

Data Sparse Numerical Models for SNOM Tips

C. Hafner and R. Hiptmair and P. Souzangar

Research Report No. 2015-26
August 2015

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

Data Sparse Numerical Models for SNOM Tips

Christian Hafner¹, Ralf Hiptmair², 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 matrix compression for the efficient numerical modeling of geometrically persistent parts in large-scale electromagnetic simulations. Our approach relies on local low-rank representation in the framework of the \mathcal{H} -matrix data sparse matrix storage format. We discuss two ways to build approximate \mathcal{H} -matrix representations of dense Schur-complement matrices: Adaptive cross approximation (ACA) and \mathcal{H} -arithmetics. We perform profound numerical test and comparisons in an axi-symmetric setting, employing the open source \mathcal{H} -matrix library AHMED by M. Bebendorf. As application we have in mind the direct simulation of SNOM imaging devices, where the tip geometry remains unchanged for many simulation runs.

Date: August 26, 2015

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

1. INTRODUCTION

1.1. SNOM tips

Scanning Near-field Optical Microscopy, SNOM, first introduced by Pohl et al. [15], 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.

For numerical tip-sample simulations, one should notice that the tip can be 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 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

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

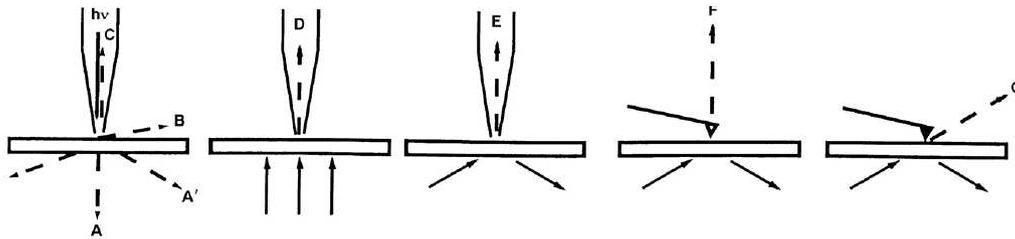


Figure 1. The various principles of SNOM techniques; aperture SNOM (A, A', B, C) with metal-coated light sources, D and E with separate source; apertureless SNOM by light tunneling (F, G), [10, Figure. 2.1]

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 it 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 tip on a 2D 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 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 computed with high resolution, the model matrix, i.e., the Schur complement of the FE matrix is generally huge and naturally dense. Therefore, we need to compress the matrix to avoid high memory consumption. For this reason, we aim to calculate a local low-rank approximation of the matrix in \mathcal{H} -matrix format [6–8]. The idea is supported by [4, Theorem 3.4] which states that an approximation of the Schur complement of a sparse matrix by an \mathcal{H} -matrix does exist.

The \mathcal{H} -matrix format is based on an admissible partitioning [3, Definition 1.15] of the matrix. The construction of this partitioning relies on,

- (I) a cluster tree [3, Definition 1.16] of row (or column) indices of the matrix, and
- (II) a block cluster tree [3, Sec. 1.5] based on the cluster tree and an admissibility condition [3, Sect. 1.3]. The block cluster tree eventually defines the partitioning of the matrix.

After matrix partitioning, the so-called admissible blocks are stored in low-rank format and the inadmissible blocks are stored as dense matrices. The inadmissible blocks should be reasonably small with the maximum size of b_{min} , which is set before clustering of DoFs.

1.3. Objective of the report

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).

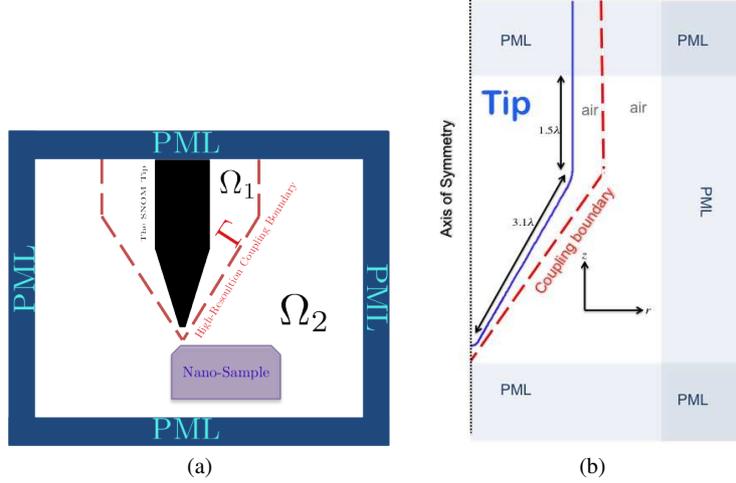


Figure 2. a) Homogeneous domain decomposition for tip-sample configurations. Ω_1 is the fixed part of imaging simulations. b) Metallic tip model.

- (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. MODELING OF THE 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,

$$\Gamma = \partial\bar{\Omega}_1 \cap \partial\bar{\Omega}_2 \quad (1a)$$

$$\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma. \quad (1b)$$

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,

$$\nabla \times \mathbf{E} = i\omega\mu\mathbf{H} \quad (2a)$$

$$\nabla \times \mathbf{H} = -i\omega\varepsilon\mathbf{E}, \quad (2b)$$

where μ and ε are the spatially 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 [13, Section 4.1]),

$$\begin{aligned} & \int_{\Omega_1} \langle \nabla \times \mathbf{E}_1, \nabla \times \mathbf{E}' \rangle - k^2 \langle \mathbf{E}_1, \mathbf{E}' \rangle d\Omega \\ & + \int_{\partial\Omega_1 \setminus \Gamma} \langle \boldsymbol{\nu} \times (\nabla \times \mathbf{E}_1), \mathbf{E}'_{\mathbf{T}} \rangle dA + \int_{\Gamma} \langle \boldsymbol{\nu} \times (\nabla \times \mathbf{E}_1), \mathbf{E}'_{\mathbf{T}} \rangle dA \\ & = \int_{\Omega_1} \langle \mathbf{F}_s, \mathbf{E}' \rangle d\Omega \end{aligned} \quad (3a)$$

