Figure 4. The matrix M has a good $\mathcal{H}$-approximation, since for all i: $\operatorname{tril}\left(u^i v^{iT}\right)$ has non-zeros only in one block of the $\mathcal{H}$-product tree. In the matrix N, the summand $\operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right)$ has non-zeros in many blocks of the $\mathcal{H}$-product tree and we would need a rank-1 summand in each of these blocks to represent the summand $\operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right)$ correctly. This means that the blocks on the upper right hand side of the original blocks have ranks increased by 1. Since this happens for many indices i, recall $r \gg n$, many blocks in N have full or almost full rank. In short N is not representable by an $\mathcal{H}$-matrix of small block-wise rank resp. the $\mathcal{H}$-matrix approximation property of M is not preserved under LR Cholesky transformations.

3.4. **$\mathcal{H}_\ell$-Matrices.** $\mathcal{H}_\ell$-matrices are $\mathcal{H}$-matrices with a simple structure. Let M be an $\mathcal{H}_\ell$-matrix of rank k, so that on the highest level we have the following recursive structure:

$$M = \begin{bmatrix} M_{11} \in \mathcal{H}_{\ell-1} & BA^T \\ AB^T & M_{22} \in \mathcal{H}_{\ell-1} \end{bmatrix}$$

Like in the previous subsection we introduce k pairs (u^i, v^i) for each rank-k-submatrix AB^T. These pairs have the following structure:

$$u^{iT} = \begin{bmatrix} 0 & \cdots & 0 & \left(A|_{i,\cdot}\right)^T \end{bmatrix} \qquad \text{and} \quad v^{iT} = \begin{bmatrix} \left(B|_{i,\cdot}\right)^T & 0 & \cdots & 0 \end{bmatrix},$$

$$\operatorname{tril}\left(u^i v^{iT}\right) = \begin{bmatrix} 0 &&&&&&&&& \\ 0 & 0 &&&&&&&& \\ 0 & 0 & 0 &&&&&&& \\ 0 & 0 & 0 & 0 &&&&&& \\ 0 & 0 & 0 & 0 & 0 &&&&& \\ 0 & 0 & 0 & 0 & 0 & 0 &&&& \\ 0 & 0 & \star & \star & 0 & 0 & 0 &&& \\ 0 & 0 & \star & \star & 0 & 0 & 0 & 0 && \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightsquigarrow \operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right) = \begin{bmatrix} 0 &&&&&&&&& \\ 0 & 0 &&&&&&&& \\ 0 & 0 & \star &&&&&&& \\ 0 & 0 & \star & \star &&&&&& \\ 0 & 0 & \star & \star & \star &&&&& \\ 0 & 0 & \star & \star & \star & \star &&&& \\ 0 & 0 & \star & \star & \star & \star & \star &&& \\ 0 & 0 & \star & \star & \star & \star & \star & \star && \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

FIGURE 5. Example for sparsity patterns of $\operatorname{tril}\left(u^i v^{iT}\right)$ and $\operatorname{tril}\left(\hat{u}^i \tilde{v}^{iT}\right)$.

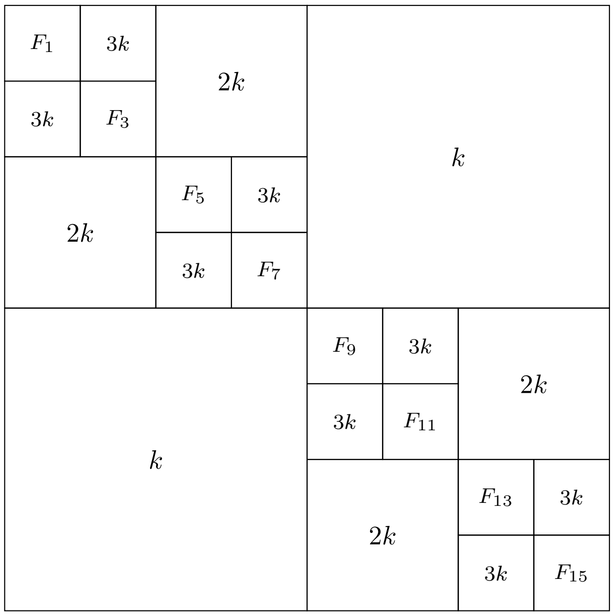

FIGURE 6. Ranks of an $\mathcal{H}_3(k)$-matrix after LR Cholesky transformation.

