Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Data Sparse Numerical Models for SNOM Tips

C. Hafner and R. Hiptmair and P. Souzangar

Research Report No. 2015-26

August 2015
Latest revision: March 2016

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Funding: ETH Zurich under CHIRP Grant1-01 11-1

Report, Seminar For Applied Mathematics, ETH Zurich
SAM Report 2015

Data Sparse Numerical Models for SNOM Tips

Christian Hafner[1], Ralf Hiptmair [2], Pegah Souzangar*[1]

[1] *Institute of Electromagnetic Fields, ETH Zurich, CH-8092 Zurich, Switzerland*
[2] *Seminar for Applied Mathematics, ETH Zurich, CH-8092 Zurich, Switzerland*

SUMMARY

We propose textcolorblacka compressed non-local surface impedance type boundary condition for the efficient numerical modeling of textcolorblacklarge geometrically persistent parts in textcolorblackmulti-scale electromagnetic simulations. textcolorblackThe underlying compressed model is an approximate Schur complement of Finite Element Galerkin matrix. Our approach relies on local low-rank representation in the framework of the $\mathcal{H}$-matrix storage format. We discuss two ways to build $\mathcal{H}$-matrix approximations of Schur-complement matrices: adaptive cross approximation (ACA) and $\mathcal{H}$-arithmetics. textcolorblackProfound numerical tests and studies of accuracy are carried out for an axisymmetric setting, employing the open source library AHMED by M. Bebendorf. textcolorblackTo demonstrate the use of our method we build the compressed model for an axisymmetric SNOM tip. The model can be embedded in three dimensional tip-sample simulations.

Date: March 7, 2016

KEY WORDS: Maxwell's equations, finite elements, matrix compression, H-matrices, ACA, Schur complement

1. INTRODUCTION

1.1. SNOM tips

Scanning Near-field Optical Microscopy, SNOM, first introduced by Pohl et al. [21], is an optical technique for imaging samples of sub-wavelength size. Some common configurations of aperture and apertureless SNOM tips are illustrated in Figure 1. In all of these configurations, the tip moves over the sample and collects data from different locations in the vicinity of the sample with the aim of achieving a comprehensive image.

In this report, we are interested in solving the forward problem of tip-sample analysis, that is computation of the electromagnetic fields for a given tip-sample geometry with known material and excitation. For this purpose, we rely on linear time-harmonic Maxwell equations and adopt the Finite Element Method (FEM), as this numerical method can easily cope with the typical problems arising from complex geometries and frequency-dependent materials.

textcolorblackIn numerical tip-sample simulations, the tip textcolorblackis several wavelengths long and orders of magnitudes larger than the sample being imaged. This intrinsic property of the SNOM imaging simulations represents major challenges for numerical methods such as FEM. Since capturing subwavelength phenomena entails high-resolution discretization, employing these numerical methods involves many degrees of freedom (DoFs), and consequently demands for huge

*Correspondence to: Ramistrasse 101, CH-8092 Zurich, Switzerland. E-mail: pegahs@ifh.ee.ethz.ch

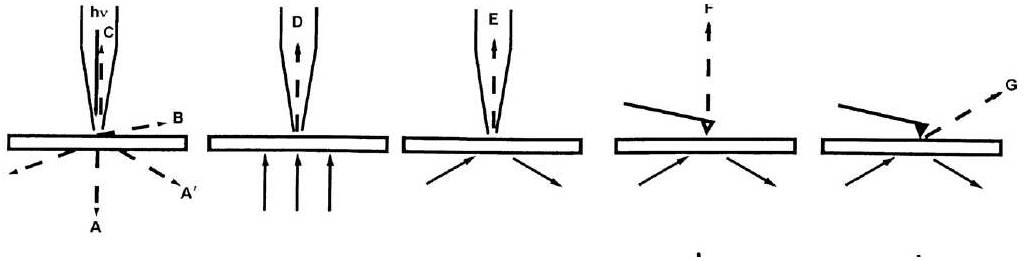


Figure 1. The various principles of SNOM techniques; aperture SNOM (A, A', B, C) with metal-coated light sources, D and E with separate source; apertureless SNOM by light tunneling (F, G), [14, Figure. 2.1]

memory resources. This leads to cumbersome computations which are hardly feasible on current computers. Furthermore, to attain a comprehensive image, the tip has to collect data from many points all around the sample. For each new position of the tip with respect to the sample a new simulation has to be carried out.

To tackle the aforementioned difficulties, we notice that the shape of the tip and its material composition remain unchanged every time the tip is moved to a new point. Clearly, the difference between different simulation runs is only in the relative location of the tip with respect to the sample. Even though the tip occupies the main portion of the computational domain and needs to be represented accurately, it remains an unchanged part for many simulations. In brief, the tip is a big and constant part of imaging simulations.

In this report, we aim to compute a numerical tip model through *offline computations* only once such that textcolorblackit can be reused as a plug-in for many imaging simulations. Assuming that the tip is rotationally symmetric, as explained in Section 2, we exploit the symmetry to model the three dimensional tip on a two dimensional mesh. Afterwards, the volume of the tip and some surrounding space is replaced by a discretized electric to magnetic field map which represents the influence of the tip in the simulation domain through a non-local surface-impedance boundary condition. This discretized map is an approximation of the Schur complement of the finite element matrix as explained in Section 2.4.

1.2. Matrix compression based on hierarchical matrices

Since the tip should be discretized with high resolution, the model matrix, i.e., the Schur complement of the FE matrix is generally huge and naturally dense. textcolorblackTo avoid high memory consumption, we calculate a local low-rank approximation of the matrix in $\mathcal{H}$-matrix format [8, 10, 11]. The idea is supported by [3, Theorem 4.31] which states that the Schur complement of a textcolorblackFE Gelerkin matrix can be well approximated by an $\mathcal{H}$-matrix.

textcolorblackSo far, $\mathcal{H}$-matrices have been used as preconditioners e.g. [5, 19, 22] and direct solvers in electromagnetics, e.g. [17, 23, 24]. In this report we demonstrate how $\mathcal{H}$-matrices can be used to efficiently store accuracy-controlled approximations of tip models.

textcolorblackWe assume that the reader is familiar with basic concepts of $\mathcal{H}$-matrix techniques such as cluster tree [3, Definition 1.16], block cluster tree [3, Sec. 1.5], admissibility condition [3, Sect. 1.3] and admissible partitioning [3, Definition 1.15]. For introduction and details we refer to the monographs [3, 7, 12].

1.3. Objective of the manuscript

We have investigated the efficiency of two different methods for model matrix compression for an axisymmetric tip:

(I) Computation of the Schur complement through sparse solvers and application of the Adaptive Cross Approximation (ACA) algorithm to build an $\mathcal{H}$-matrix approximation of the model matrix (see Section 4).

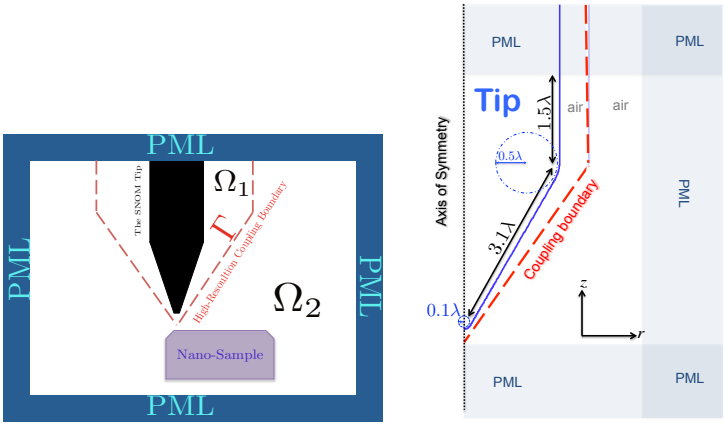


Figure 2. a) Homogeneous domain decomposition for tip-sample configurations. Ω_1 is the fixed part of imaging simulations. b) Metallic tip model.

(II) Conversion of sparse FE matrices to $\mathcal{H}$-matrices and use of arithmetic operations in $\mathcal{H}$-matrix format ($\mathcal{H}$-arithmetics) to obtain the Schur complement directly in $\mathcal{H}$-matrix format (see Section 5).

The computations with $\mathcal{H}$-matrices are performed using algorithms provided by the AHMED C++ class library developed by M. Bebendorf [2]. For each approach, we monitor the cost of generating the compressed tip model versus the predefined accuracy. For a high resolution FE model matrix, we investigate the efficiency of compression in terms of memory needed to store the discrete tip model for different accuracies. A comparison of the two approaches and some concluding remarks are presented in Section 6.

2. NUMERICAL MODELING OF A SNOM TIP

The idea of homogeneous domain decomposition is used to model the tip and its surroundings separately. As shown in Figure 2a, we truncate the computational domain by means of PML, then decompose it into two disjoint subdomains, Ω_1 and Ω_2, such that,

$$\Gamma = \partial\bar{\Omega}_1 \cap \partial\bar{\Omega}_2 \tag{1a}$$

$$\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma. \tag{1b}$$

The inner subdomain Ω_1 is the domain that encompasses the tip and its surroundings. The outer subdomain Ω_2 includes the sample and is the domain of interest in which the fields need to be computed. This subdomain varies for different samples. The interface boundary, Γ, has only a mathematical meaning and we choose it to have a regular shape which is a cone in the lower part.

As mentioned earlier, the subdomain Ω_1 remains unchanged in many imaging simulations. In a preprocessing step, we compute a discretized interface operator on Γ by calculating the fields in Ω_1. The interface operator takes the transmission boundary condition on Γ into account and substitutes Ω_1 in imaging simulations. Using this operator, the fields have to be computed only in Ω_2.

2.1. *Maxwell equations*

For the FE analysis of the tip-sample setup, we assume that the time-harmonic Maxwell's equations in a non-magnetic isotropic medium hold,

$$\nabla \times \mathbf{E} = i\omega\mu\mathbf{H} \tag{2a}$$
$$\nabla \times \mathbf{H} = -i\omega\varepsilon\mathbf{E}, \tag{2b}$$

where μ and ϵ are the spatialy varying permeability and permittivity functions and ω is the angular frequency.

We write the equations (2) only in terms of the electric field $\mathbf{E}$. After splitting the domain into Ω_1 and Ω_2, we solve the following problem: we seek $\mathbf{E_1} \in H(\mathbf{curl};\Omega_1)$, $\mathbf{E_2} \in H(\mathbf{curl};\Omega_2)$, such that for all $\mathbf{E}' \in H(\mathbf{curl};\Omega), \mathbf{F_s} \in L_2(\Omega)$ (see [18, Section 4.1]) ,

$$\begin{aligned}&\int_{\Omega_1} \langle \nabla\times\mathbf{E_1}, \nabla\times\mathbf{E}'\rangle - k^2\langle \mathbf{E_1}, \mathbf{E}'\rangle d\Omega \\ &+ \int_{\partial\Omega_1\setminus\Gamma} \langle \boldsymbol{\nu}\times(\nabla\times\mathbf{E_1}), \mathbf{E}'_{\mathbf{T}}\rangle dA + \int_{\Gamma} \langle \boldsymbol{\nu}\times(\nabla\times\mathbf{E_1}), \mathbf{E}'_{\mathbf{T}}\rangle dA = \int_{\Omega_1} \langle \mathbf{F_s}, \mathbf{E}'\rangle d\Omega\end{aligned} \tag{3a}$$

$$\begin{aligned}&\int_{\Omega_2} \langle \nabla\times\mathbf{E_2}, \nabla\times\mathbf{E}'\rangle - k^2\langle \mathbf{E_2}, \mathbf{E}'\rangle d\Omega \\ &+ \int_{\partial\Omega_2\setminus\Gamma} \langle \boldsymbol{\nu}\times(\nabla\times\mathbf{E_2}), \mathbf{E}'_{\mathbf{T}}\rangle dA - \int_{\Gamma} \langle \boldsymbol{\nu}\times(\nabla\times\mathbf{E_2}), \mathbf{E}'_{\mathbf{T}}\rangle dA = \int_{\Omega_2} \langle \mathbf{F_s}, \mathbf{E}'\rangle d\Omega,\end{aligned} \tag{3b}$$

in combination with the following transmission conditions on Γ,

$$\mathbf{E_1}\times\boldsymbol{\nu} = \mathbf{E_2}\times\boldsymbol{\nu} \qquad \text{on } \Gamma \tag{4a}$$
$$\nabla\times\mathbf{E_1}\times\boldsymbol{\nu} = \nabla\times\mathbf{E_2}\times\boldsymbol{\nu} \qquad \text{on } \Gamma, \tag{4b}$$

where $k = \omega\sqrt{\mu\varepsilon}$, $\boldsymbol{\nu}$ is the normal unit vector on $\partial\Omega$ and Γ that is pointing outward Ω_1 on Γ, $\mathbf{E}'_{\mathbf{T}} = (\boldsymbol{\nu}\times\mathbf{E}')\times\boldsymbol{\nu}$ and $\mathbf{F_s}$ is the source term. Note that the the coupling boundary is entirely located in air region.