$$\begin{aligned} & \int_{\Omega_2} \langle \nabla \times \mathbf{E}_2, \nabla \times \mathbf{E}' \rangle - k^2 \langle \mathbf{E}_2, \mathbf{E}' \rangle d\Omega \\ & + \int_{\partial\Omega_2 \setminus \Gamma} \langle \boldsymbol{\nu} \times (\nabla \times \mathbf{E}_2), \mathbf{E}'_{\mathbf{T}} \rangle dA - \int_{\Gamma} \langle \boldsymbol{\nu} \times (\nabla \times \mathbf{E}_2), \mathbf{E}'_{\mathbf{T}} \rangle dA \\ & = \int_{\Omega_2} \langle \mathbf{F}_s, \mathbf{E}' \rangle d\Omega, \end{aligned} \quad (3b)$$

in combination with the following transmission conditions on Γ (note that the the coupling boundary is entirely located in air region),

$$\mathbf{E}_1 \times \boldsymbol{\nu} = \mathbf{E}_2 \times \boldsymbol{\nu} \quad \text{on } \Gamma \quad (4a)$$

$$\nabla \times \mathbf{E}_1 \times \boldsymbol{\nu} = \nabla \times \mathbf{E}_2 \times \boldsymbol{\nu} \quad \text{on } \Gamma, \quad (4b)$$

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.

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

2.2. Axisymmetric Formulations

We assume that the tip is rotationally symmetric. Thus, we opt for the cylindrical coordinates system in the tip domain, Ω_1 . We denote the cylindrical coordinates 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, we assume that the axis of rotation is along $\hat{\mathbf{z}}$.

We expand the electric field in Ω_1 into a Fourier series with respect to the azimuthal angle ϕ ,

$$\mathbf{E}_1(r, \phi, z) = \sum_{m=0}^{\infty} \mathbf{E}_p^m(r, z) \cos(m\phi) + E_{\phi}^m(r, z) \sin(m\phi) \hat{\boldsymbol{\phi}} \quad (5)$$

where $E_{\phi}^m(r, z)$ is a scalar function and $\mathbf{E}_p^m(r, z) = E_r^m \hat{\mathbf{r}} + E_z^m \hat{\mathbf{z}}$ is a vector field [11, 12]. We refer to the m th mode in the Fourier series by $\mathbf{E}^m = \mathbf{E}_p^m \cos(m\phi) + E_{\phi}^m \sin(m\phi) \hat{\boldsymbol{\phi}}$.

Thanks to the axisymmetric feature of the tip, the 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 the Fourier series (5) to a short finite sum.

For each mode in the Fourier series, the ϕ dependency of the field is analytically known, which means that one can simulate the fields only in a cross section of the tip in 2D. To do this, we reformulate (3a) for axisymmetric cases.

First, we define the following planar operators [12],

$$\nabla_p V := \partial_r V \hat{\mathbf{r}} + \partial_z V \hat{\mathbf{z}} \quad (6a)$$

$$\nabla_p \cdot \mathbf{V}_p := \partial_r V_r + \partial_z V_z \quad (6b)$$

$$\nabla_p \times \mathbf{V}_p := (\partial_z V_r - \partial_r V_z) \hat{\phi}, \quad (6c)$$

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 vector field. One can decompose any vector \mathbf{V} into the planar and azimuthal components: $\mathbf{V} = \mathbf{V}_p + V_\phi \hat{\phi}$. The vector operators in cylindrical coordinates are written as follows,

$$\nabla V = \frac{1}{r} \partial_\phi V \hat{\phi} + \nabla_p V \quad (7a)$$

$$\nabla \cdot \mathbf{V} = \frac{1}{r} \partial_\phi V_\phi + \frac{1}{r} \nabla_p (r \mathbf{V}_p) \quad (7b)$$

$$\nabla \times \mathbf{V} = \nabla_p \times \mathbf{V}_p + \mathbf{P} \frac{1}{r} (\partial_\phi \mathbf{V}_p - \nabla_p (r V_\phi)), \quad (7c)$$

where $\mathbf{P} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

We rewrite (3a) in terms of \mathbf{E}^m and the vector operators in (7). In addition, we define $q_\phi^m := r E_\phi^m$ and solve the equations for this new unknown, see [12]. Assuming $\mathbf{F}_s = 0$, solving (3a) for \mathbf{E}^m boils down to the following problem:

We seek,

$$\mathbf{E}_p^m, q_\phi^m \in L^2(\Omega_{p1})$$

such that

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

and,

$$\begin{aligned} & \int_{\Omega_{p1}} \frac{1}{r} \langle m \mathbf{E}_p^m + \nabla_p q_\phi^m, m \mathbf{E}_p^m + \nabla_p q_\phi^m \rangle d\Omega_p \\ & + \int_{\Omega_{p1}} r \langle \nabla_p \times \mathbf{E}_p^m, \nabla_p \times \mathbf{E}_p^m \rangle d\Omega_p \\ & - k^2 \int_{\Omega_{p1}} r \langle \mathbf{E}_p^m, \mathbf{E}_p^m \rangle + \frac{1}{r} \langle q_\phi, q_\phi' \rangle d\Omega_p \\ & + \int_{\partial\Omega_{p1}} \langle \boldsymbol{\nu} \times (\nabla_p \times \mathbf{E}_p^m + \frac{1}{r} \mathbf{P} (m \mathbf{E}_p^m - \nabla_p q_\phi^m)), (\mathbf{E}_p^m + \frac{q_\phi^m}{r} \hat{\phi})_{\mathbf{T}} \rangle dl \\ & = \mathbf{0}, \end{aligned} \quad (8)$$

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

where $\Omega_1 = \Omega_{p1} \times [0, 2\pi]$.

