Iterative Refinement With Hierarchical Low-Rank Preconditioners Using Mixed Precision

Thomas Spendlhofer[1] and Rio Yokota[2]
[1]Dept. of Computer Science, Tokyo Institute of Technology, Japan; [2] Global Scientific Information Center, Tokyo Institute of Technology, Japan

1 Abstract

The solution to a dense linear system can be accelerated by using mixed precision iterative refinement (IR) relying on an approximate LU-factorization. Such a factorization is usually obtained by simply reducing the employed floating-point precision, but certain matrices allow for alternative approaches such as low-rank approximations. For example, matrices arising from the boundary element method (BEM) are known to contain a *block low-rank* structure, where certain blocks of the matrix can be replaced by low-rank approximations. We investigate the combination of both techniques (reduced precision and low-rank approximations) for dense matrices with sufficient structure and give indications on how the respective parameters are impacting performance of the IR process.

2 Iterative Refinement (IR)

Alg. 1 outlines the general iterative refinement process using four precisions (denoted in terms of the unit round-off), where u_s is essentially a parameter on the method used to solve the correction equation in line 10.

Algorithm 1: Iterative Refinement

Input: $A \in \mathbb{R}^{n\times n}$; $b \in \mathbb{R}^n$; $k_{max} \in \mathbb{N}$; $\epsilon_{max} \in \mathbb{R}$

Output: approximate solution $\hat{x} \in \mathbb{R}^n$ to $Ax = b$

1 solve $Ax_0 = b$ in precision u_f
2 solve $LUx_0 = b$ in precision u_f
3 store x_0 at precision u
4 **for** $k = 1$ **to** k_{max} **do**
5 compute $r_k = b - Ax_{k-1}$ in precision u_r
6 round r_k to precision u
7 **if** $\|r_k\| \leq \epsilon_{max}$ **then**
8 **return** x_{k-1}
9 **end**
10 solve $Ad_k = r_k$ in precision u_s
11 store d_k at precision u
12 $x_k = x_{k-1} + d_k$ in precision u
13 **end**
14 % iteration has not converged

Generally, the precisions can be chosen freely, as long as the order $u_r \leq u \leq u_s \leq u_f$ is maintained.

3 Hierarchical Off-Diagonal Low-Rank (HODLR) Matrix

In order to exploit the block row-rank property of a matrix, a suitable partition P (also referred to as the *cluster tree*) of (sub-)blocks needs to be found. Depending on the size, location and hierarchy of these blocks, several different formats of varying algorithmic complexity can be distinguished. In this work, we used a so called *weak admissibility* condition (dense blocks only along the diagonal) and employed the HODLR format (see Fig. 1), which is able to achieve both LU-factorization and matrix-vector multiplication in $\mathcal{O}(n\ log\ n)$.