In the following subsection, we review the axisymmetric formulation for the electric field in Ω_1, and consequently compute the descretized interface operator on Γ.

2.2. *Axisymmetric formulations*

textcolorblackWe use cylindrical coordinates system in the tip domain, Ω_1. The coordinates are denoted by (r,ϕ,z), and $\hat{\mathbf{r}}, \hat{\boldsymbol{\phi}}, \hat{\mathbf{z}}$ are the respective unit vectors. In the rest of the report, the axis of rotation is along $\hat{\mathbf{z}}$. textcolorblack We expand the electric field in Ω_1 into a Fourier series with respect to the azimuthal angle ϕ,

$$\begin{aligned}\mathbf{E_1}(r,\phi,z) = &\sum_{m=0}^{\infty} \mathbf{E}^{\mathbf{m}}_{\mathbf{p}}(r,z)\cos(m\phi) + E^m_{\phi}(r,z)\sin(m\phi)\hat{\boldsymbol{\phi}} \\ &+ \sum_{m=0}^{\infty} -\mathbf{E}^{\mathbf{m}}_{\mathbf{p}'}(r,z)\sin(m\phi) + E^m_{\phi'}(r,z)\cos(m\phi)\hat{\boldsymbol{\phi}}.\end{aligned} \tag{5}$$

where $E^m_{\phi}(r,z)$ and $E^m_{\phi'}$ are scalar functions and $\mathbf{E}^{\mathbf{m}}_{\mathbf{p}}(r,z) = E^m_r\hat{\mathbf{r}} + E^m_z\hat{\mathbf{z}}$ and $\mathbf{E}^{\mathbf{m}}_{\mathbf{p}'}(r,z) = E^m_{r'}\hat{\mathbf{r}} + E^m_{z'}\hat{\mathbf{z}}$ are vector fields [15, 16]. It can be shown that the two parts of the Fourier expansion in (5) are decoupled in Ω_1 and the same variantional equation governs both. Only the outer domain and the Fourier expansion of the excitation field determine which part will matter. Therefore, for our model problem presented in Section 3 we omit the second term in (5) and refer to the mth mode in the Fourier series by $\mathbf{E^m} = \mathbf{E}^{\mathbf{m}}_{\mathbf{p}}\cos(m\phi) + E^m_{\phi}\sin(m\phi)\hat{\boldsymbol{\phi}}$. So,

$$\mathbf{E_1}(r,\phi,z) = \sum_{m=0}^{\infty} \mathbf{E^m}. \tag{6}$$

Thanks to the rotational symmetry of the tip, the textcolorblackFourier modes in Ω_1 are fully decoupled. Therefore, we solve (3a) only for a few modes in the Fourier series, assuming that a good approximation of the total field $\mathbf{E_1}$ is attained by truncating (6) to a short finite sum.

For each mode in the Fourier series, the ϕ dependence of the field is known analytically, which means that one can simulate the fields only in a cross section of the tip in textcolorblacka two dimensional domain Ω_p where $\Omega_1 = \Omega_p \times [0, 2\pi]$. To do this, we reformulate (3a) for the axisymmetric case.

First, we define the following planar operators [16],

$$\nabla_p V \coloneqq \partial_r V \hat{\mathbf{r}} + \partial_z V \hat{\mathbf{z}} \tag{7a}$$

$$\nabla_p \times \mathbf{V_p} \coloneqq (\partial_z V_r - \partial_r V_z)\hat{\boldsymbol{\phi}}, \tag{7b}$$

where ∂_τ denotes the partial derivative with respect to τ, $V(r, z)$ is a scalar function and $\mathbf{V_p}(r, z) = V_r\hat{\mathbf{r}} + V_z\hat{\mathbf{z}}$ is a textcolorblacktwo dimensional vector field. One can decompose any vector $\mathbf{V}$ into planar and azimuthal components: $\mathbf{V} = \mathbf{V_p} + V_\phi\hat{\boldsymbol{\phi}}$. The vector operators in cylindrical coordinates are written as follows,

$$\nabla V = \frac{1}{r}\partial_\phi V \hat{\boldsymbol{\phi}} + \nabla_p V \tag{8a}$$

$$\nabla \times \mathbf{V} = \nabla_p \times \mathbf{V_p} + \frac{1}{r}\mathcal{P}(\partial_\phi \mathbf{V_p} - \nabla_p(rV_\phi)), \tag{8b}$$

textcolorblackwhere $\mathcal{P}$ is an operator mapping the vector field $\mathbf{V} = V_r\hat{\boldsymbol{r}} + V_\phi\hat{\boldsymbol{\phi}} + V_z\hat{\boldsymbol{z}}$ according to $\mathcal{P}(\mathbf{V}) \coloneqq V_z\hat{\boldsymbol{r}} - V_r\hat{\boldsymbol{z}}$.

We rewrite (3a) in terms of $\mathbf{E^m}$ and the vector operators in (8). In addition, we define $q_\phi^m \coloneqq rE_\phi^m$ and solve the equations for this new unknown, see [16]. Assuming $\mathbf{F_s} = 0$, solving (3a) for $\mathbf{E^m}$ boils down to the following problem:

We seek,

$$\mathbf{E_p^m}, q_\phi^m \in L^2(\Omega_p)$$

such that

$$\mathbf{E_p^m}\cos(m\phi) + \frac{q_\phi^m}{r}\sin(m\phi)\hat{\boldsymbol{\phi}} \in H(\mathbf{curl}, \Omega_1)$$

and,

$$\begin{aligned}
&\int_{\Omega_p} \frac{1}{r}\langle m\mathbf{E_p^m} + \nabla_p q_\phi^m, m\mathbf{E_p'^m} + \nabla_p q_\phi'^m\rangle \mathrm{d}\Omega_p + \int_{\Omega_p} r\langle \nabla_p \times \mathbf{E_p^m}, \nabla_p \times \mathbf{E_p'^m}\rangle \mathrm{d}\Omega_p \\
&- k^2 \int_{\Omega_p} r\langle \mathbf{E_p^m}, \mathbf{E_p'^m}\rangle + \frac{1}{r}\langle q_\phi, q_\phi'\rangle \mathrm{d}\Omega_p \\
&+ \int_{\partial\Omega_p} \langle \boldsymbol{\nu} \times (\nabla_p \times \mathbf{E_p^m} + \frac{1}{r}\mathcal{P}(m\mathbf{E_p^m} - \nabla_p q_\phi^m)), (\mathbf{E_p'^m} + \frac{q_\phi'^m}{r}\hat{\boldsymbol{\phi}})_\mathbf{T}\rangle \mathrm{d}l = \mathbf{0},
\end{aligned} \tag{9}$$

$$\forall \mathbf{E_p'^m}, q_\phi'^m : \mathbf{E_p'^m}\cos(m\phi) + \frac{q_\phi'^m}{r}\sin(m\phi)\hat{\boldsymbol{\phi}} \in H(\mathbf{curl}, \Omega_1).$$

By reducing the computational domain from Ω_1 to Ω_p, the axis of rotation which is inside Ω_1 coincides with a part of $\partial\Omega_p$. Therefore, a new boundary condition is introduced on the axis of rotation. To obtain the boundary condition on the axis, we notice that $\mathbf{E^m} \in H(\mathbf{curl}, \Omega_1)$. This involves the following condition on the axis,

$$m\mathbf{E_p^m}(0, z) + \nabla_p q_\phi^m(0, z) = 0. \tag{10}$$

From the above condition, one can show that the boundary condition on the axis for the tangential component of $\mathbf{E_p}$, i.e., E_z, is the following:

$$\begin{cases} m > 0: & \text{homogeneous Dirichlet boundary condition,} \\ m = 0: & \text{homogeneous Neumann boundary condition.} \end{cases} \tag{11}$$

q_ϕ is zero on the axis by definition (homogeneous dirichlet boundary condition).

2.3. *Finite element Galerkin discretization*

In this report, we use the first order polynomial conforming FEM on a triangular mesh in $\Omega_p \in \mathbb{R}^2$. The FEM basis functions are chosen according to [16]. To expand q_ϕ local piecewise linear nodal functions (also known as hat functions) are used. textcolorblackThe first order two dimensional edge elements [9] are chosen for $\mathbf{E_p}$.

The FE matrices are obtained based on the bilinear form in (9). The final FE system of equations is,

$$\begin{bmatrix} \mathbf{A_{ee}} & \mathbf{A_{ev}} \\ \mathbf{A_{ve}} & \mathbf{A_{vv}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu_e} \\ \boldsymbol{\mu_v} \end{bmatrix} = \begin{bmatrix} \mathbf{f_e} \\ \mathbf{f_v} \end{bmatrix}, \tag{12}$$

where $\boldsymbol{\mu_e}$ and $\boldsymbol{\mu_v}$ are the column vectors of FE DoFs on the edges and vertices, respectively.

2.4. *Schur complement of the FE matrix*

We aim to obtain a discretized analogue of the Dirichlet-to-Neumann map on Γ in Figure 2a to substitute the computational domain Ω_1. In the case of three dimensional electromagnetic fields the electric to magnetic field operator $\mathcal{C}$ is defined [18, Section 9.4] textcolorblack by $\mathbf{E_T} \mapsto \boldsymbol{\nu} \times \mathbf{H}$, where $\mathbf{H}$ solves (2) in Ω_1 with the boundary value given by $\mathbf{E_T}$, i.e., the tangential component of $\mathbf{E}$ on Γ.