By reducing the computational domain from Ω_1 to Ω_{p1} , the axis of rotation which is inside Ω_1 coincides with a part of $\partial\Omega_{p1}$. 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,

$$m \mathbf{E}_p^m(0, z) + \nabla_p q_\phi^m(0, z) = 0. \quad (9)$$

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:

$$\begin{cases} m > 0 : & \text{homogeneous Dirichlet boundary condition,} \\ m = 0 : & \text{homogeneous Neumann boundary condition.} \end{cases} \quad (10)$$

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 lowest order polynomial conforming FEM on a triangular mesh in $\Omega_p \in \mathbb{R}^2$. We have chosen the FEM basis functions according to [12]. For q_ϕ the local piecewise linear nodal functions (also known as hat functions), are used. The nodal basis functions for 2D edge elements (also known as RWG basis functions [16]), are chosen for \mathbf{E}_p .

The FE matrices are obtained based on the bilinear form in (8). The final FE system of equations is,

$$\begin{bmatrix} \mathbf{A}_{ee} & \mathbf{A}_{ev} \\ \mathbf{A}_{ve} & \mathbf{A}_{vv} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_e \\ \boldsymbol{\mu}_v \end{bmatrix} = \begin{bmatrix} \mathbf{f}_e \\ \mathbf{f}_v \end{bmatrix}, \quad (11)$$

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

As mentioned earlier, 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 3D electromagnetic fields the following electric to magnetic field operator C is defined [13, Section 9.4] 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, 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):

$$\begin{bmatrix} \mathbf{A}_{in,in} & \mathbf{A}_{in,\Gamma} & \mathbf{0} \\ \mathbf{A}_{\Gamma,in} & \mathbf{A}_{\Gamma,\Gamma}^{in} + \mathbf{A}_{\Gamma,\Gamma}^{out} & \mathbf{A}_{\Gamma,out} \\ \mathbf{0} & \mathbf{A}_{out,\Gamma} & \mathbf{A}_{out,out} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_{in} \\ \boldsymbol{\mu}_\Gamma \\ \boldsymbol{\mu}_{out} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{in} \\ \mathbf{f}_\Gamma \\ \mathbf{f}_{out} \end{bmatrix}, \quad (12)$$

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 $\mathbf{f}_\Gamma = 0$,

$$\begin{bmatrix} \mathbf{A}_{\Gamma,\Gamma}^{out} + \mathbf{S} & \mathbf{A}_{\Gamma,out} \\ \mathbf{A}_{out,\Gamma} & \mathbf{A}_{out,out} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_\Gamma \\ \boldsymbol{\mu}_{out} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{tip} \\ \mathbf{f}_{out} \end{bmatrix}, \quad (13)$$

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

$$\mathbf{S} = -\mathbf{A}_{\Gamma,in}\mathbf{A}_{in,in}^{-1}\mathbf{A}_{in,\Gamma} + \mathbf{A}_{\Gamma,\Gamma}^{in} \quad (14a)$$

$$\mathbf{f}_{tip} = -\mathbf{A}_{\Gamma,in}\mathbf{A}_{in,in}^{-1}\mathbf{f}_{in}. \quad (14b)$$

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 (5). Therefore, we want to represent \mathbf{S} in the data sparse \mathcal{H} -matrix-format, see Section 4 and Section 5.

3. A MODEL PROBLEM

For all numerical experiments in sections 4 and 5, we consider an axisymmetric metallic tip with permittivity $\varepsilon_r = -10.155 + 2.1281e - 08i$ placed in air and illuminated by a dipole located on the

axis of symmetry. The cross section of the tip is shown in Figure 2b. The apex radius and shaft radius of the tip are $\lambda/10$ and $\lambda/2$ respectively. The tip angle is $\pi/6$ but the angle of the coupling cone is chosen a little wider (35°) 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 (5), respectively.

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

$$\mathbf{A} = \frac{\mu_0}{4\pi\sqrt{r^2+z^2}} e^{ik\sqrt{r^2+z^2}} \delta l \hat{\mathbf{z}} \quad (15)$$

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

$$\begin{aligned} \mathbf{A} &= \frac{\mu_0}{4\pi\sqrt{r^2+z^2}} e^{ik\sqrt{r^2+z^2}} \delta l \hat{\mathbf{x}} \\ &= \frac{\mu_0}{4\pi\sqrt{r^2+z^2}} e^{ik\sqrt{r^2+z^2}} \delta l (\cos(\phi)) \hat{\mathbf{r}} - \sin(\phi) \hat{\phi} \end{aligned} \quad (16)$$

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.

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. FE Meshes

To investigate the efficiency of the compression of the model matrix \mathbf{S} , we perform our experiments 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 $\mathbf{A}_{\text{in,in}}$ in (12).

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 (11) 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 (11). 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 3rd mesh in Table I.

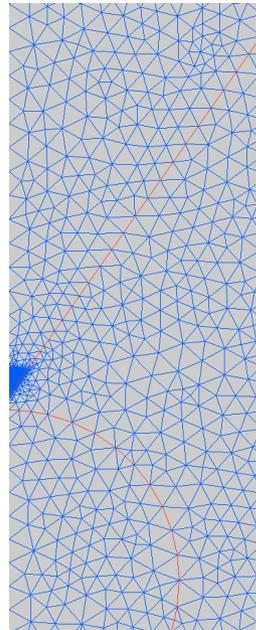
In the following sections we explain how to build an \mathcal{H} -matrix approximation of \mathbf{S} .

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

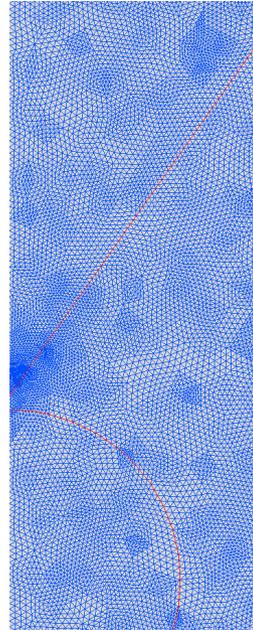
In this section, we compress the model matrix \mathbf{S} by means of ACA algorithm [1, 5]. First, \mathbf{S} is calculated through (14a) using arithmetics of sparse matrices and sparse Gaussian elimination.