and so the sparsity patterns are

$$\hat{u}^{iT} = \begin{bmatrix} \star & \cdots & \star & \star & \cdots & \star \end{bmatrix} \qquad \text{and} \quad \tilde{v}^{iT} = \begin{bmatrix} \star & \cdots & \star & \star & \cdots & \star \end{bmatrix},$$

after the LR Cholesky transformation. Like for matrices with rank structure, the non-zeros from the diagonal blocks do not spread into the off-diagonal block of AB^T. The rank of the off-diagonal block on the highest level will be k, as shown in Figure 6.

The rank of the blocks on the next lower level will be increased by k due to the pairs from the highest level. By a recursion we get the rank structure from Figure 6, where the blocks F_i are dense matrices of full rank.

So the structure of $\mathcal{H}_\ell(k)$-matrices is *not* preserved under LR Cholesky transformations, but the maximal block-wise rank is bounded by ℓk. Since the smallest blocks have the largest ranks, the storage required by the low-rank parts of the lower triangular part of M is increased from $nk\ell$ to $nk\frac{\ell(\ell+1)}{2}$, meaning that the matrix has a storage complexity of $\mathcal{O}(kn\left(\log_2 n\right)^2)$ instead of $\mathcal{O}\left(kn \log_2 n\right)$. This is still small enough to perform an LR Cholesky algorithm in almost quadratic complexity.

If M has additionally an HSS structure [XCGL10], $M \in \text{HSS}(k)$, this structure will be destroyed in the first step and M_i will have only the structure of an $\mathcal{H}_\ell(k\ell)$-matrix.

4. Numerical Examples

In the last subsection it was shown that the algorithm described in Section 2 can be used to compute the eigenvalues of $\mathcal{H}_\ell$ matrices. In this section we will test the algorithm. Therefor we use randomly generated $\mathcal{H}_\ell(1)$- and $\mathcal{H}_\ell(2)$-matrices of dimension $n = 64$ to $n = 262\,144$, with a minimum block-size of $n_{\min} = 32$. Since the structure of $\mathcal{H}_\ell$-matrices is almost preserved, we expect that the required CPU time for the algorithm to grow like

$$\mathcal{O}\left(k^2 n^2 (\log_2 n)^2\right),$$

since we expect $\mathcal{O}(n)$ iterations, each computing a Cholesky decomposition of an $\mathcal{H}_\ell(k\ell)$-matrix, which costs $\mathcal{O}\left(k^2 n (\log_2 n)^2\right)$, see [HKK04]. In Figure 7 one can see that for the $\mathcal{H}_\ell(1)$-matrices, the CPU time grows even slower than our expectation. That is probably an effect of the block diagonal structure of the matrix, which is only perturbed by rank-1 matrices in the off-diagonal blocks. Such a structure supports deflation like in the left hand part of Figure 2 after only a few iterations.

For the $\mathcal{H}_\ell(2)$-matrices, we see the k^2 in the complexity estimate.

The graph for the CPU time of the LAPACK [ABB+99] function `dsyev` is only for comparison, as we use LAPACK 3.1.1. We expect that for matrices larger than $10^6 \times 10^6$ our algorithm is faster than the current LAPACK implementation of the dense QR algorithm. The expected CPU time for such a matrix is approx. 20 years. So the main advantage of the $\mathcal{H}$-LR Cholesky algorithm is the reduced storage consumption.

It is obvious that the computing time for `dsyev` is lower than for the $\mathcal{H}_\ell$ LR Cholesky algorithm. Nevertheless, as the latter algorithm's complexity follows the $\mathcal{O}(n^2(\log_2(n))^2)$ curve while `dsyev` has complexity $\mathcal{O}(n^3)$, the LR Cholesky algorithm will become more efficient for large enough n. Given a sophisticated implementation of the new algorithm on a level as available in LAPACK, it can be expected that the difference in computing times becomes smaller, and the break-even point is reached at a much earlier stage than with the current trial implementation.