On the discrete level, $\mathcal{C}$ corresponds to the Schur complement of the FEM matrix, which is obtained by decomposing the DoFs into those located in the interior domain, Ω_1 exterior domain Ω_2 and DoFs on the boundary (Γ in Figure 2a):

$$\begin{bmatrix} \mathbf{A_{in,in}} & \mathbf{A_{in,\Gamma}} & \mathbf{0} \\ \mathbf{A_{\Gamma,in}} & \mathbf{A^{in}_{\Gamma,\Gamma}} + \mathbf{A^{out}_{\Gamma,\Gamma}} & \mathbf{A_{\Gamma,out}} \\ \mathbf{0} & \mathbf{A_{out,\Gamma}} & \mathbf{A_{out,out}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu_{in}} \\ \boldsymbol{\mu_\Gamma} \\ \boldsymbol{\mu_{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{f_{in}} \\ \mathbf{f_\Gamma} \\ \mathbf{f_{out}} \end{bmatrix}, \tag{13}$$

where $\boldsymbol{\mu_{in}}$, $\boldsymbol{\mu_\Gamma}$, and $\boldsymbol{\mu_{out}}$ are the FE DoFs in Ω_1, on Γ and in Ω_2, respectively. Since we are interested in the outer domain, Ω_2, the system of equations is written exclusively for this domain. Assuming $\mathbf{f_\Gamma} = 0$,

$$\begin{bmatrix} \mathbf{A^{out}_{\Gamma,\Gamma}} + \mathbf{S} & \mathbf{A_{\Gamma,out}} \\ \mathbf{A_{out,\Gamma}} & \mathbf{A_{out,out}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu_\Gamma} \\ \boldsymbol{\mu_{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{f_{tip}} \\ \mathbf{f_{out}} \end{bmatrix}, \tag{14}$$

where $\mathbf{S}$ is the Schur complement matrix and $\mathbf{f_{tip}}$ represents the excitation in the tip domain, Ω_1,

$$\mathbf{S} = -\mathbf{A_{\Gamma,in}}\mathbf{A_{in,in}}^{-1}\mathbf{A_{in,\Gamma}} + \mathbf{A^{in}_{\Gamma,\Gamma}} \tag{15a}$$

$$\mathbf{f_{tip}} = -\mathbf{A_{\Gamma,in}}\mathbf{A_{in,in}}^{-1}\mathbf{f_{in}}. \tag{15b}$$

We rely on $\mathbf{S}$ to represent the tip model matrix. This matrix may be huge and dense and also we need to compute it for several modes in (6). Therefore, we want to represent $\mathbf{S}$ in the data sparse $\mathcal{H}$-matrix-format, see Section 4 and Section 5.

textcolorblackFrom [3, Theorem 4.31] together with [3, Theorem 4.28], we know that the $\mathcal{H}$-matrix approximation of $\mathbf{S}$ exists and the maximum rank of its blocks behaves like $O(|\log(\varepsilon)|^3 \log^2 N_\Gamma)$ where ε is a relative error of matrix approximation. The numerical approximation methods in Section 4 and Section 5 do not guarantee that the optimum rank for the blocks is found. So, we assume that in practice the maximum rank of the blocks may have the complexity $O(\log^3 N_\Gamma)$ for a fixed ε. Taking [3, Theorem 2.6] into account, we anticipate a storage complexity better than $O(N_\Gamma \log^4 N_\Gamma)$ to save the $\mathcal{H}$-matrix approximation of $\mathbf{S}$.

textcolorblack

2.5. *Solving the linear system of equations*

textcolorblackAs $\mathbf{S}$ is dense, textcolorblack direct elimination techniques cannot be used to solve (14) and iterative solvers are the only option. The cost of a single iteration will mainly hinge on the

effort required for matrix-vector multiplication. textcolorblackTo study this we split the matrix in (14) into two parts:

$$\begin{bmatrix} \mathbf{A}^{\mathbf{out}}_{\mathbf{\Gamma},\mathbf{\Gamma}} + \mathbf{S} & \mathbf{A}_{\mathbf{\Gamma},\mathbf{out}} \\ \mathbf{A}_{\mathbf{out},\mathbf{\Gamma}} & \mathbf{A}_{\mathbf{out},\mathbf{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{\mathbf{out}}_{\mathbf{\Gamma},\mathbf{\Gamma}} & \mathbf{A}_{\mathbf{\Gamma},\mathbf{out}} \\ \mathbf{A}_{\mathbf{out},\mathbf{\Gamma}} & \mathbf{A}_{\mathbf{out},\mathbf{out}} \end{bmatrix} + \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{16}$$

textcolorblack The first term in the right hand side of (16) is due to the outer domain and is independent of the tip model. textcolorblack If a data-sparse matrix is saved in $\mathcal{H}$-matrix format, the complexity of matrix-vector multiplication is almost linear[†]. textcolorblack Thus the use of a compressed tip model will make a contribution to the total effort of a single matrix-vector multiplication that scales merely almost linearly in the number of DoFs on the tip boundary, if $\mathbf{S}$ is available in $\mathcal{H}$-matrix format.

textcolorblackThe (approximate) computation of $\mathbf{f_{tip}}$ in (15b) can be carried out in advance together with the precomputation $\mathbf{S}$ in (15a). Note that $A^{-1}_{in,in}$ becomes available during the computation of $\mathbf{S}$ as detailed in sections 4 and 5. Therefore, $\mathbf{f_{tip}}$ in (15b) virtually comes for free, even when there is excitation through the tip.

3. A MODEL PROBLEM

For all numerical experiments in sections 4 and 5, we consider an axisymmetric metallic tip with permitivity $\varepsilon_r = -10.155 + 2.1281e-08i$ placed in air and illuminated by atextcolorblackn electrical dipole located on the axis of symmetry. The cross section of the tip is shown in Figure 2b. The apex radius textcolorblack(radius of the curvature of the tip apex) and shaft radius textcolorblack(radius of the junction curvature of the conical and cylindrical part) are $\lambda/10$ and $\lambda/2$ respectively, textcolorblackwhere λ is the free space wavelength. The tip textcolorblackapex angle is $\pi/6$ but the angle of the coupling cone is chosen a little wider (35^0) so that the tip apex can be closer to the sample. The PML thickness in Figure 2b is 1.4λ.

3.1. Excitation

To examine the behavior of the compressed model matrix, the metallic tip in Figure 2b is illuminated by a dipole in two cases: a dipole parallel to the axis of symmetry and a perpendicular one. These two cases excite $m = 0$ and $m = 1$ modes in (6), respectively. textcolorblackIn both cases the dipole is located 1.5λ below the tip apex.

Case $\mathbf{m = 0}$. In this case, the magnetic vector potential is in z direction. The magnetic potential vector of this dipole in cylindrical coordinates is,

$$\mathbf{A} = \frac{\mu_0}{4\pi\sqrt{r^2+z^2}} e^{ik\sqrt{r^2+z^2}} \delta l \hat{\mathbf{z}} \tag{17}$$

where δl is the length of the dipole. In this case, the electrical field has no dependence on ϕ, which means that the Fourier series of the field only contains the $m = 0$ mode.

Case $\mathbf{m = 1}$. In this case, magnetic vector potential is in x direction of Cartesian coordinates. The magnetic potential vector of this dipole is,

$$\begin{aligned} \mathbf{A} &= \frac{\mu_0}{4\pi\sqrt{r^2+z^2}} e^{ik\sqrt{r^2+z^2}} \delta l \hat{\mathbf{x}} \\ &= \frac{\mu_0}{4\pi\sqrt{r^2+z^2}} e^{ik\sqrt{r^2+z^2}} \delta l (\cos(\phi)\hat{\mathbf{r}} - \sin(\phi)\hat{\boldsymbol{\phi}}) \end{aligned} \tag{18}$$

In this case, the electrical field dependence on ϕ has the form of $\cos(\phi)$ and $\sin(\phi)$, which means that the Fourier series of the field only contains the $m = 1$ mode.

[†]We use the term *almost linear complexity* when the complexity behaves like $O(n\log^q n)$ for some $q \geq 0$ independent of n, see [12, Sect. 1.2.1].

Since the electromagnetic field goes to infinity in the location of the dipole, we introduce a ball around the dipole and impose the excitation on the boundary of this ball such that we have total fields outside the ball and only scattered fields exist inside the ball. In our experiments this radius set to be 1.4λ.

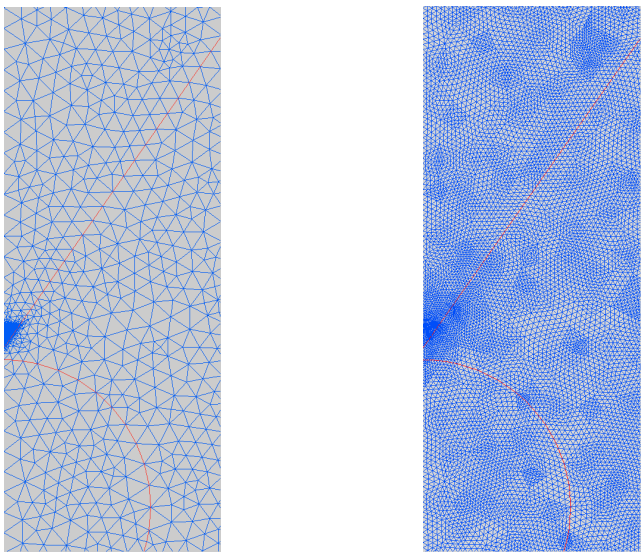
3.2. Finite element meshes

To investigate the efficiency of the compression of the model matrix $\mathbf{S}$, our experiments are performed for two kinds of mesh refinement: adaptive and regular. In adaptive refinement, the elements inside the coupling cone and below the curvature of the tip apex are refined in each step, while in regular refinement all the elements in the domain are refined regularly.

The numbers of DoFs for adaptively and regularly refined meshes are given in Table I. In this table, N_Γ is the number of DoFs on the boundary and is the number of rows and columns of the model matrix $\mathbf{S}$. Likewise, N_{Ω_1} is the number of DoFs in Ω_1 and is the number of rows and columns of the matrix $\mathbf{A}_{\mathbf{in},\mathbf{in}}$ in (13).

Mesh	Adaptive				Regular			
	$m=0$		$m=1$		$m=0$		$m=1$	
	N_Γ	$N_{\Omega 1}$	N_Γ	$N_{\Omega 1}$	N_Γ	$N_{\Omega 1}$	N_Γ	$N_{\Omega 1}$
1	43	1319	84	1607	43	1319	84	1607
2	56	1649	110	2019	86	5293	170	6752
3	85	4381	168	5574	172	21206	342	27662
4	190	26829	378	35275	344	84892	686	111962
5	404	208372	806	276656	688	339704	1374	450482

Table I. Number of DoFs for adaptively and regularly refined meshes for the configuration of Figure 2b. In this table N_{Ω_1} refers to the dimension of $\mathbf{A}_{\mathbf{in},\mathbf{in}}$ and N_Γ refers to the size of the schur complement matrix $\mathbf{S}$, see (15a).



(a) Adaptively refined (b) Regularly refined

Figure 3. Illustration of mesh 3 from Table I. Dipole is located 1.5λ below the apex of the tip.

The meshes in Table I are the same for both $m=0$ and $m=1$ modes, however the numbers of DoFs are different for each mode. The reason is the following: in the case $m=0$ there is a total

decoupling between edge and vertex DoFs, i.e., $\mathbf{A_{ev}}$ and $\mathbf{A_{ve}}$ in (12) are vanishing. Considering the fact that $E_\phi = 0$ in the case of $m = 0$, the experiments are carried out only for $\mathbf{A_{ee}}$ and not for the entire FE matrix in (12). In other words, only DoFs defined on the edges are considered.

The meshing of the structure of Figure 2b, is shown in Figure 3 for adaptive and regular refinement for the textcolorblackmesh 3 in Table I. In the following sections we explain how to build an $\mathcal{H}$-matrix approximation of $\mathbf{S}$.

4. BUILDING AN $\mathcal{H}$-MATRIX APPROXIMATION OF $\mathbf{S}$ USING ACA

In this section, we compress the model matrix $\mathbf{S}$ by means of ACA algorithm [1, 6]. First, $\mathbf{S}$ is calculated through (15a) using arithmetics of sparse matrices and sparse Gaussian elimination. Then, $\mathbf{S}$ is partitioned and the ACA algorithm is applied to admissible blocks to provide a low-rank approximation with a guaranteed accuracy, for details see [3, Sect. 3.4]. We refer to this approach as "ACA-based approach".