Mesh	Adaptive				Regular			
	m=0		m=1		m=0		m=1	
	N_Γ	N_{Ω_1}	N_Γ	N_{Ω_1}	N_Γ	N_{Ω_1}	N_Γ	N_{Ω_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 of DoFs for adaptively and regularly refined meshes generated by MatLab for the configuration of Figure 2b. In this table N_{Ω_1} refers to the dimension of $\mathbf{A}_{\text{in},\text{in}}$ and N_Γ refers to the size of the schur complement matrix \mathbf{S} , see (14a).



(a) Adaptively refined mesh



(b) Regularly refined mesh

Figure 3. Illustration of the mesh 3 in Table I. Dipole is located about 1.51λ below the apex of the tip.

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, we assemble the FE matrix according to (11). Then, we use the sparse direct solver of MatLab to compute *all the entries* of \mathbf{S} through (14a).

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 , we adopt a geometrical clustering approach [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.

We use the `genClusterTree2d_pca` routine implemented in AHMED to build the geometrical cluster tree based on the PCA algorithm [14]. 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 [3, eq. 4.11]. We use the `genBlockClusterTree` routine implemented in AHMED for this purpose.

The admissible blocks [3, Sect. 1.3] 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.

The ACA algorithm finds low rank approximations of admissible blocks based on the entries of \mathbf{S} for a predefined error. The predefined error is a control parameter of the relative error of the low rank approximation and is used as a stopping criterion for the ACA algorithm. Another stopping criterion might be the maximum rank of the approximated block, which is not used in our experiments. The cost of the ACA algorithm is of order of $k^2(m+n)$ where k is the rank of the admissible block and $m \times n$ is the block dimension.

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. 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.

Figure 4 shows the compression rate of \mathbf{S} . The compression rate is defined by:

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

where M_{dense} is the memory needed to save a dense matrix and $M_{\text{compressed}}$ is the memory needed to save its \mathcal{H} -matrix approximation. This figure shows that 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.

Figure 5 shows the relative Euclidean norm (computed by the `norm`-function of MatLab) of the approximation error versus the ACA error control parameter. This figure proves that the error parameter of the ACA algorithm is a reliable control for the accuracy of the compressed model matrix.

Finally, we examine the effect of the compression on the FE solution in the outer domain, Ω_2 in Figure 2a. The electrical field computed from (13) in Ω_2 by FEM is considered to 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 is shown in Figure 6. Clearly, a more accurate solution in the outer domain is achieved where the approximation of \mathbf{S} is more accurate.

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

In this section, we adopt another strategy 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 (14) 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".

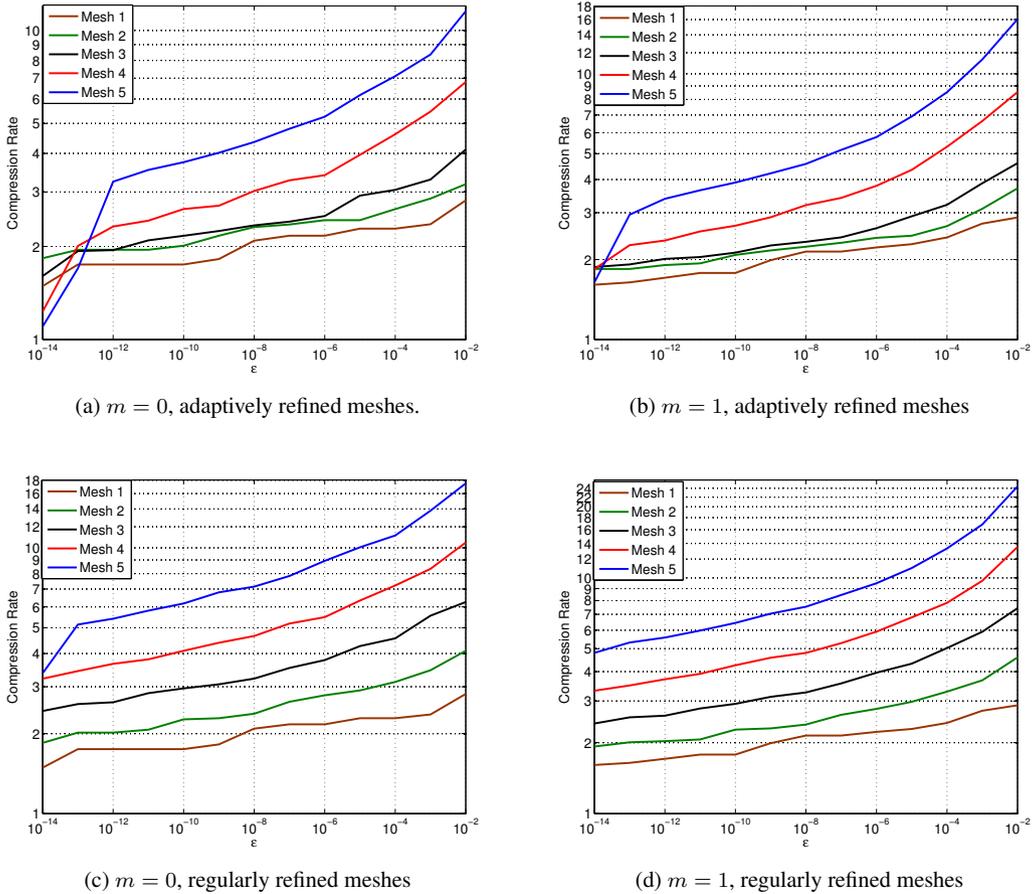


Figure 4. Compression rate for the model matrix \mathbf{S} vs the error control parameter ϵ and for $b_{min} = 5$ for different meshes (see Table I) by using ACA.