5. The General Case

If M is not symmetric, then we must use the LU decomposition, which was called LR decomposition by Rutishauser, instead of the Cholesky decomposition. The following lemma and proof are analog to Theorem 3.2.

Lemma 5.1. *Let $M \in \mathbb{R}^{n \times n}$ be a diagonal plus semiseparable matrix of rank (r, s) in generator representable form:*

$$M = \operatorname{diag}(d) + \sum_{j=1}^{r} \operatorname{tril}\left(u^j v^{jT}\right) + \sum_{i=1}^{s} \operatorname{triu}\left(w^i x^{iT}\right).$$

Then the LU decomposition of M leads to

$$L = \operatorname{diag}\left(\tilde{d}\right) + \sum_{j=1}^{r} \operatorname{tril}\left(u^j \tilde{v}^{jT}\right),$$

$$U = \operatorname{diag}(\tilde{e}) + \sum_{i=1}^{s} \operatorname{triu}\left(\tilde{w}^i x^{iT}\right).$$

The multiplication in reverse order gives the next iterate $N = UL$ of the form

$$N = \operatorname{diag}\left(\hat{d}\right) + \sum_{j=1}^{r} \operatorname{tril}\left(\hat{u}^j \tilde{v}^{jT}\right) + \sum_{i=1}^{s} \operatorname{triu}\left(\tilde{w}^i \hat{x}^{iT}\right),$$

where r and s are unchanged.

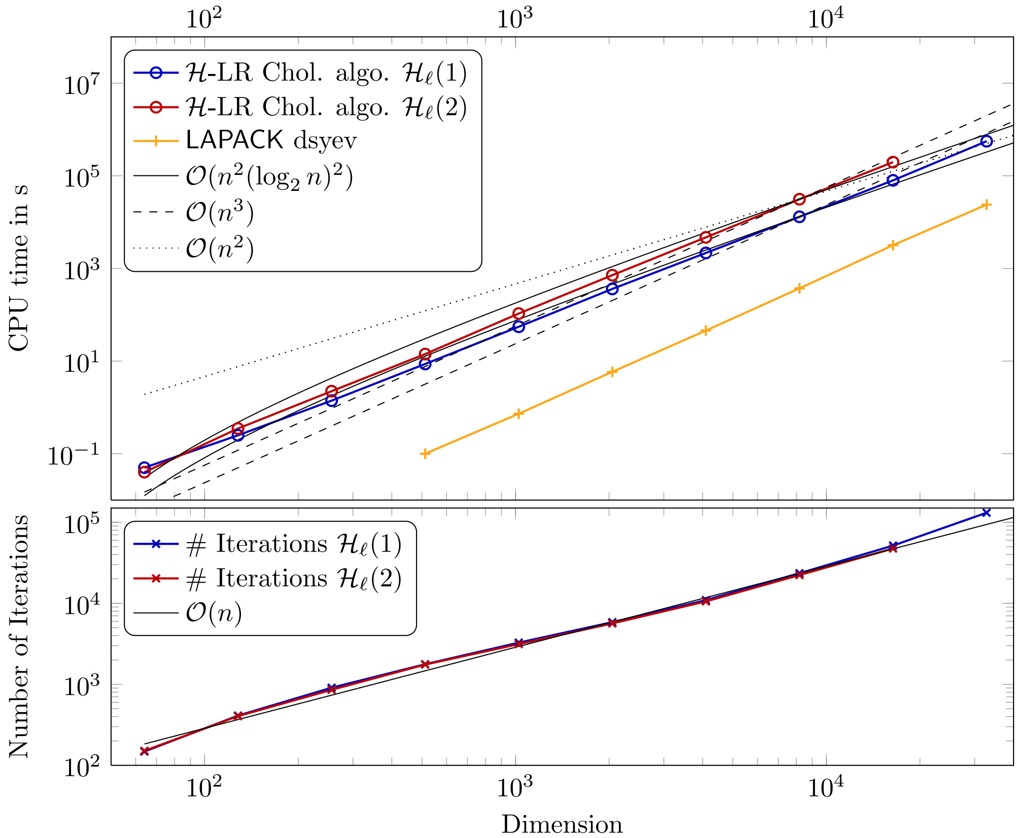

FIGURE 7. CPU time LR Cholesky algorithm for $\mathcal{H}_\ell(1)$, $n_{\min} = 32$.