4.1. ACA-based approach

In this approach, the FE matrix is assembled according to (12). Then, the sparse direct solver of MatLab is used to compute *all the entries* of $\mathbf{S}$ through (15a). Before approximating $\mathbf{S}$ by an $\mathcal{H}$-matrix, we apply a diagonal scaling to $\mathbf{S}$, to balance the effect of all DoFs which are computed on non-regular meshes or are of different kind (defined on edges or nodes).

First, we find an admissible partitioning of $\mathbf{S}$ as mentioned in Section 1.2. Taking advantage of the availability of the position information for DoFs on Γ, a geometrical clustering approach is adopted [3, Sect. 1.4.1.1].

In geometrical clustering, a coordinate is associated to each row index of $\mathbf{S}$: the coordinates of the middle points of the edges are associated to the DoFs defined on the edges, and the coordinates of vertices are associated to the DoFs defined on the vertices. textcolorblackWe denote the indices of rows (or columns) of $\mathbf{S}$ by I and x_i is the coordinate assigned to DoF i. Let t and s be subsets of I. We define:textcolorblack

$$\begin{aligned}
\text{DoF locations:}\quad & \chi_t := \bigcup x_i, i \in t \quad \text{and} \quad \chi_s := \bigcup x_i, \quad i \in s,\\
\text{centers:}\quad & c_t := \frac{\sum_i x_i}{\#t}, i \in t \quad \text{and} \quad c_s := \frac{\sum_i x_i}{\#s}, i \in s,\\
& \mathrm{diam}\chi_t := \max\|x_i - x_j\|, i, j \in t \quad \text{and} \quad \mathrm{diam}\chi_s := \max\|x_i - x_j\|, \quad i, j \in s,\\
& \mathrm{dist}(\chi_t, \chi_s) := \|c_t - c_s\|.
\end{aligned}$$

The block $t \times s$ is admissible if

$$\min\{\mathrm{diam}\chi_t, \mathrm{diam}\chi_s\} \leq \mathrm{dist}(\chi_t, \chi_s). \tag{19}$$

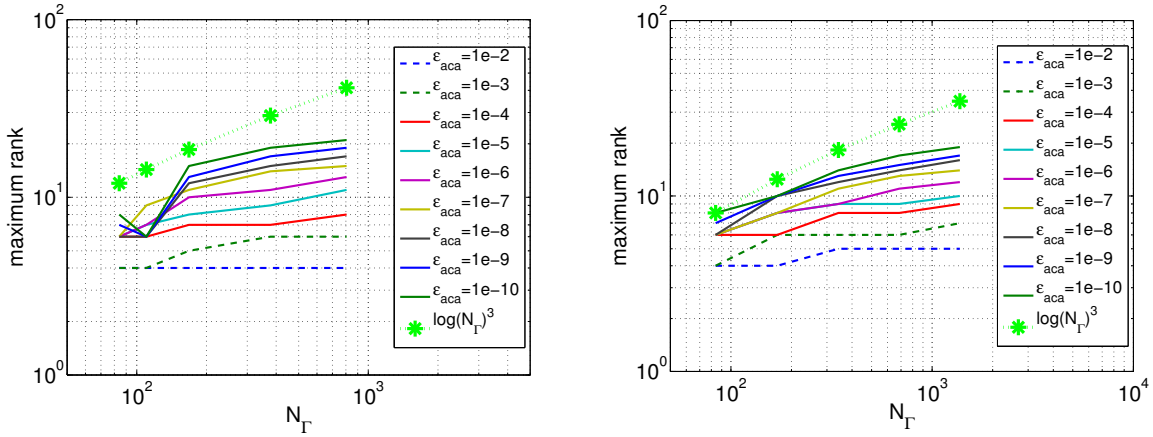
Inequality (19) is the geometrical admissibility condition used in this section.

We use the `genClusterTree2d_pca` routine implemented in AHMED to build the geometrical cluster tree based on the PCA algorithm [20]. The input parameters for the geometrical clustering algorithm are the coordinates of DoFs and the smallest size of blocks in the partitioned matrix, b_{min}. A block cluster tree is built based on the aforementioned cluster tree and a geometric admissibility condition (19). We use the `genBlockClusterTree` routine implemented in AHMED for this purpose.

The admissible blocks of the partitioned $\mathbf{S}$ are approximated by low-rank matrices and inadmissible blocks are saved in dense format. We apply the ACA algorithm to compress the admissible blocks by using the `AcaRowApproximator` routine in AHMED.

textcolorblackWhen denoting an admissible matrix block i by $\mathbf{B}_i$, the error control parameter of the ACA algorithm ε_{aca} guarantees,

$$\|\mathbf{R}_{k_i}\|_F \leq \varepsilon_{aca}\|\mathbf{B}_i\|_F, \tag{20}$$



(a) $m = 1$, adaptively refined meshes.

(b) $m = 1$, regularly refined meshes

Figure 4. textcolorblackMaximum rank of the blocks in the compressed model matrix $\mathbf{S}$ for $b_{min} = 5$ for different ACA error control parameters in the ACA-based approach (see Table I).

where $\mathbf{R}_{k_i}$ is the difference between $\mathbf{B}_i$ and its rank-k_i approximation[‡]. In Section 4.2, the parameter ε_{aca} is used as a stopping criterion for the ACA algorithm. The cost for getting the rank-k_i approximation of $\mathbf{B}_i$ through ACA is $k_i^2(m_i + n_i)$, where $m_i \times n_i$ is the dimension of $\mathbf{B}_i$.

4.2. Numerical results

All the numerical results of this section are obtained based on the model problem introduced in Section 3. We compress the Schur complement matrix $\mathbf{S}$ for all meshes in Table I. So, the number of DoFs on the boundary N_Γ agrees with the size of $\mathbf{S}$. The minimum size of the blocks in the partitioned $\mathbf{S}$, i.e., the size of inadmissible blocks is set to $b_{min} = 5$ for all clusterings in our experiments.

textcolorblackAs already mentioned in Section 2.4, we expect a maximum rank complexity better than $O(\log^3 N_\Gamma)$ and storage complexity better than $O(N_\Gamma \log^4 N_\Gamma)$ to save the $\mathcal{H}$-matrix approximation of $\mathbf{S}$. Figure 4 shows how the maximum rank of the blocks grows by using ACA and Figure 5 illustrates memory needed (in bytes) to save $\mathcal{H}$-matrix approximation of $\mathbf{S}$. Considering the complexity of the ACA algorithm (discussed in Section 4.1) and maximum rank of the blocks it can be concluded that the ACA algorithm runs with almost linear complexity.

textcolorblackTo study the compression efficiency, we define the compression rate by:

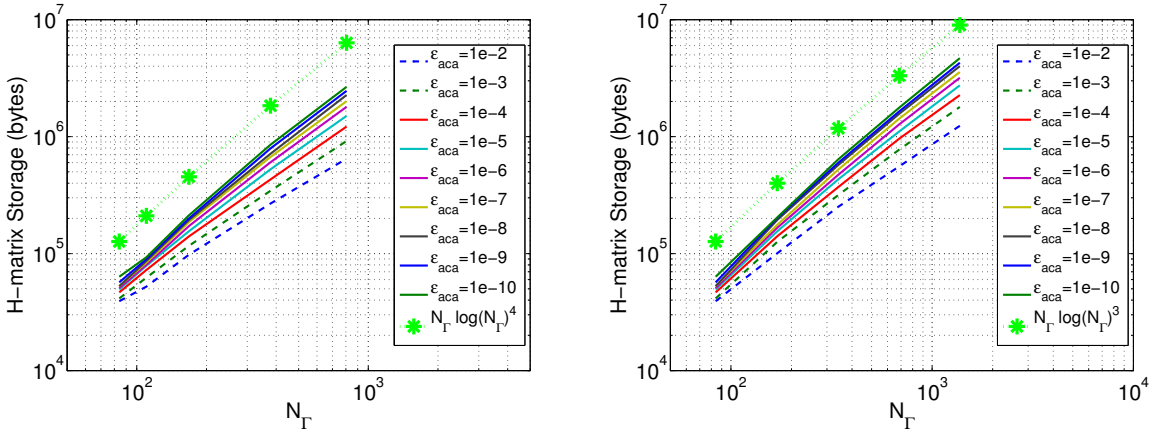
$$\text{compression ratio} := \frac{M_{\text{dense}}}{M_{\text{cmprss}}},$$

where M_{dense} is the memory needed to save a dense matrix and M_{cmprss} is the memory needed to save its $\mathcal{H}$-matrix approximation. Figure 6 shows the compression rate of $\mathbf{S}$. For larger sizes of $\mathbf{S}$ the compression rate increases which means that the compression is more efficient. This is expected since $\mathcal{H}$-matrices can store an approximation of a dense Schur complement of a FE matrix with almost linear complexity.

textcolorblackFigure 7 shows the relative Euclidean norm ε_S^{err} (computed by the `norm`-function of MatLab) of the $\mathbf{S}$ approximation error versus the ACA error control parameter, ε_{aca}. This figure indicates that ε_{aca} is a reliable control parameter for the accuracy of the compressed model matrix.

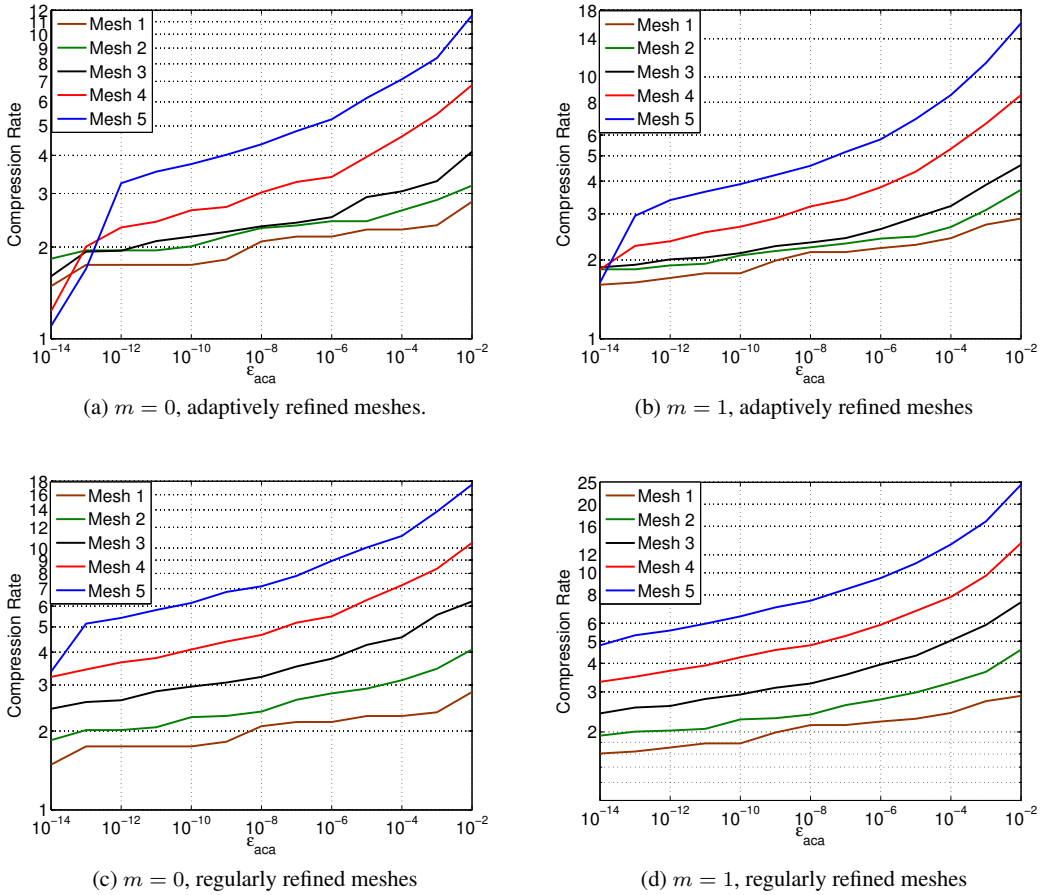
Finally, we examine the effect of textcolorblackcompressing $\mathbf{S}$ on the FE solution in the outer domain, Ω_2 in Figure 2a. The electrical field computed from (14) in Ω_2 by FEM is considered to

[‡]By $\|\cdot\|_F$ we denote the Frobenius norm of the matrix.