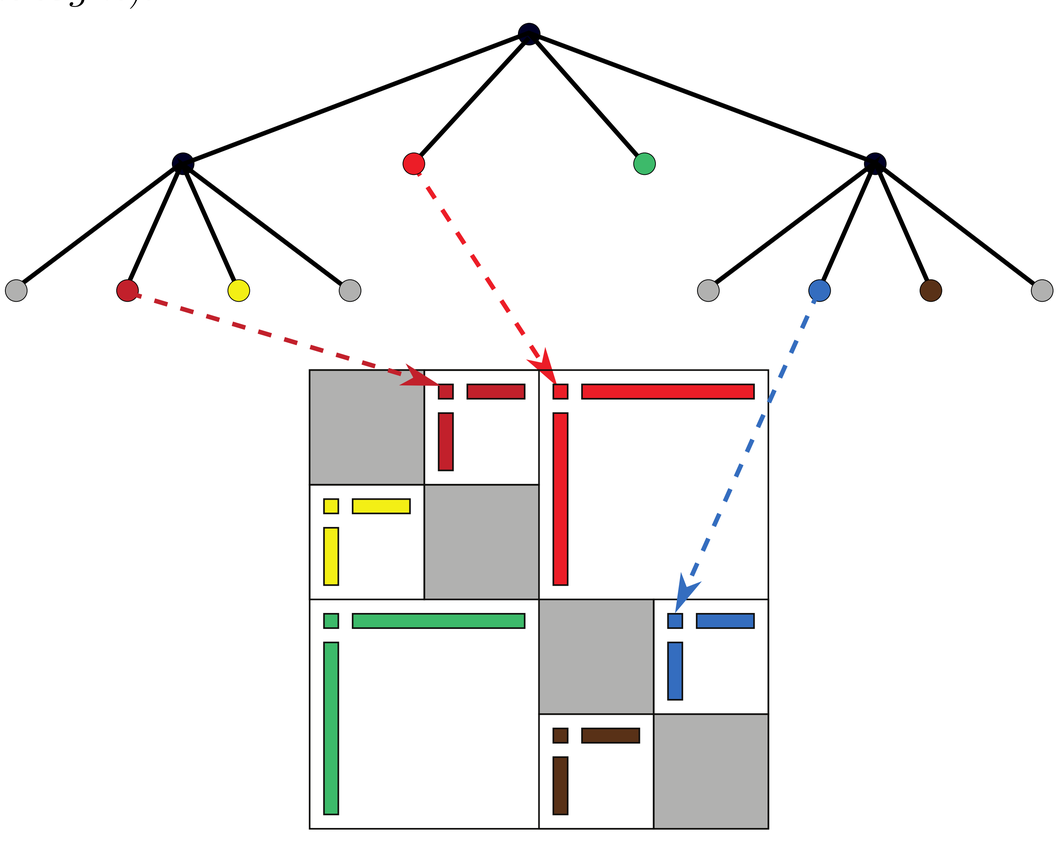

Figure 1: HODLR matrix with corresponding cluster tree.

4 Results

We evaluate the combination of HODLR matrices with IR on a set of kernel functions (Laplace, Gaussian and IMQ) and geometries (unit circle/square) while adjusting the condition numbers by adding small values on the diagonal ($A = A + \sigma I$), leading to a range of $\kappa_\infty(A) \approx 10^2 - 10^{12}$. For each matrix several HODLR approximations were created with a specified upper error bound $\epsilon \in \{10^{-2}, 10^{-4}.10^{-6}.10^{-8}.10^{-12}\}$. Those matrices were then factorized and the resulting LU factors used as input for the IR process. Time was measured as time-to-solution (i.e. including the HODLR construction) on an AMD Ryzen Threadripper 3960X processor featuring 24 cores. Multi-threading was enabled via Intel MKL (for the dense parts) or OpenMP tasks (for the HODLR parts) and results are displayed in Fig. 2.

