# Application of Hierarchical Matrices For Solving Multiscale Problems

Dissertation, Universität Leipzig, 2006
Litvinenko Alexander

## Abstract

In this work we combine hierarchical matrix techniques (Hackbusch, 1999) and domain decomposition methods to obtain fast and efficient algorithms for the solution of multiscale problems. This combination results in the hierarchical domain decomposition (HDD) method.

*Multiscale problems* are problems that require the use of different length scales. Using only the finest scale is very expensive, if not impossible, in computational time and memory.

A *hierarchical matrix* $M \in \mathbb{R}^{n\times m}$ (which we refer to as an $\mathcal{H}$-matrix) is a matrix which consists mostly of low-rank subblocks with a maximal rank $k$, where $k \ll \min\{n,m\}$. Such matrices require only $\mathcal{O}(kn\log n)$ (w.l.o.g. $n \geq m$) units of memory. The complexity of all arithmetic operations with $\mathcal{H}$-matrices is $\mathcal{O}(k^\alpha n \log^\alpha n)$, where $\alpha = 1,2,3$. The accuracy of the $\mathcal{H}$-matrix approximation depends on the rank $k$.

*Domain decomposition methods* decompose the complete problem into smaller systems of equations corresponding to boundary value problems in subdomains. Then fast solvers can be applied to each subdomain. Subproblems in subdomains are independent, much smaller and require less computational resources as the initial problem.

We consider the elliptic boundary value problem with $L^\infty$ coefficients and with Dirichlet boundary condition:

$$\begin{cases} -\sum\limits_{i,j=1}^{d} \frac{\partial}{\partial_j}\alpha_{ij}\frac{\partial}{\partial_i}u = f & \text{in } \Omega \subset \mathbb{R}^d, \quad d = 2,3, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{1}$$

with $\alpha_{ij} = \alpha_{ji}(\mathbf{x}) \in L^\infty(\Omega)$ such that the matrix function $\mathcal{A}(\mathbf{x}) = (\alpha_{ij})_{i,j=1,\ldots,d}$ satisfies $0 < \underline{\lambda} \leq \lambda_{\min}(\mathcal{A}(\mathbf{x})) \leq \lambda_{\max}(\mathcal{A}(\mathbf{x})) \leq \overline{\lambda}$ for all $\mathbf{x} \in \Omega$. This setting allows us to treat oscillatory as well as jumping coefficients.

This equation can represent incompressible single-phase porous media flow or steady state heat conduction through a composite material. In the single-phase flow, $u$ is the flow potential and $\alpha$ is the permeability of the porous medium. For heat conduction in composite materials, $u$ is the temperature, $q = -\alpha\nabla u$ is the heat flow density and $\alpha$ is the thermal conductivity.

HDD computes two discrete hierarchical solution operators $\mathcal{F}_h$ and $\mathcal{G}_h$ such that:

$$u_h = \mathcal{F}_h f_h + \mathcal{G}_h g_h, \tag{2}$$

where $u_h(f_h, g_h)$ is the FE solution of (1), $f_h$ the FE right-hand side, and $g_h$ the FE Dirichlet boundary data. Both operators $\mathcal{F}_h$ and $\mathcal{G}_h$ are approximated by $\mathcal{H}$-matrices. Let $n_h$ and $n_H$ be the numbers of degrees of freedom on a fine grid and on a coarse grid. The complexities of the one-grid version and two-grid version of HDD are

$$\mathcal{O}(n_h \log^3 n_h) \quad \text{and} \quad \mathcal{O}(\sqrt{n_h n_H}\log^3\sqrt{n_h n_H}), \quad \text{respectively.}$$

The storage requirements of the one-grid version and two-grid version of HDD are

$$\mathcal{O}(n_h \log^2 n_h) \quad \text{and} \quad \mathcal{O}(\sqrt{n_h n_H}\log^2\sqrt{n_h n_H}), \quad \text{respectively.}$$