(a) $m = 1$, adaptively refined meshes. (b) $m = 1$, regularly refined meshes

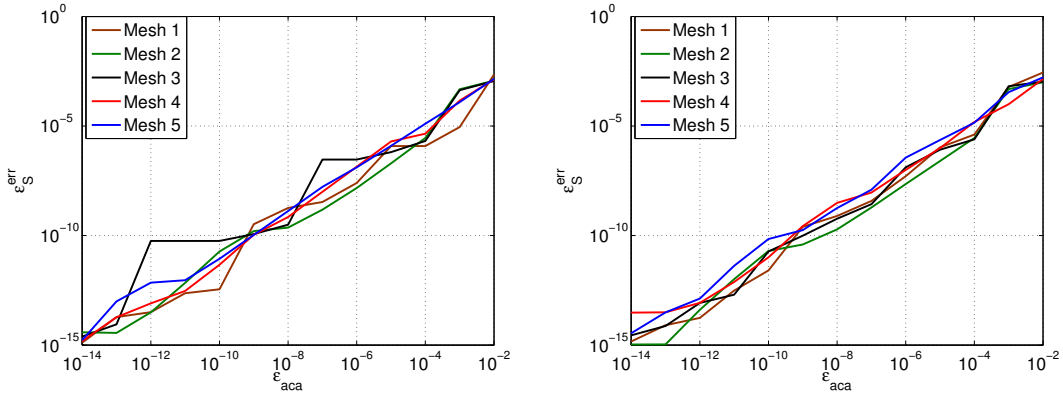
Figure 5. textcolorblackMemory needed to save the compressed model matrix $\mathbf{S}$ for $b_{min} = 5$ for different ACA error control parameters in the ACA-based approach (see Table I).



(a) $m = 0$, adaptively refined meshes. (b) $m = 1$, adaptively refined meshes

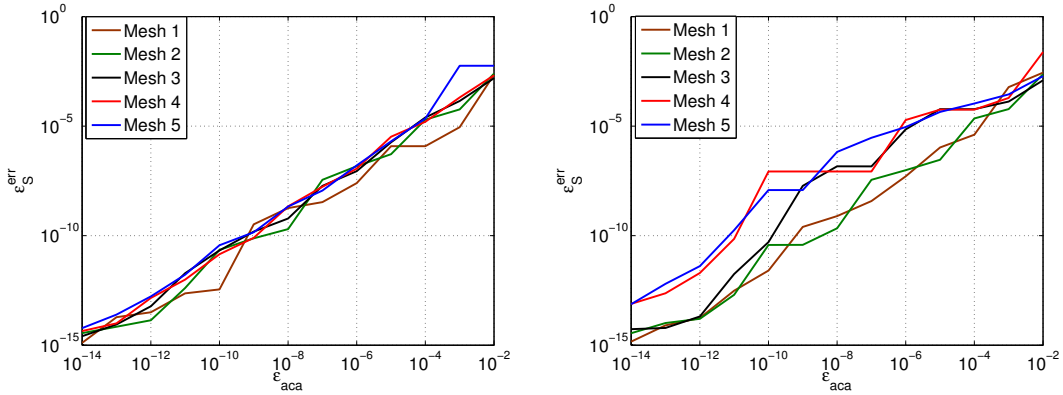
(c) $m = 0$, regularly refined meshes (d) $m = 1$, regularly refined meshes

Figure 6. textcolorblackCompression rate for the model matrix $\mathbf{S}$ vs the ACA error control parameter ε_{aca} and for $b_{min} = 5$ for different meshes (see Table I) in the ACA-based approach.



(a) $m = 0$, adaptively refined meshes.

(b) $m = 1$, adaptively refined meshes



(c) $m = 0$, regularly refined meshes

(d) $m = 1$, regularly refined meshes

Figure 7. Relative Euclidean error norm of the compressed model matrix textcolorblack ε_S^{err} vs the error control parameter ε_{aca} and for $b_{min} = 5$ for different meshes (see Table I) in ACA-based approach.

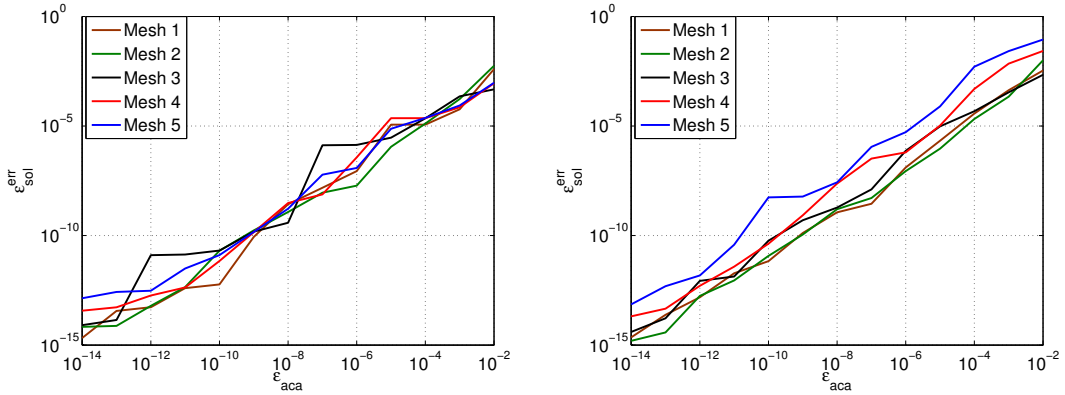
be the reference solution when the exact entries of $\mathbf{S}$ are used on the coupling boundary. The FE solution obtained by the compressed $\mathbf{S}$ is the approximate solution. The relative L_2 norm of the difference of reference and solution error textcolorblackε_{sol}^{err} is shown in Figure 8. Clearly, a more accurate solution in the outer domain is achieved where the approximation of $\mathbf{S}$ is more accurate.

5. APPROXIMATING THE SCHUR COMPLEMENT THROUGH $\mathcal{H}$-ARITHMETICS

In this section, another strategy is adopted to build an $\mathcal{H}$-matrix approximation of $\mathbf{S}$ such that one does not need to precompute all the entries of $\mathbf{S}$. We begin with the direct conversion of the sparse FE matrices in (15) into $\mathcal{H}$-matrices and then exploit $\mathcal{H}$-arithmetics to obtain $\mathbf{S}$ directly in an $\mathcal{H}$-matrix format. We refer to this approach as the "$\mathcal{H}$-arithmetic approach".

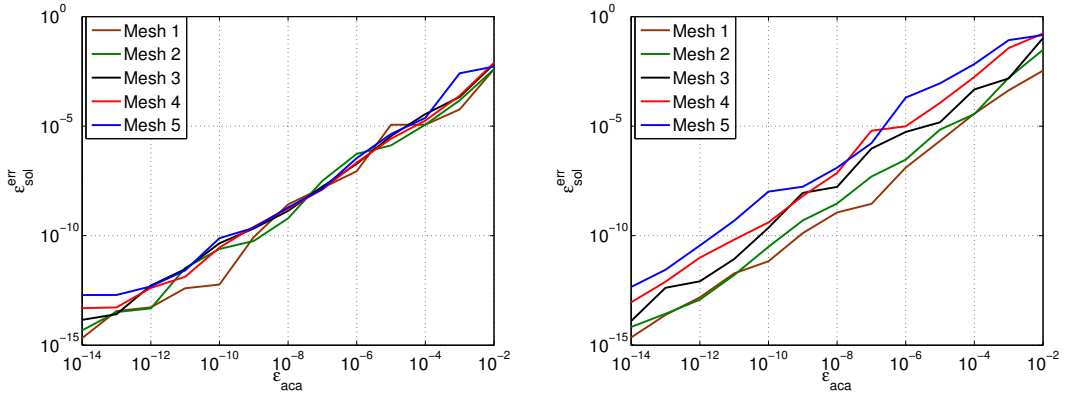
5.1. Partitioning based on the matrix graph

Reconsidering (15a), all FE matrices contributing to the calculation of $\mathbf{S}$ are sparse. We compute these FE matrices and save them in Compressed Row Storage (CRS) format. To convert these matrices directly to $\mathcal{H}$-matrices, they are first partitioned as described textcolorblacklater. In contrast to the clustering in Section 4.1, the cluster and block cluster tree are built through matrix graph based clustering.



(a) $m = 0$, adaptively refined meshes.

(b) $m = 1$, adaptively refined meshes



(c) $m = 0$, regularly refined meshes

(d) $m = 1$, regularly refined meshes

Figure 8. Convergence of the L_2 norm of the solution error textcolorblackε_{sol}^{err} in Ω_2 when $b_{min} = 5$. Here, ε_{aca} is the ACA error control parameter.

In the $\mathcal{H}$-arithmetic approach, textcolorblackfirst we partition the undirected matrix graphs associated to each of the symmetric sparse matrices in (15a). By using a nested dissection algorithm for graph partitioning, the obtained ternary cluster tree is called a nested dissection cluster tree and is defined in [3, Sect. 4.5.1]. A vertex separator algorithm implemented in the Metis library [13], is used by the `cluster_alg` routine in AHMED to generate nested dissection cluster trees. Besides the sparsity pattern, the other parameter involved in building the algebraic cluster tree is the minimum size of the blocks in the partitioned matrix. Then, the block cluster tree is built by using the following algebraic admissibility condition: textcolorblackfor a given connected sparse symmetric matrix $\mathbf{A}$ we write $d_{ij}(\mathbf{A})$, $i, j \in I$, for the length of the shortest path in the matrix graph connecting the indices i and j (I is the set of the indices of rows (or columns) of $\mathbf{A}$). Note that symmetry of $\mathbf{A}$ guarantees $d_{ij}(\mathbf{A}) = d_{ji}(\mathbf{A})$. Now, for $t \subset I$and $s \subset I$ we define [3, eq. 1.12]:

$$\operatorname{dist}(t, s) := \min_{i \in t, j \in s} d_{ij}(\mathbf{A}) \quad \text{and} \quad \operatorname{diam}(t) := \max_{i,j \in t} d_{ij}(\mathbf{A}).$$

Then algebraic admissibility condition for block $t \times s$ reads

$$\min\{\operatorname{diam}(t), \operatorname{diam}(s)\} \leq \operatorname{dist}(t, s). \tag{21}$$

Remark 1
Obviously, the idea of clustering of the matrix indices based on the undirected matrix graph is

applicable only to symmetric matrices, whereas the matrices $\mathbf{A}_{\mathbf{\Gamma},\mathbf{in}}$ and $\mathbf{A}_{\mathbf{in},\mathbf{\Gamma}}$ in (15a) are not even square. Furthermore, in these rectangular matrices most of the row and column vertices are not connected. These facts render the algebraic admissibility condition useless for the partitioning of the rectangular matrices.

One may treat the rectangular matrices in (15a) as regular dense matrices such that computations related to rectangular matrices are performed column by column. This increases the complexity of the computations by a factor of N_Γ. Another possibility is partitioning of the rectangular matrices by choosing a type of admissibility condition different from the algebraic one.

