Computing the Eigenvalues of Hierarchical Matrices by LR-Cholesky Transformations

Peter Benner

(joint work with Thomas Mach)

We investigate the use of Rutishauser’s LR-Cholesky transformation [8] to compute all eigenvalues of symmetric hierarchical ($\mathcal{H}$-) matrices. Historically, the LR transformation was the first algorithm of the class of GR algorithms [11].

\[
L_{m+1}L_{m+1}^T = M_{m} - \mu_m I,
\]
\[
M_{m+1} = L_{m+1}^T L_{m+1} = L_{m+1}^{-1} M_{m} L_{m+1}.
\]

The hierarchical matrix format, see, e.g., [6, 7, 4], allows storing a variety of dense matrices from certain applications in a special data-sparse way with linear-polylogarithmic complexity. Most $\mathcal{H}$-arithmetic operations, including the $\mathcal{H}$-Cholesky decomposition and the $\mathcal{H}$-QR decomposition [1], have linear-polylogarithmic complexity, too. As for simpler structured matrices, like semiseparable matrices, GR algorithms exist [10], it is plausible that an eigenvalue algorithm of bulge-chasing type might also exist for $\mathcal{H}$-matrices. For this GR algorithms $O(n)$ iterations are sufficient, so we expect to find an algorithm for $\mathcal{H}$-matrices with a complexity of $O(n^2 \log^\alpha n)$. The quest for such an algorithm poses merely an academic challenge, but it might become useful in a variety of applications requiring many (small, inner) eigenvalues of differential operators discretized by boundary- or finite-element methods.

The first straightforward approach is to substitute the Cholesky decomposition in the LR-Cholesky transformation (1) by the $\mathcal{H}$-Cholesky decomposition. This yields a meaningful LR-Cholesky algorithm in $\mathcal{H}$-matrix arithmetic, since the limit of the iterative process is a diagonal matrix and hence an $\mathcal{H}$-matrix, too. But some of the intermediate iterates may not have good $\mathcal{H}$-matrix approximations. If we multiply two $\mathcal{H}$-matrices, the rank of the product is bounded by [5]

\[
k_{\mathcal{H}\mathcal{H}} \leq C_{id} C_{sp} (p + 1) k_{\mathcal{H}},
\]

with $C_{id}$ the idempotency constant, $C_{sp}$ the sparsity constant and $p$ the depth of the $\mathcal{H}$-tree. This means, that the maximal rank of the matrix $M_m$ can grow really fast with each iteration step and they actually do, see Figure 1. This can be regarded as the $\mathcal{H}$-arithmetic analog to fill-in. Large blocks of full rank destroy the data sparsity of the hierarchical matrix and increase the complexity to $O(n^3)$.

![Figure 1. Matrix FEM32 (left) and the same matrix after 10 steps (with shift) of LR Cholesky transformation (right), dark green and red (dark grey) blocks have full rank](image)

In GR algorithms it is necessary to shift the matrix $M_m$ to accelerate the convergence. In the case of LR-Cholesky transformation the shift must preserve the positive definiteness of $M_m$. There are shift strategies ensuring this by Rutishauser...
### Table 1. Computation times for LR Cholesky transformation

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Dimension</th>
<th>Time</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEM8</td>
<td>64</td>
<td>0.04</td>
<td>9.4e-6</td>
</tr>
<tr>
<td>FEM16</td>
<td>256</td>
<td>3.20</td>
<td>1.1e-2</td>
</tr>
<tr>
<td>FEM32</td>
<td>1024</td>
<td>243.05</td>
<td>4.2e-2</td>
</tr>
<tr>
<td>FEM64</td>
<td>4096</td>
<td>11,116.96</td>
<td>2.6e-1</td>
</tr>
</tbody>
</table>

We use some steps of inverse power iteration to find an approximation of the smallest eigenvalue and subtract a safety margin to be sure having a shift smaller than the smallest eigenvalue.

If the smallest eigenvalue $\lambda_n$ is found, we have to deflate this eigenvalue. After the deflation we can increase the shift, since the smallest eigenvalue of the deflated matrix is $\lambda_{n-1} \geq \lambda_n$. There is a second type of deflation. In the QR algorithm for Hessenberg matrices, this occurs if a subdiagonal element becomes small enough. In our case there is no Hessenberg or band structure, so this deflation occurs if the submatrix $M(j + 1 : n, 1 : j)$ is zero. Then we can divide the matrix into two smaller matrices. The spectrum of $M_m$ is the union of the spectrum of the two smaller matrices $M(1 : j, 1 : j)$ and $M(j + 1 : n, j + 1 : n)$.

Table 1 shows the needed CPU-time if we use the LR-Cholesky transformation together with the described shift and deflation strategies for four $\mathcal{H}$-matrices of different size. This matrices are the finite-element discretisation of the 2D-Laplacian with 8 to 64 discretisation points in each direction. We see that the CPU-time grows faster than $O(n^2 \log^2 n)$.

The hierarchical semi-separable matrices (HSS matrices, [2]) are a subset of $\mathcal{H}$-matrices. The $\mathcal{H}$-tree of an HSS matrices have only admissible knots or knots of the form $s \times s$. Further the admissible blocks satisfy the condition, that $A_{t \times s}$ is in the linear span of $[0; A_{s \times t_1}]$, with $s < t$ and $t = t_1 \cup t_2$. The LR-Cholesky transformation for HSS matrices produce no $\mathcal{H}$-fill-in.

For practical purposes eigenvalue algorithms for computing a small subset of the spectrum are interesting, too. There is a well known $\mathcal{H}$-matrix-vector product of linear-polylogarithmic complexity. This product is sufficient to implement the power method:

$$y_k = Ax_k$$
$$x_k = \frac{y_k}{\|y_k\|_2},$$

or the Jacobi-Davidson algorithm [3]. As in standard arithmetic, the Jacobi-Davidson algorithm converges faster than the power method and permits the computation of inner eigenvalues. This is current work.

**Acknowledgments.** The work of Thomas Mach is supported by a grant of the Free State of Saxony (501-G-209).
REFERENCES


Mathematik in Industrie und Technik, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, Germany; \{benner,thomas.mach\}@mathematik.tu-chemnitz.de.