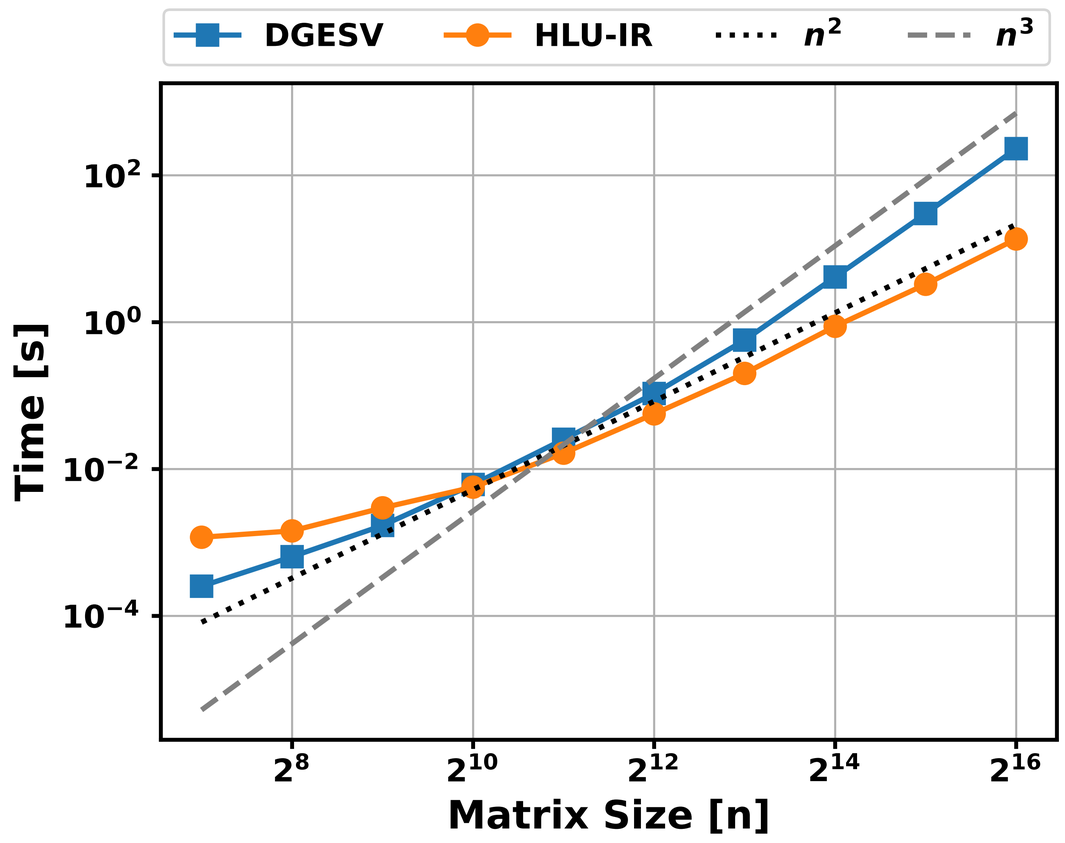

Figure 2: Time-to-solution of a dense solver and HODLR-based IR

While a higher approximation error is associated with faster HODLR calculation, it might also require a larger number of iterations before convergence. As can be seen from Fig. 3, best performance was observed for an approximation error that does not exceed the reciprocal of the condition number, i.e. $\epsilon \leq \kappa_\infty(A)^{-1}$.

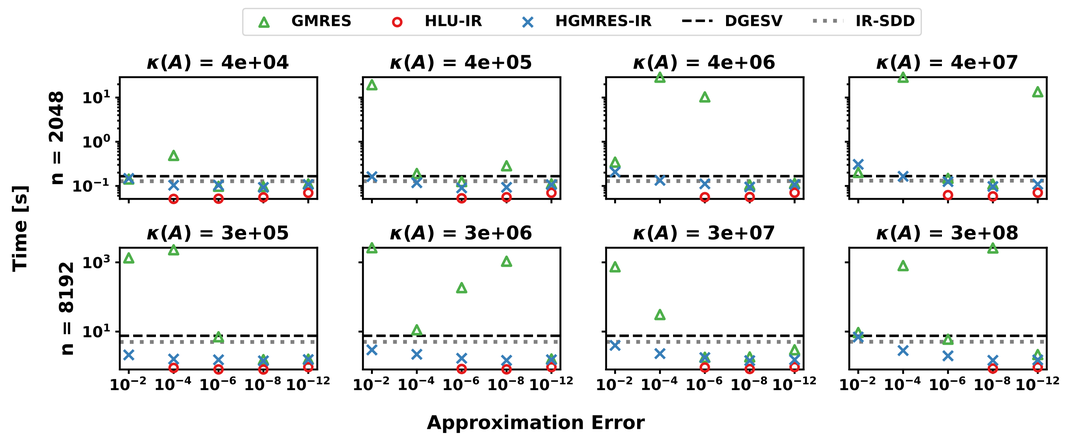

Figure 3: Computation time as a function of the approximation error.

5 Conclusion

We demonstrate that for matrices with a sufficient structure a solution accurate to a direct solver in double precision can be achieved at a reduced complexity of $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$. This difference in scaling resulted in a speedup of more than $16X$ on the largest tested matrix size. Similar to how a result accurate to double precision can be refined from a single precision factorization, the method is able to take advantage of mixed precision calculations to either reduce the computational time or enhance the accuracy of the result even further.

6 Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP20K20624, JP21H03447, JP22H03598. This work is supported by JST CREST Grant Number JPMJCR2112. This work is supported by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" in Japan (Project ID: jh230053-NAH, jh230009-NAHI).