In this work, we use the geometric admissibility condition (19), to partition the rectangular matrices. In light of the sparsity of the matrices, we expect that a reasonable number of the blocks are empty or low rank and partitioning improves the complexity of $\mathcal{H}$-arithmetics. The two algorithms are compared in Section 6, Figure 15.

Remark 2
As mentioned above, in the experiments for the $\mathcal{H}$-arithmetic approach we build the algebraic cluster trees only for $\mathbf{A}_{\mathbf{in},\mathbf{in}}$ and $\mathbf{A}^{\mathbf{in}}_{\mathbf{\Gamma},\mathbf{\Gamma}}$ in (15a). In the case of $m = 0$, as discussed in Section 3.2, the compression is only applied to $\mathbf{A}_{\mathbf{ee}}$ in (12). Since the supports of the DoFs on the edges of the coupling boundary do not share an element, the matrix $\mathbf{A}^{\mathbf{in}}_{\mathbf{\Gamma},\mathbf{\Gamma}}$ is a diagonal matrix in this case and no connectivity between DoFs is presented in its matrix graph. Hence, clustering of the row indices of $\mathbf{A}^{\mathbf{in}}_{\mathbf{\Gamma},\mathbf{\Gamma}}$ based on the matrix graph partitioning fails. In this case, we add some very small entries (in the order of the machine precision) to $\mathbf{A}^{\mathbf{in}}_{\mathbf{\Gamma},\mathbf{\Gamma}}$ to indicate which edges share a vertex on the coupling boundary.

5.2. *Computations in $\mathcal{H}$-arithmetics*

Before converting the sparse matrices to $\mathcal{H}$-matrices, we have to find a way of dealing with the different nature of DoFs involved. We use diagonal scaling to make the the diagonal entries of $\mathbf{A}_{\mathbf{in},\mathbf{in}}$ and $\mathbf{A}^{\mathbf{in}}_{\mathbf{\Gamma},\mathbf{\Gamma}}$ equal to unity.

After matrix partitioning, the sparse matrices are converted to $\mathcal{H}$-matrices directly as follows: inadmissible blocks are saved in dense format and admissible blocks are saved in low-rank format, where low-rank approximations of admissible blocks are obtained through truncated SVD. As soon as the sparse matrices in (15) are available in $\mathcal{H}$-matrix format, we rely on arithmetics for hierarchical matrices ($\mathcal{H}$-arithmetics) to compute the $\mathcal{H}$-matrix format of $\mathbf{S}$.

Instead of computing $\mathbf{A}^{-\mathbf{1}}_{\mathbf{in},\mathbf{in}}$, which is the only huge and dense matrix in (15a), we compute hierarchical LU factorization (H-LU),

$$\mathbf{A}_{\mathbf{in},\mathbf{in}} \approx \mathbf{L}_{\mathbf{H}}\mathbf{U}_{\mathbf{H}} \tag{22}$$

where $\mathbf{L}_{\mathbf{H}}$ and $\mathbf{U}_{\mathbf{H}}$ are triangular $\mathcal{H}$-matrices. In [4, Theorem 3.1] it has been shown that $\mathbf{L}_{\mathbf{H}}$ and $\mathbf{U}_{\mathbf{H}}$ do exist (in combination with [4, Theorem 3.4]) and can be approximated with almost linear complexity [3, Sect. 4.5.2].

Remark 3
Since $\mathbf{A}^{-\mathbf{1}}_{\mathbf{in},\mathbf{in}}$ is the only full matrix in (15a) and also the biggest one, one may come up with the idea of storing only this matrix as its LU factors, namely $\mathbf{L}_{\mathbf{H}}$ and $\mathbf{U}_{\mathbf{H}}$ in (22), and use matrix-vector multiplication to acquire the entries of $\mathbf{S}$. However, this approach is not promising, because, the size of $\mathbf{A}_{\mathbf{in},\mathbf{in}}$ can grow quadrically with respect to the size of $\mathbf{S}$ while the storage in $\mathcal{H}$-matrix format is of almost linear complexity.

After converting the sparse matrices to $\mathcal{H}$-matrices, all arithmetics in (15a) (namely LU-factorization, backward/forward substitution, matrix multiplication and addition) are performed by $\mathcal{H}$-arithmetics. The required $\mathcal{H}$-arithmetics for the computation of $\mathbf{S}$ in (15) is the following,

(I) Compute $\mathbf{L}_{\mathbf{H}}$ and $\mathbf{U}_{\mathbf{H}}$ (by using the `HLU` routine of AHMED),

(II) Forward substitution: $\mathbf{L}_{\mathbf{H}}\mathbf{X} = \mathbf{A}^{\mathbf{H}}_{\mathbf{in},\mathbf{\Gamma}}$ (by using the `LtHGeH_solve` routine of AHMED),

(III) Backward substitution: $\mathbf{U}_{\mathbf{H}}\mathbf{Y} = \mathbf{X}$ (by using the `UtHGeH_solve` routine of AHMED),

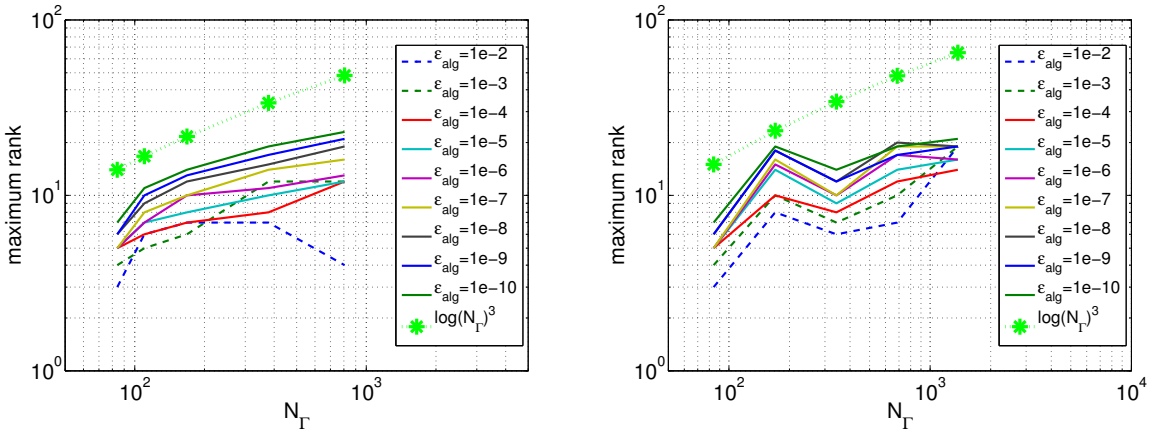
(IV) $\mathcal{H}$-multiplication: $\mathbf{Z} = -\mathbf{A}^{\mathbf{H}}_{\mathbf{\Gamma},\mathbf{in}}\mathbf{Y}$ (by using the `mltaGeHGeH` routine of AHMED),

(V) $\mathcal{H}$-addition: $\mathbf{S} = \mathbf{A}^{\mathbf{H}}_{\mathbf{\Gamma},\mathbf{\Gamma}} + \mathbf{Z}$ (by using the `addGeHGeH` routine of AHMED).

5.3. Numerical results

All numerical results in this section are obtained based on the model problem introduced in Section 3. In all the experiments, the parameter textcolorblackε_{alg} specifies the accuracy of the computations such as SVD and $\mathcal{H}$-arithmetics. This parameter is used as an error control parameter. Similar to the ACA-based approach, we set the smallest size of the blocks, b_{min}=5 for all $\mathbf{S}$ compression experiments.

textcolorblackThe maximum rank of the blocks and memory needed to save $\mathcal{H}$-matrix approximation of $\mathbf{S}$ are shown in Figure 9 and Figure 10, respectively. These figures show that the dependence of the memory consumption on N_Γ is in agreement with previous discussion in Section 2.4.



(a) $m = 1$, adaptively refined meshes.

(b) $m = 1$, regularly refined meshes

Figure 9. textcolorblackMaximum rank of blocks in the compressed model matrix $\mathbf{S}$ for $b_{min} = 5$ in the $\mathcal{H}$-arithmetics approach (see Table I).

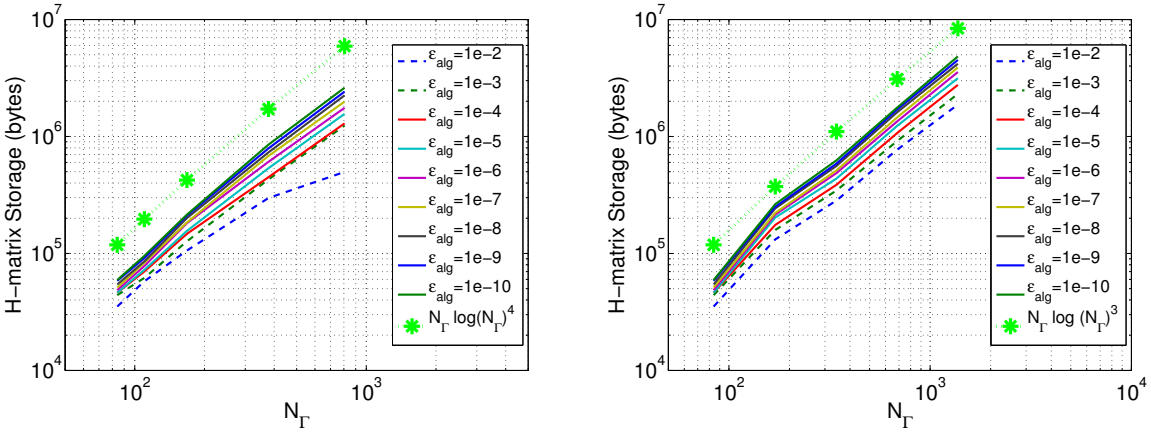
We have compressed the matrix model for meshes in Table I through the $\mathcal{H}$-arithmetic approach described in sections 5.1 and 5.2.

Figure 11 shows the compression rate which is defined similar to the ACA-based approach in Section 4.2. This figure shows that the compression is more efficient when the size of $\mathbf{S}$ increases. Figure 12 shows the relative Euclidean error norm textcolorblackε_S^{err} for different accuracies of compressed $\mathbf{S}$ matrices. This figure shows that controlling the accuracy of the compression by means of the error control of H-arithmetic textcolorblackε_{lag} is reliable for both cases $m = 0$ and $m = 1$.

Figure 13 shows the L_2 norm of the solution error textcolorblackε_{sol}^{err} computed in the same way as discussed in the ACA-based approach in Section 4.2.