5.1. Partitioning Based on the Matrix Graph

Reconsidering (14a), 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, we first partition them as described below.

We adopt the same systematic procedure of partitioning mentioned in Section 1.2. On the contrary to the clustering in Section 4.1, the cluster and block cluster tree are built through matrix graph based clustering.

In the \mathcal{H} -arithmetic approach, we partition the undirected matrix graphs associated to each of the symmetric sparse matrices in (14a). 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 [9], 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 algebraic admissibility condition [3, eq. 4.50].

Partitioning of the Rectangular Matrices. Obviously, the idea of clustering of the matrix indices based on the undirected matrix graph is applicable only to symmetric matrices, whereas the matrices $\mathbf{A}_{\Gamma, in}$ and $\mathbf{A}_{in, \Gamma}$ in (14a) are not even square. Furthermore, in these rectangular matrices

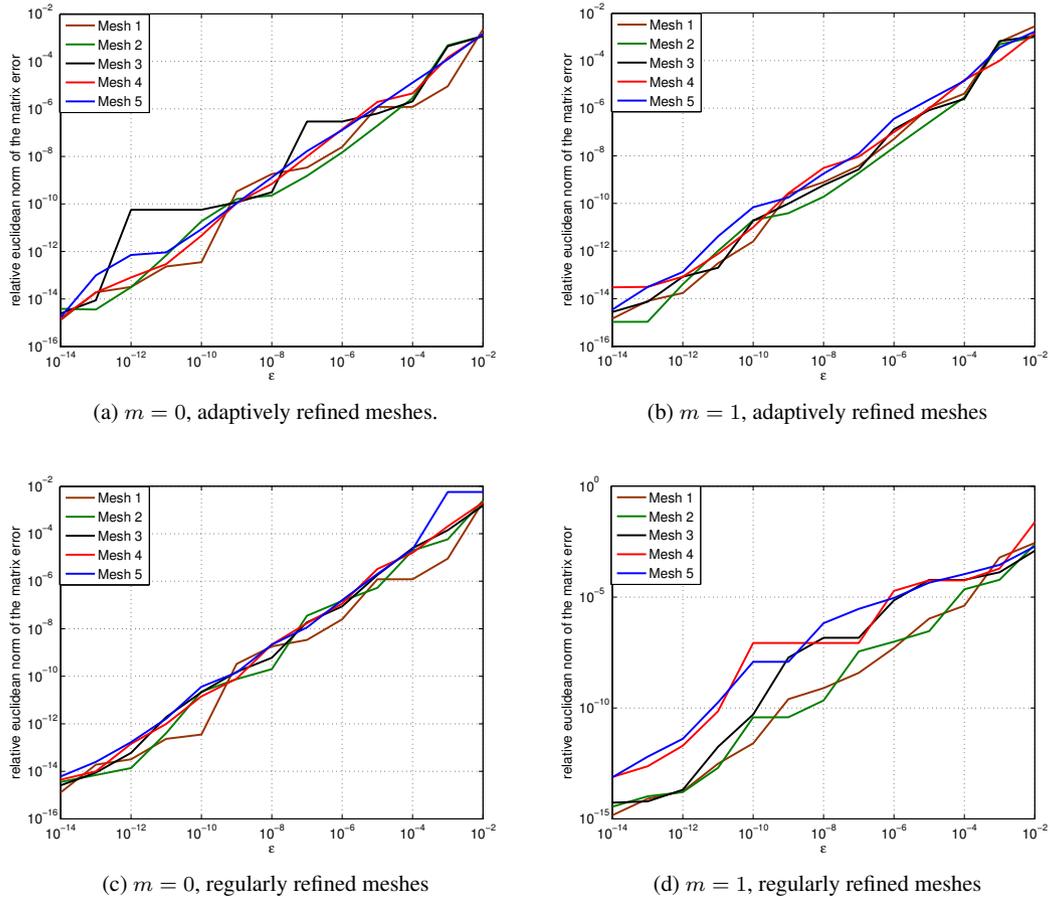


Figure 5. Relative Euclidean error norm of the compressed model matrix vs the error control parameter ε and for $b_{min} = 5$ for different meshes (see Table I) by using ACA.

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 (14a) as normal 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, same as for ACA, 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 10.

Remarks on the case $m = 0$. As mentioned above, in the experiments for the \mathcal{H} -arithmetic approach we build the algebraic cluster trees only for $\mathbf{A}_{in,in}$ and $\mathbf{A}_{\Gamma,\Gamma}^{in}$ in (14a). In the case of $m = 0$, as discussed in Section 3.2, the compression is only applied to \mathbf{A}_{ee} in (11). Since the supports of the DoFs on the edges of the coupling boundary do not share an element, the matrix $\mathbf{A}_{\Gamma,\Gamma}^{in}$ 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 $\mathbf{A}_{\Gamma,\Gamma}^{in}$ based on the matrix graph partitioning fails. In this case, we add some very small entries (in the order of the machine precision) to $\mathbf{A}_{\Gamma,\Gamma}^{in}$ to indicate which edges share a vertex on the coupling boundary.