Proof. From $M = LU$ we know that

$$L_{p,1:p-1}U_{1:p-1,1:p-1} = M_{p,1:p-1}, \qquad L_{p,p} = 1, \tag{8}$$

$$L_{1:p-1,1:p-1}U_{p,1:p-1} = M_{1:p-1,p}, \qquad U_{p,p} = M_{p,p} - L_{p,1:p-1}U_{1:p-1,p}. \tag{9}$$

The argumentation is now analog to the one in the proof of Theorem 3.2. For each p we first compute the new column of U, then the diagonal entry of the last column of U, and finally the new row of L. We assume U has the form

$$U = \operatorname{diag}(\tilde{e}) + \sum_{i=1}^{s} \operatorname{triu}\left(\tilde{w}^i \tilde{x}^{iT}\right),$$

then Equation (9) becomes

$$L_{1:p-1,1:p-1}\tilde{w}^i_{1:p-1}\tilde{x}^i_p = w^i_{1:p-1}x^i_p \qquad \forall i \in \{1,\ldots,s\}.$$

This equation holds for $\tilde{x}^i = x^i$ and $L\tilde{w}^i = w^i$ and can be solved up to row $p-1$, since $L_{p-1,p-1} = 1$ by definition. The equation for the diagonal entry $U_{p-1,p-1}$ is fulfilled by choosing a suitable $\tilde{e}_{p-1}$. Further, we assume L to be of the form

$$L = \operatorname{diag}\left(\tilde{d}\right) + \sum_{j=1}^{r} \operatorname{triu}\left(\tilde{u}^j \tilde{v}^{jT}\right),$$

meaning that we must choose $\tilde{d}_p$ so that $L_{pp} = 1$. Further, we have to fulfill Equation (8),

$$\tilde{u}^j_p \tilde{v}^j_{1:p-1} U_{1:p-1,1:p-1} = u^j_p v^{jT}_{1:p-1}.$$

This can be achieved by setting $\tilde{u} = u$ and

$$U^T_{1:p-1,1:p-1}\tilde{v}^j_{1:p-1} = v^j_{1:p-1}.$$

So both factors have the desired form.

The next iterate is computed by

$$\begin{aligned}
N = UL &= \left(\operatorname{diag}(\tilde{e}) + \sum_{i=1}^{s} \operatorname{triu}\left(\tilde{w}^i x^{iT}\right)\right)\left(\operatorname{diag}\left(\tilde{d}\right) + \sum_{j=1}^{r} \operatorname{tril}\left(u^j \tilde{v}^j\right)\right)\\
&= \operatorname{diag}(\tilde{e}) \operatorname{diag}\left(\tilde{d}\right) + \sum_{j=1}^{r} \operatorname{diag}(\tilde{e}) \operatorname{tril}\left(u^j \tilde{v}^j\right) + \sum_{i=1}^{s} \operatorname{triu}\left(\tilde{w}^i x^{iT}\right) \operatorname{diag}\left(\tilde{d}\right)\\
&\quad + \sum_{i=1}^{s}\sum_{j=1}^{r} \operatorname{triu}\left(\tilde{w}^i x^{iT}\right) \operatorname{tril}\left(u^j \tilde{v}^j\right).
\end{aligned}$$