6. COMPARISON

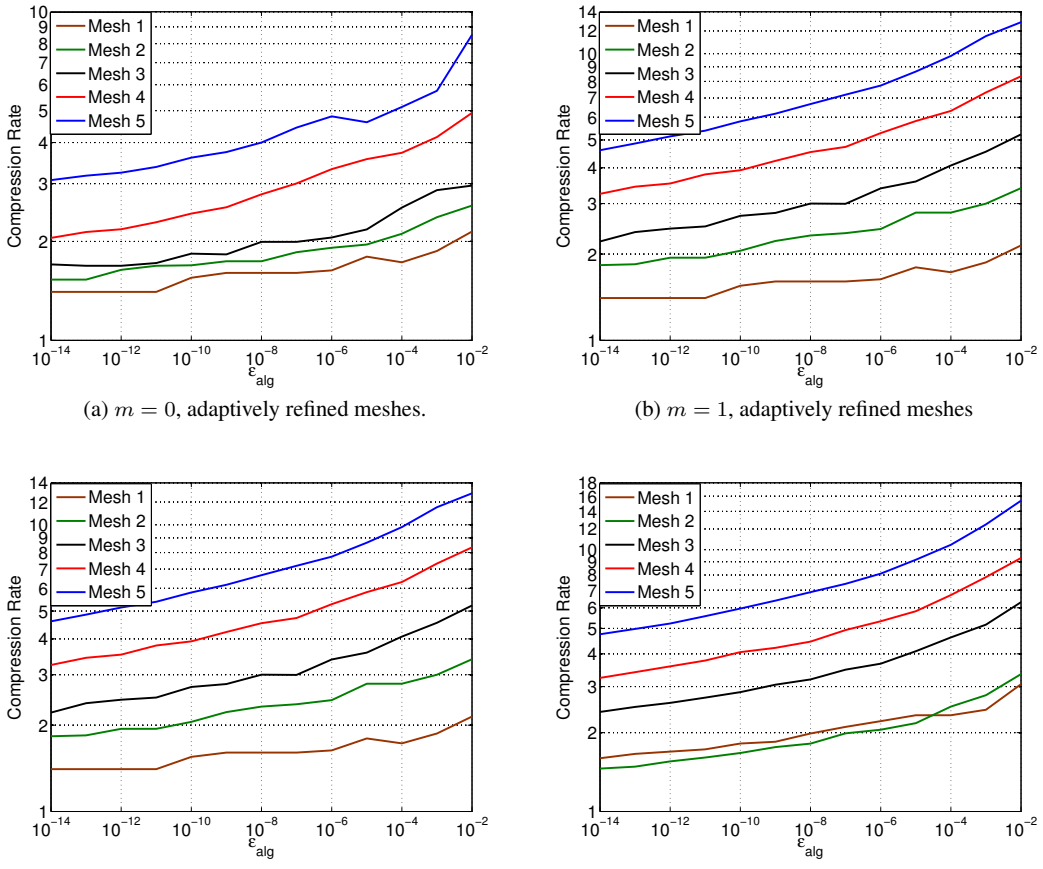
textcolorblack In Section 4 and Section 5, we presented two methods of compressing the tip model matrix $\mathbf{S}$. In this section, some studies are provided to compare the two methods with respect to the compression efficiency and the computational cost of $\mathcal{H}$-matrix approximation of $\mathbf{S}$. textcolorblack



(a) $m = 1$, adaptively refined meshes.

(b) $m = 1$, regularly refined meshes

Figure 10. textcolorblackMemory needed to save the compressed model matrix $\mathbf{S}$ for $b_{min} = 5$ in the $\mathcal{H}$-arithmetics approach (see Table I).



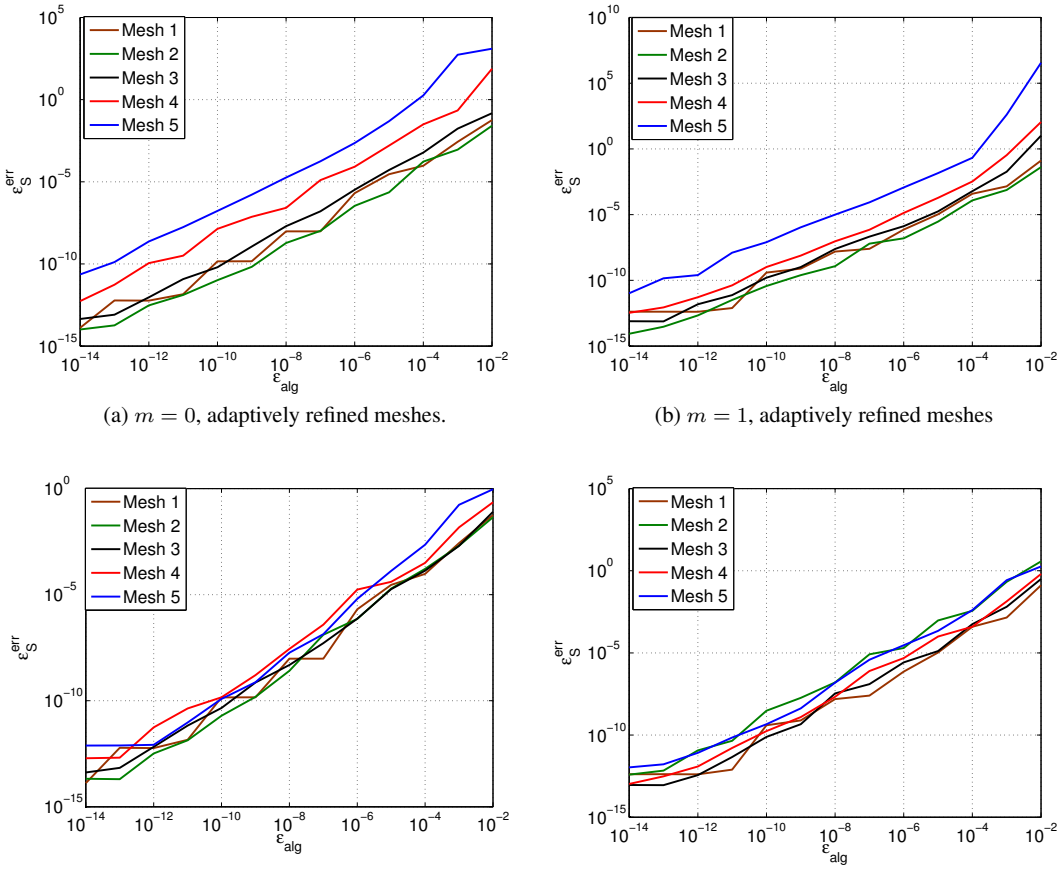
(a) $m = 0$, adaptively refined meshes.

(b) $m = 1$, adaptively refined meshes

(c) $m = 0$, regularly refined meshes

(d) $m = 1$, regularly refined meshes

Figure 11. Compression rate of $\mathbf{S}$ for matrices of different size (see Table I) using $\mathcal{H}$-arithmetics.



(a) $m = 0$, adaptively refined meshes.

(b) $m = 1$, adaptively refined meshes

(c) $m = 0$, regularly refined meshes

(d) $m = 1$, regularly refined meshes

Figure 12. Relative Euclidean error norm of the compressed model matrix textcolorblackε_S^{err} vs the error control parameter textcolorblackε_{lag} and $b_{min} = 5$ for different meshes (see Table I) using $\mathcal{H}$-arithmetics.

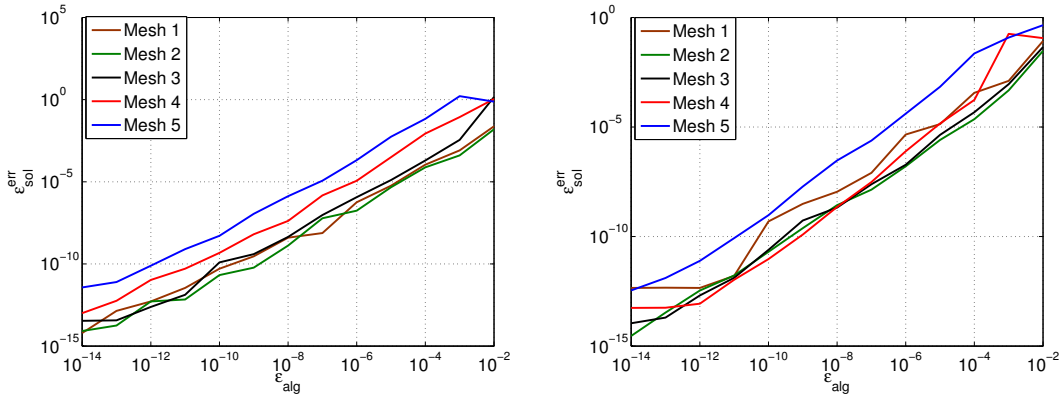
6.1. Compression efficiency

textcolorblack In previous sections, we have shown that the relative Euclidean norm of the compression error ε_S^{err} for each method is controlled by the associated error control parameter (see Figure 7 and Figure 12). Figure 14 compares the compression rate of $\mathbf{S}$ versus ε_S^{err} for both methods. This Figure illustrates that the two methods are rather equivalent in the case of regular refinement, however, the ACA-based approach shows better compression efficiency in the case of adaptively refined meshes. Also, one can observe that the maximum accuracy obtainable by means of $\mathcal{H}$-arithmetic approach is decreasing as the size of $\mathbf{S}$ increases due to loss of significance digits.
textcolorblack

6.2. Cost of generating the compressed model matrix

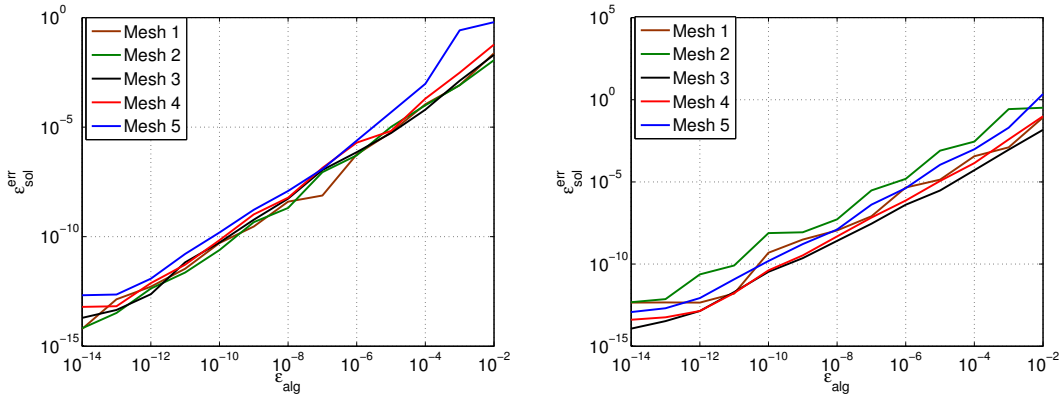
textcolorblack In the ACA-based approach the entries of $\mathbf{S}$ have to be available which involves LU-factorization of $\mathbf{A_{in,in}}$. In most practical situations the LU-factoriztion of sparse matrices arising from two dimensional FE discretizations can be done with a complexity close to $O(N_{\Omega 1})$. Under this optimistic assumption, considering that we need to compute $\mathbf{S}$ column-wise, the complexity of computation of $\mathbf{S}$ is not better than $O(N_\Gamma^3)$ (note that $N_{\Omega_1} \backsim N_\Gamma^2$).

textcolorblack As already mentioned in Section 4.2, the $\mathcal{H}$-matrix approximation of $\mathbf{S}$ by the ACA algorithm can be done with almost linear complexity which is better than $O(N_\Gamma^2)$. Therefore,



(a) $m = 0$, adaptively refined meshes.

(b) $m = 1$, adaptively refined meshes



(c) $m = 0$, regularly refined meshes

(d) $m = 1$, regularly refined meshes

Figure 13. Convergence of the L_2 norm of the solution error textcolorblackε_{sol}^{err} in Ω_2 for $b_{min} = 5$. Here, textcolorblackε_{lag} is the error control parameter for $\mathcal{H}$-arithmetics.