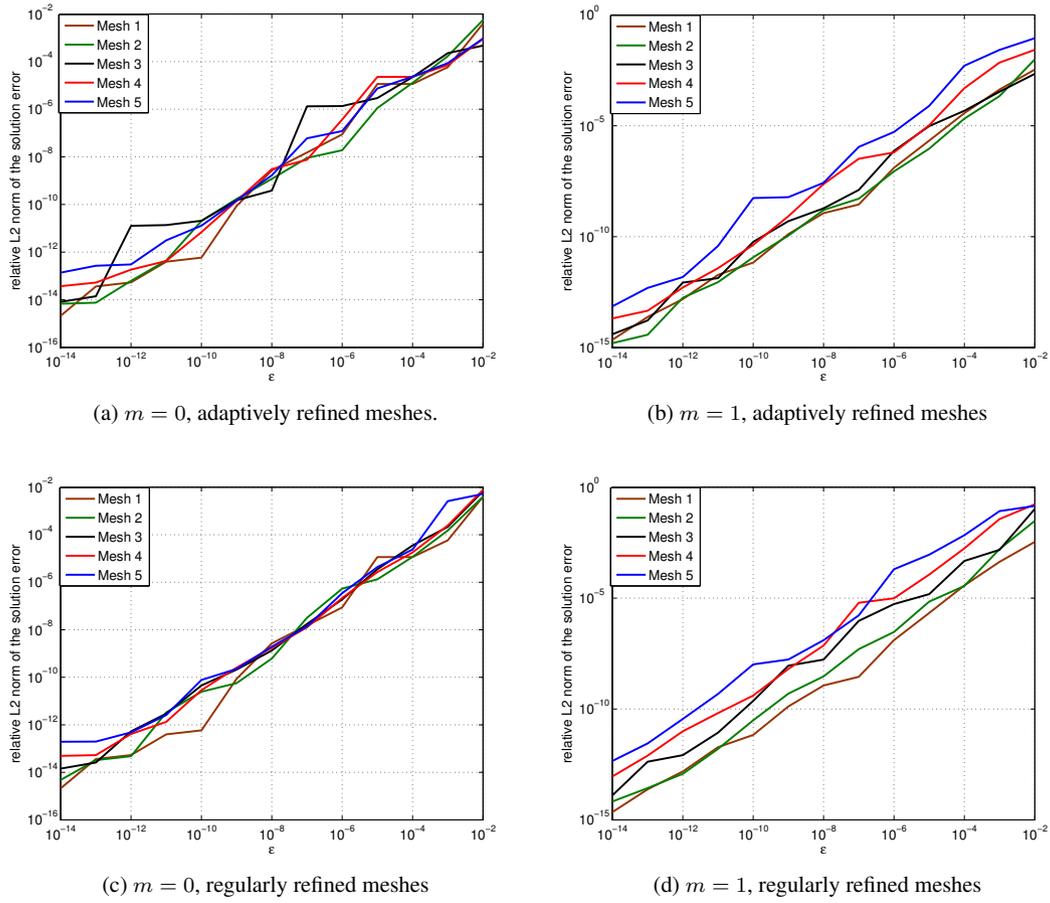


Figure 6. Convergence of the L_2 norm of the solution error in Ω_2 when $b_{min} = 5$. Here, ϵ is the ACA error control parameter.

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 diagonal entries of $\mathbf{A}_{in,in}$ and $\mathbf{A}_{\Gamma,\Gamma}^{in}$ 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 (14) 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 $\mathbf{A}_{in,in}^{-1}$, which is the only huge and dense matrix in (14a), we compute hierarchical LU factorization (H-LU),

$$\mathbf{A}_{in,in} \approx \mathbf{L}_H \mathbf{U}_H \quad (17)$$

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

Remark 1

Since $\mathbf{A}_{in,in}^{-1}$ is the only full matrix in (14a) and also the biggest one, one may come up with the idea of storing only this matrix as its LU factors, namely \mathbf{L}_H and \mathbf{U}_H in (17), and use matrix-vector

multiplication to acquire the entries of \mathbf{S} . However, this approach is not promising, because, the size of $\mathbf{A}_{\text{in},\text{in}}$ can grow quadratically 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 (14a) (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 (14) is the following,

- (I) Compute $\mathbf{L}_{\mathbf{H}}$ and $\mathbf{U}_{\mathbf{H}}$ (by using the `HLU` routine of AHMED),
- (II) Forward substitution: $\mathbf{L}_{\mathbf{H}}\mathbf{X} = \mathbf{A}_{\text{in},\Gamma}^{\mathbf{H}}$ (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: $\mathbf{Z} = -\mathbf{A}_{\Gamma,\text{in}}^{\mathbf{H}}\mathbf{Y}$ (by using the `mltaGeHGeH` routine of AHMED),
- (V) \mathcal{H} -addition: $\mathbf{S} = \mathbf{A}_{\Gamma,\Gamma}^{\mathbf{H}} + \mathbf{Z}$ (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 ε 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, $b_{\text{min}}=5$ for all \mathbf{S} compression experiments.

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 7 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 8 shows the relative Euclidean error norm 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 \mathcal{H} -arithmetic is reliable for both cases $m = 0$ and $m = 1$.

Figure 9 shows the L_2 norm of the solution error computed in the same way as discussed in the ACA-based approach in Section 4.2.

6. CONCLUSION

In this work, we obtained a data sparse \mathcal{H} -matrix-format of a finite element tip model matrix, \mathbf{S} , with controlled accuracies through two approaches.

Table I shows that N_{Ω_1} grows quadratically with respect to N_{Γ} . Since the complexity of the ACA algorithm is $O(N_{\Gamma} \log(N_{\Gamma}))$ (see [3, Sect. 3.4.4]), the complexity of the computations in the ACA-based approach is dominated by the complexity of the computation of the entries of \mathbf{S} from (14a). This computation involves LU-factorization of $\mathbf{A}_{\text{in},\text{in}}$. In most practical situations the LU-factorization of sparse matrices arising from 2D FE discretizations can apparently 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 the computations of the ACA-based method is not better than $O(N_{\Omega_1}^{3/2})$ (note that $N_{\Omega_1} \sim N_{\Gamma}^2$).

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] and multiplication of \mathcal{H} -matrices [3, Theorem 2.26] and LU-factorization as discussed in Section 5.2. Without partitioning of the rectangular matrices, $\mathbf{A}_{\Gamma,\text{in}}$ and $\mathbf{A}_{\text{in},\Gamma}$ a complexity similar to that of the ACA-based approach is expected. However, by partitioning the rectangular matrices in (14a) we may optimally expect almost linear complexity for the \mathcal{H} -arithmetic approach.