We will now show that $\operatorname{tril}(N,-1) = \sum_{j=1}^{r} \operatorname{tril}\left(\hat{u}^j \tilde{v}^{jT}, -1\right)$:

$$\operatorname{tril}(N,-1) = \sum_{j=1}^{r} \operatorname{diag}(\tilde{e}) \operatorname{tril}\left(u^j \tilde{v}^j, -1\right) + +\operatorname{tril}\left(\sum_{i=1}^{s}\sum_{j=1}^{r} \underbrace{\operatorname{triu}\left(\tilde{w}^i x^{iT}\right) \operatorname{tril}\left(u^j \tilde{v}^j\right)}_{:=T^{ij}}, -1\right).$$

The other summands are zero in the lower triangular. We have $T^{ij}_{pq} = \tilde{w}^i_p x^{iT}_{p:n} u^j_{p:n} \tilde{v}^{jT}_q$, if $p > q$. We define a matrix Z by

$$Z_{p,:} = \sum_{i=1}^{s} \tilde{w}^i_p \begin{bmatrix} 0 & \cdots & 0 & x^i_p & x^i_{p+1} & \cdots & x^i_n \end{bmatrix}.$$

Finally we get

$$\operatorname{tril}(N,-1) = \sum_{j=1}^{r} \operatorname{tril}\left(\left(\operatorname{diag}(\tilde{e})\, u^j + Z u^j\right) \tilde{v}^{jT}, -1\right) = \sum_{j=1}^{r} \operatorname{tril}\left(\hat{u}^j \tilde{v}^{jT}, -1\right).$$

The analog argumentation for the upper triangular of N completes the proof. □

The result of the proof is similar to the symmetric case in the lower triangular we get sparsity patterns like in Figure 5. Analog in the upper triangular but for the transpose version of Figure 5. This mean for hierarchical matrices that also the unsymmetric LR transformations destroys the structure.

6. Conclusions

We have presented a new more constructive proof for the fact that the structure of diagonal plus semiseparable matrices is invariant under LR Cholesky transformations. We used the knowledge about the structure of $N = \mathrm{LRCH}(M)$ that we acquired by the proof, to show once again that rank structured matrices and especially the subsets of tridiagonal and band matrices are invariant under LR Cholesky transformations. Besides this, we showed that a small increase of the blockwise ranks of $\mathcal{H}_\ell$-matrices is sufficient to compute the eigenvalues with an LR Cholesky algorithm within the structure of $\mathcal{H}_\ell(k\ell)$-matrices. The same is true for the subset of HSS matrices.

Further, we used the theorem to show that the structure of hierarchical matrices is not preserved under LR Cholesky transformations in general. There are subsets of $\mathcal{H}$-matrices, where the LR Cholesky algorithm works well, like the $\mathcal{H}_\ell$-matrices. If one finds a way to transform an $\mathcal{H}$-matrix into an $\mathcal{H}_\ell$-matrix or any other LR Cholesky transformations invariant structure, then one would be able to compute the eigenvalues using the LR Cholesky transformation. So we are missing a generalization of the Hessenberg transformation for hierarchical matrices. In [DFV09] such a transformation is given for $\mathcal{H}^2$-matrices with a special block-structure. This path to an eigenvalue algorithm for $\mathcal{H}^2$-matrices deserves further investigation.

References

[ABB+99] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, PA, third edition, 1999.

[BBD11] R. Bevilacqua, E. Bozzo, and G. M. Del Corso. qd-type methods for quasiseparable matrices. *SIAM J. Matrix Anal. Appl.*, 32(3):722–747, 2011.

[BDD+00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, PA, 2000.

[Beb08] M. Bebendorf. *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, volume 63 of *Lecture Notes in Computational Science and Engineering (LNCSE)*. Springer Verlag, Berlin Heidelberg, 2008.

[BFW97] P. Benner, H. Faßbender, and D. S. Watkins. Two connections between the SR and HR eigenvalue algorithms. *Linear Algebra Appl.*, 272:17–32, 1997.

[BGH03] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Eng. Anal. Boundary Elements*, 27:405–422, 2003.

[BM10] P. Benner and T. Mach. On the QR decomposition of $\mathcal{H}$-matrices. *Computing*, 88(3–4):111–129, 2010.

[BM12a] P. Benner and T. Mach. Computing all or some eigenvalues of symmetric $\mathcal{H}_\ell$-matrices. *SIAM J. Sci. Comput.*, 34(1):A485–A496, 2012.