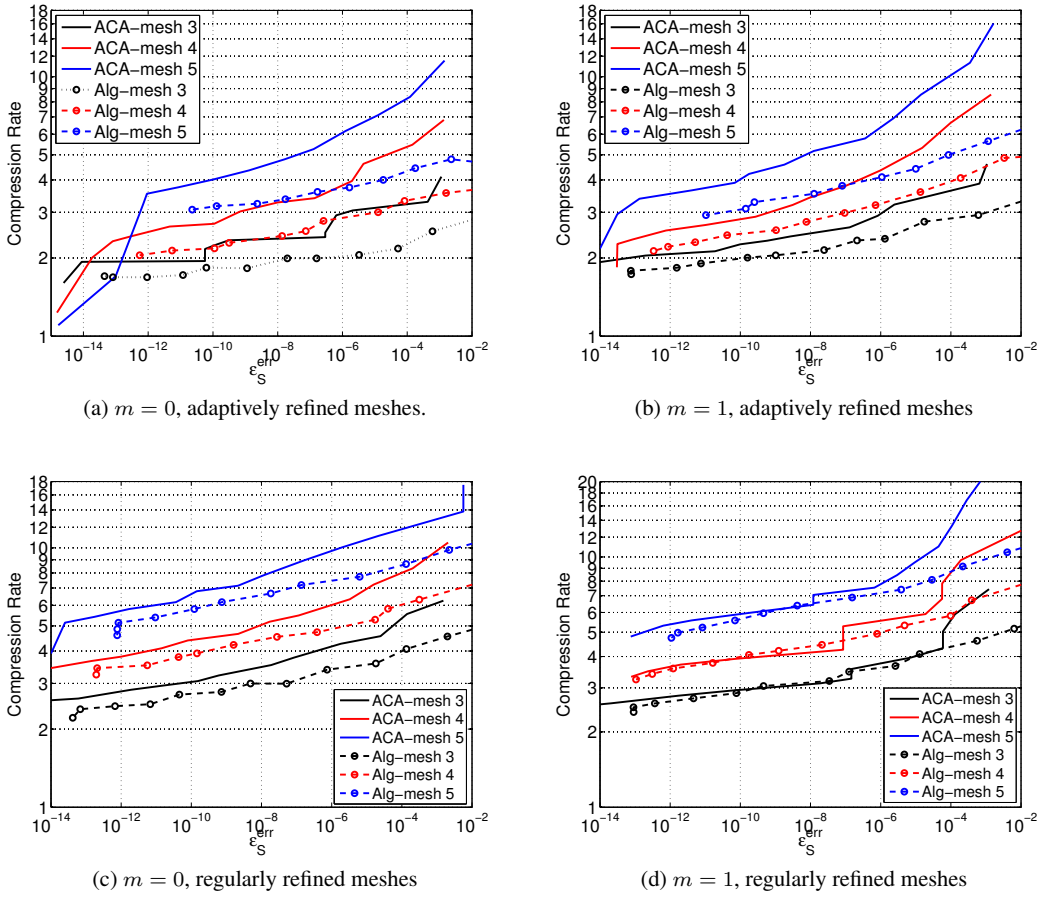
the complexity of the ACA-based approach is dominated by complexity of the computation of $\mathbf{S}$ entries from (15a) which is optimally $O(N_\Gamma^3)$.

textcolorblack The $\mathcal{H}$-arithmetics used in Section 5 can be performed with almost linear complexity, for example, addition of $\mathcal{H}$-matrices [3, Theorem 2.17], multiplication of $\mathcal{H}$-matrices [3, Theorem 2.26] and LU-factorization as discussed in Section 5.2. Without partitioning of the rectangular matrices, $\mathbf{A}_{\mathbf{\Gamma,in}}$ and $\mathbf{A}_{\mathbf{in,\Gamma}}$ a complexity similar to that of the ACA-based approach is expected. However, by partitioning the rectangular matrices in (15a) we may optimally expect almost linear complexity with respect to $N_{\Omega 1}$ in the $\mathcal{H}$-arithmetic approach.

textcolorblackTo compare the complexity of the two approaches in our experiments numerically, we measure the time needed for $\mathcal{H}$-arithmetics in the $\mathcal{H}$-arithmetic approach (for both partitioned and unpartitioned rectangular matrices) and the time needed for computing the entries of $\mathbf{S}$ through sparse matrix arithmetic in MatLab. The computations are performed on an Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz with Linux operating system. The measured times are shown in Figure 15. This figure shows better computational complexity for the $\mathcal{H}$-arithmetic approach when the rectangular matrices are partitioned.

7. CONCLUSION

textcolorblackIn the context of finite element discretization we proposed a data-sparse non-local surface impedance boundary condition (“compressed model") to take into account the presence of a scatterer – a SNOM tip in our application – in frequency domain electromagnetic simulations.



(a) $m = 0$, adaptively refined meshes.

(b) $m = 1$, adaptively refined meshes

(c) $m = 0$, regularly refined meshes

(d) $m = 1$, regularly refined meshes

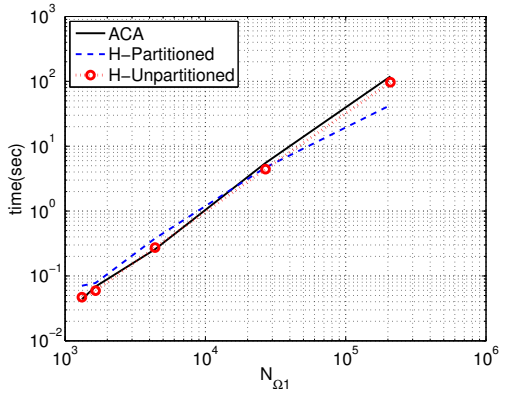
Figure 14. Compression rates of $\mathcal{H}$-matrices obtained through the ACA-based approach (ACA) and the $\mathcal{H}$-arithmetic one (Alg) versus the relative Euclidean error norm of the compressed matrix, textcolorblackε_S^{err}
.

We rely on $\mathcal{H}$-matrix representations that can be precomputed beforehand in two ways, which we compared in numerical experiments. Inside an iterative solver the compressed model can be applied with almost linear complexity in the number of unknowns located on the coupling interface.

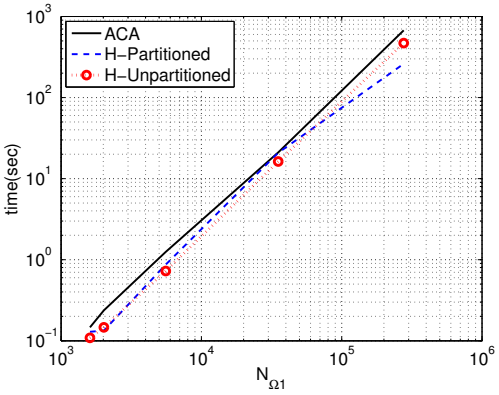
textcolorblackAs a promising application of our technique we envisage Boundary Element field simulations in a three dimensional sample-tip setting with a moving tip. textcolorblackThen the precomputed compressed model can be reused for many simulation runs.

REFERENCES

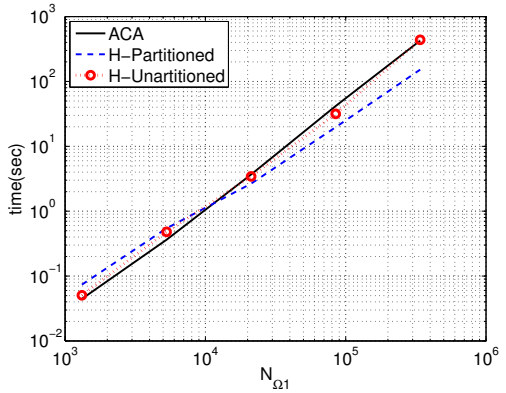
1. Mario Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.
2. Mario Bebendorf. Another software library on hierarchical matrices for elliptic differential equations (AHMED). *Universität Leipzig, Fakultät für Mathematik und Informatik*, 2005.
3. Mario Bebendorf. *Hierarchical matrices*. Springer, 2008.
4. Mario Bebendorf and Thomas Fischer. On the purely algebraic data-sparse approximation of the inverse and the triangular factors of sparse matrices. *Numerical Linear Algebra with Applications*, 18(1):105–122, 2011.
5. Mario Bebendorf and Florian Kramer. Hierarchical matrix preconditioning for low-frequency–full-maxwell simulations. *Proceedings of the IEEE*, 101(2):423–433, 2013.
6. Mario Bebendorf and Sergej Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.
7. Steffen Börm. $\mathcal{H}^2$-matrix arithmetics in linear complexity. *Computing*, 77(1):1–28, 2006.
8. Steffen Börm, Lars Grasedyck, and Wolfgang Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405–422, 2003.



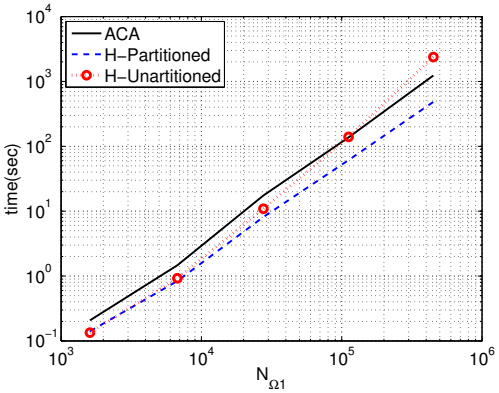
(a) $m = 0$, adaptively refined meshes.



(b) $m = 1$, adaptively refined meshes



(c) $m = 0$, regularly refined meshes



(d) $m = 1$, regularly refined meshes

Figure 15. Time required to compute the $\mathcal{H}$-matrix approximation of $\mathbf{S}$. This figure illustrates the time required to compute entries of $\mathbf{S}$ by MatLab sparse calculations (ACA), and the time required to calculate $\mathcal{H}$-matrix through $\mathcal{H}$-arithmetic approach when the rectangular matrices are partitioned (H-Partitioned) and not partitioned (H-Unpartitioned)

9. Roberto D Graglia, Donald R Wilton, and Andrew F Peterson. Higher order interpolatory vector bases for computational electromagnetics. *Antennas and Propagation, IEEE Transactions on*, 45(3):329–342, 1997.
10. Lars Grasedyck and Wolfgang Hackbusch. Construction and arithmetics of h-matrices. *Computing*, 70(4):295–334, 2003.
11. Wolfgang Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. part i: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.
12. Wolfgang Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*. Springer, 2016.
13. George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.
14. Gerd Kaupp. *Atomic force microscopy, scanning nearfield optical microscopy and nanoscratching: application to rough and natural surfaces*. Springer Science & Business Media, 2006.
15. Patrick Lacoste. Solution of maxwell equation in axisymmetric geometry by fourier series decompostion and by use of h (rot) conforming finite element. *Numerische Mathematik*, 84(4):577–609, 2000.
16. Jin-Fa Lee, GM WIKLINS, and Raj Mittra. Finite-element analysis of axisymmetric cavity resonator using a hybrid edge element technique. *IEEE transactions on microwave theory and techniques*, 41(11):1981–1987, 1993.
17. Haixin Liu and Dan Jiao. Existence of-matrix representations of the inverse finite-element matrix of electrodynamic problems and-based fast direct finite-element solvers. *Microwave Theory and Techniques, IEEE Transactions on*, 58(12):3697–3709, 2010.
18. Peter Monk. *Finite element methods for Maxwell's equations*. Oxford University Press, 2003.
19. Jörg Ostrowski, Zoran Andjelic, Mario Bebendorf, Bogdan Cranganu-Cretu, and J Smajić. Fast bem-solution of laplace problems with h-matrices and aca. *Magnetics, IEEE Transactions on*, 42(4):627–630, 2006.
20. Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

21. Dieter W Pohl, W Denk, and M Lanz. Optical stethoscopy: Image recording with resolution $\lambda/20$. *Applied physics letters*, 44(7):651–653, 1984.
22. Jasmin Smajic, Zoran Andjelic, and Mario Bebendorf. Fast bem for eddy-current problems using h-matrices and adaptive cross approximation. *Magnetics, IEEE Transactions on*, 43(4):1269–1272, 2007.
23. Ting Wan, Zhao Neng Jiang, and Yi Jun Sheng. Hierarchical matrix techniques based on matrix decomposition algorithm for the fast analysis of planar layered structures. *Antennas and Propagation, IEEE Transactions on*, 59(11):4132–4141, 2011.
24. Bangda Zhou and Dan Jiao. Direct finite-element solver of linear complexity for large-scale 3-d electromagnetic analysis and circuit extraction. *Microwave Theory and Techniques, IEEE Transactions on*, 63(10):3066–3080, 2015.