To compare the two approaches in our experiments, we measure the time needed for \mathcal{H} -arithmetics in the \mathcal{H} -arithmetic approach (for both partitioned and unpartitioned rectangular

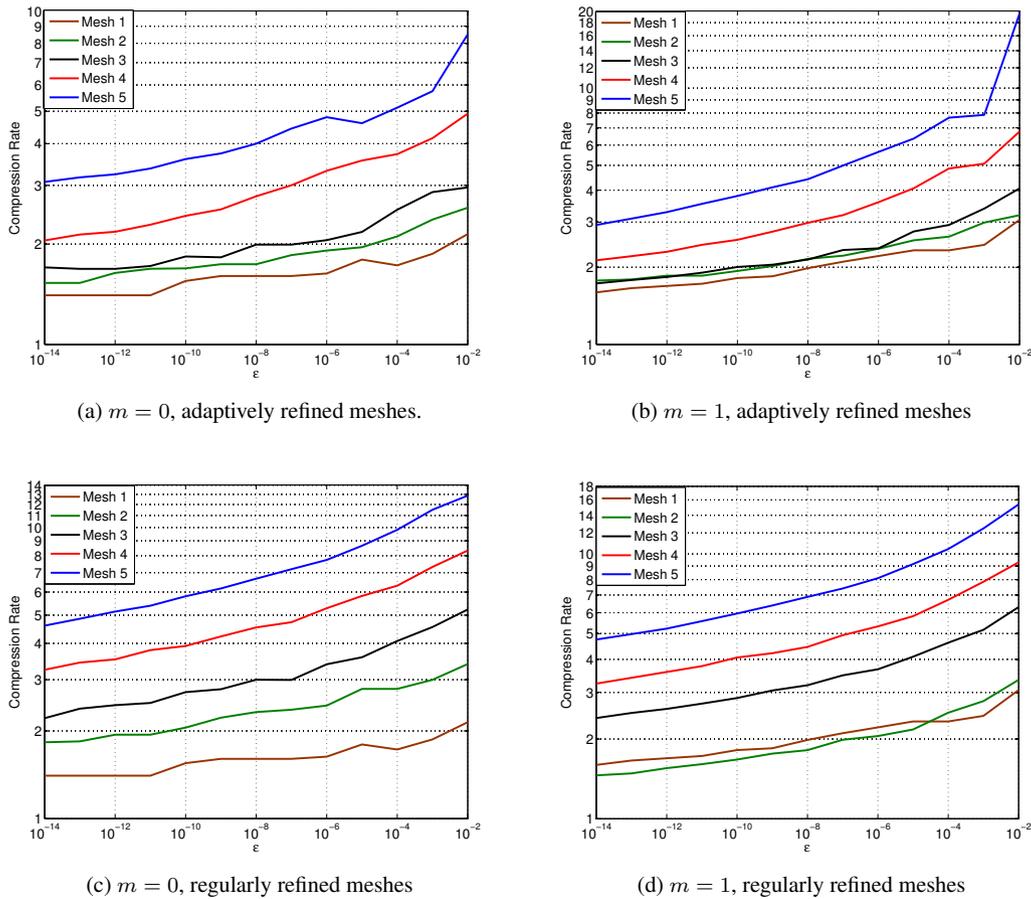


Figure 7. Compression rate of \mathbf{S} for matrices of different size (see Table I) using \mathcal{H} -arithmetics.

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 10. This figure shows slightly better computational complexity for the \mathcal{H} -arithmetic approach when rectangular matrices are partitioned.

Finally, the efficiency of compression for the two methods are shown in Figure 11. This figure shows the compression rates of \mathcal{H} -matrices obtained through the two approaches, the ACA-based approach and the \mathcal{H} -arithmetic one versus the relative Euclidean error norm of the compressed matrix, ϵ . One can conclude that the ACA-based method shows slightly better compression efficiency specially in the case of adaptively refined meshes.