[BM12b] P. Benner and T. Mach. The preconditioned inverse iteration for hierarchical matrices. *Numer. Lin. Alg. Appl.*, 2012. published online. 17 pages.

[DFV09] S. Delvaux, K. Frederix, and M. Van Barel. Transforming a hierarchical into a unitary-weight representation. *Electr. Trans. Num. Anal.*, 33:163–188, 2009.

[DV05] S. Delvaux and M. Van Barel. Structures preserved by the QR-algorithm. *J. Comput. Appl. Math.*, 187(1):29–40, 2005.

[DV06] S. Delvaux and M. Van Barel. Rank structures preserved by the QR-algorithm: the singular case. *J. Comput. Appl. Math.*, 189:157–178, 2006.

[Fas05] D. Fasino. Rational Krylow matrices and QR steps on Hermitian diagonal-plus-semiseparable matrices. *Numer. Lin. Alg. Appl.*, 12:743–754, 2005.

[GH03] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[Gra01] L. Grasedyck. *Theorie und Anwendungen Hierarchischer Matrizen*. Dissertation, University of Kiel, Kiel, Germany, 2001. In German, available at `http://e-diss.uni-kiel.de/diss_454`.

[GV96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[Hac99] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[Hac09] W. Hackbusch. *Hierarchische Matrizen. Algorithmen und Analysis*. Springer-Verlag, Berlin, 2009.

[HKK04] W. Hackbusch, B. N. Khoromskij, and R. Kriemann. Hierarchical matrices based on a weak admissibility criterion. *Computing*, 73:207–243, 2004.

[HLi09] $\mathcal{H}$lib 1.3. `http://www.hlib.org`, 1999–2009.

[Par80] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, first edition, 1980.

[PVV08] B. Plestenjak, M. Van Barel, and E. Van Camp. A Cholesky LR algorithm for the positive definite symmetric diagonal-plus-semiseparable eigenproblem. *Linear Algebra Appl.*, 428:586–599, 2008.

[RS63] H. Rutishauser and H. R. Schwarz. The LR transformation method for symmetric matrices. *Numer. Math.*, 5(1):273–289, 1963.

[Rut55] H. Rutishauser. Une méthode pour la détermination des valeurs propres d'une matrice. *C. R. Math. Acad. Sci. Paris*, 240:34–36, 1955.

[Rut58] H. Rutishauser. Solution of eigenvalue problems with the LR-transformation. *Nat. Bur. Standards Appl. Math. Ser.*, 49:47–81, 1958.

[Rut60] H. Rutishauser. Über eine kubisch konvergente Variante der LR-Transformation. *Z. Angew. Math. Mech.*, 40(1-3):49–54, 1960.

[VVM05a] R. Vandebril, M. Van Barel, and N. Mastronardi. An implicit Q theorem for Hessenberg-like matrices. *Mediterranean J. Math.*, 2:259–275, 2005.

[VVM05b] R. Vandebril, M. Van Barel, and N. Mastronardi. An implicit QR algorithm for symmetric semiseparable matrices. *Numer. Lin. Alg. Appl.*, 12(7):625–658, 2005.

[Wat00] D. S. Watkins. *QR*-like algorithms for eigenvalue problems. *J. Comput. Appl. Math.*, 123:67–83, 2000.

[Wat07] D. S. Watkins. *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*. SIAM, Philadelphia, PA, USA, 1 edition, 2007.

[WE95] D. S. Watkins and L. Elsner. Convergence of algorithms of decomposition type for the eigenvalue problem. *Linear Algebra Appl.*, 143, 1995.

[Wil65] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford, 1965.

[XCGL10] J. Xia, S. Chandrasekaran, M. Gu, and X.S. Li. Fast algorithms for hierachically semiseparable matrices. *Numer. Lin. Alg. Appl.*, 17(6):953–976, 2010.

[Xu98] H. Xu. The relation between the QR and LR algorithms. *SIAM J. Matrix Anal. Appl.*, 19(2):551–555, 1998.