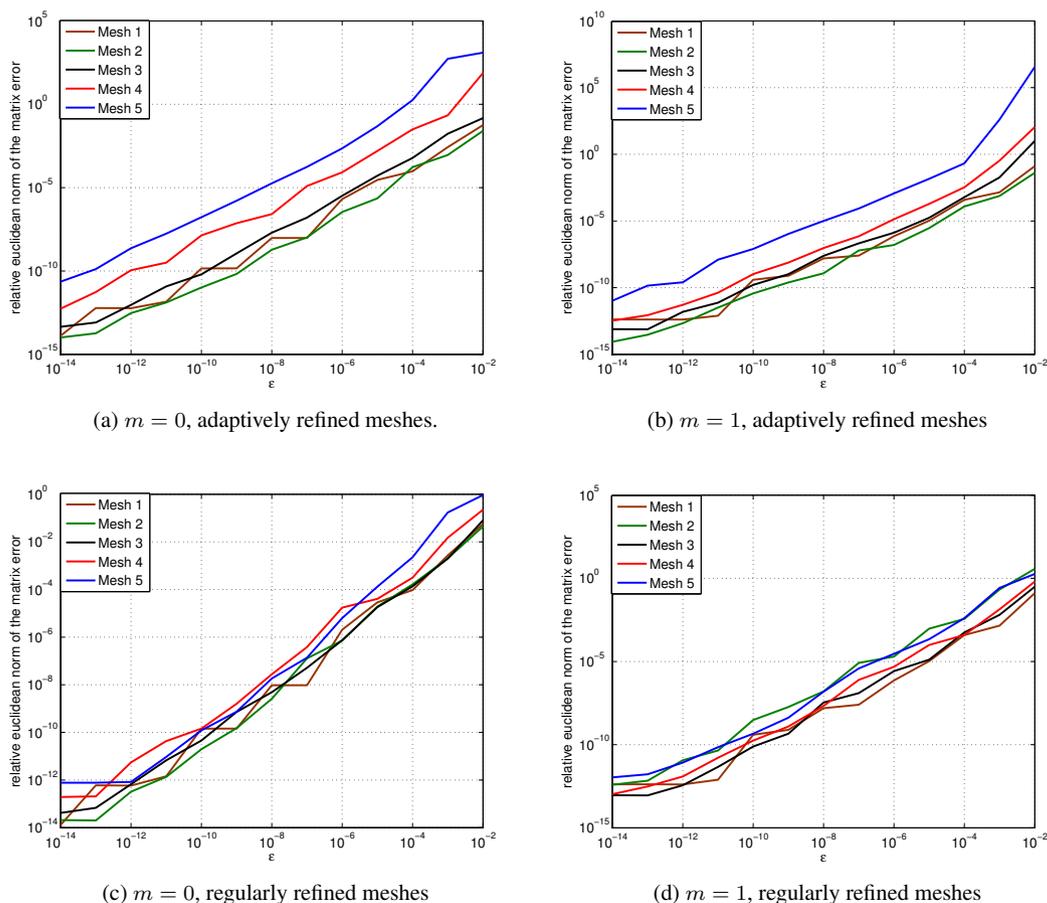


Figure 8. Relative Euclidean error norm of the compressed model matrix vs the error control parameter ϵ and $b_{min} = 5$ for different meshes (see Table I) using \mathcal{H} -arithmetics.

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 Sergej Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.
6. Steffen Börm, Lars Grasedyck, and Wolfgang Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405–422, 2003.
7. Lars Grasedyck and Wolfgang Hackbusch. Construction and arithmetics of h-matrices. *Computing*, 70(4):295–334, 2003.
8. Wolfgang Hackbusch. A sparse matrix arithmetic based on \mathcal{H} -matrices. part i: Introduction to \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
9. 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.
10. Gerd Kaupp. *Atomic force microscopy, scanning nearfield optical microscopy and nanoscratching: application to rough and natural surfaces*. Springer Science & Business Media, 2006.
11. Patrick Lacoste. Solution of maxwell equation in axisymmetric geometry by fourier series decomposition and by use of h (rot) conforming finite element. *Numerische Mathematik*, 84(4):577–609, 2000.
12. 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.
13. Peter Monk. *Finite element methods for Maxwell's equations*. Oxford University Press, 2003.

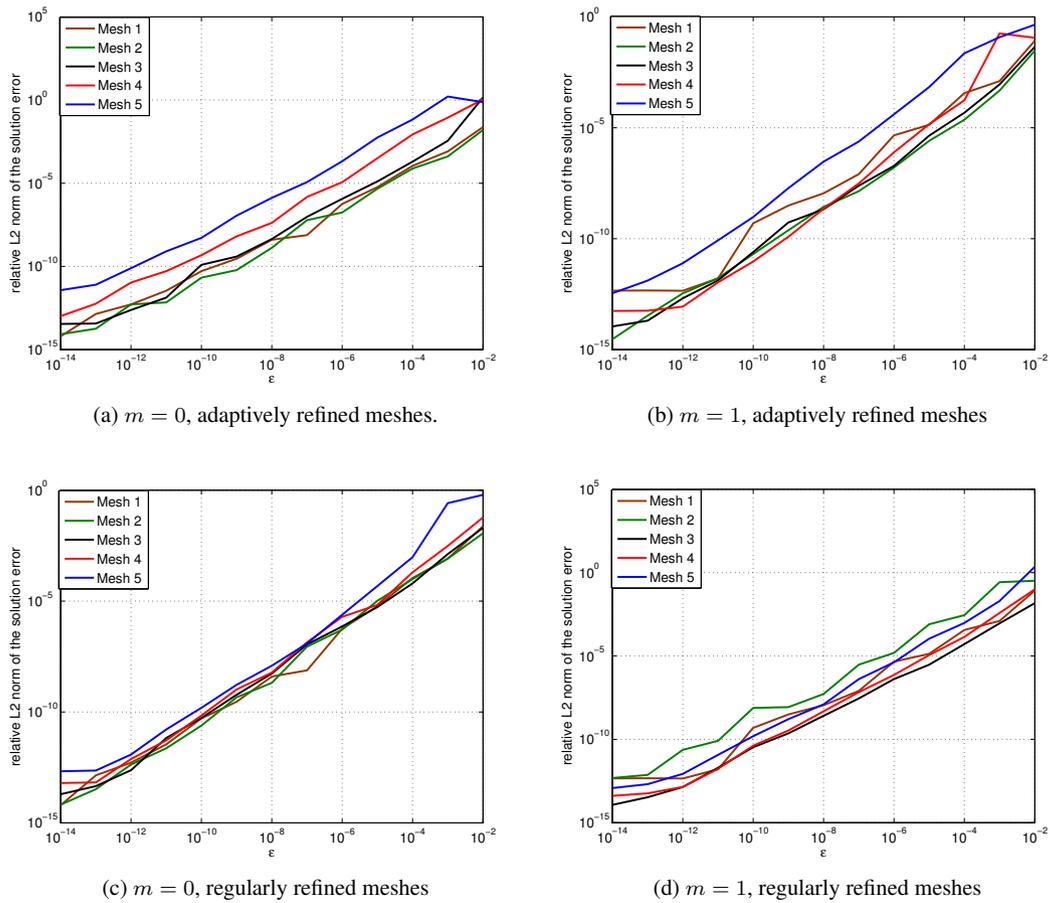


Figure 9. Convergence of the L_2 norm of the solution error in Ω_2 for $b_{min} = 5$. Here, ε is the error control parameter for \mathcal{H} -arithmetics.

14. 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.
15. Dieter W Pohl, W Denk, and M Lanz. Optical stethoscopy: Image recording with resolution $\lambda/20$. *Applied physics letters*, 44(7):651–653, 1984.
16. Sadasiva M Rao, Donald R Wilton, and Allen W Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *Antennas and Propagation, IEEE Transactions on*, 30(3):409–418, 1982.

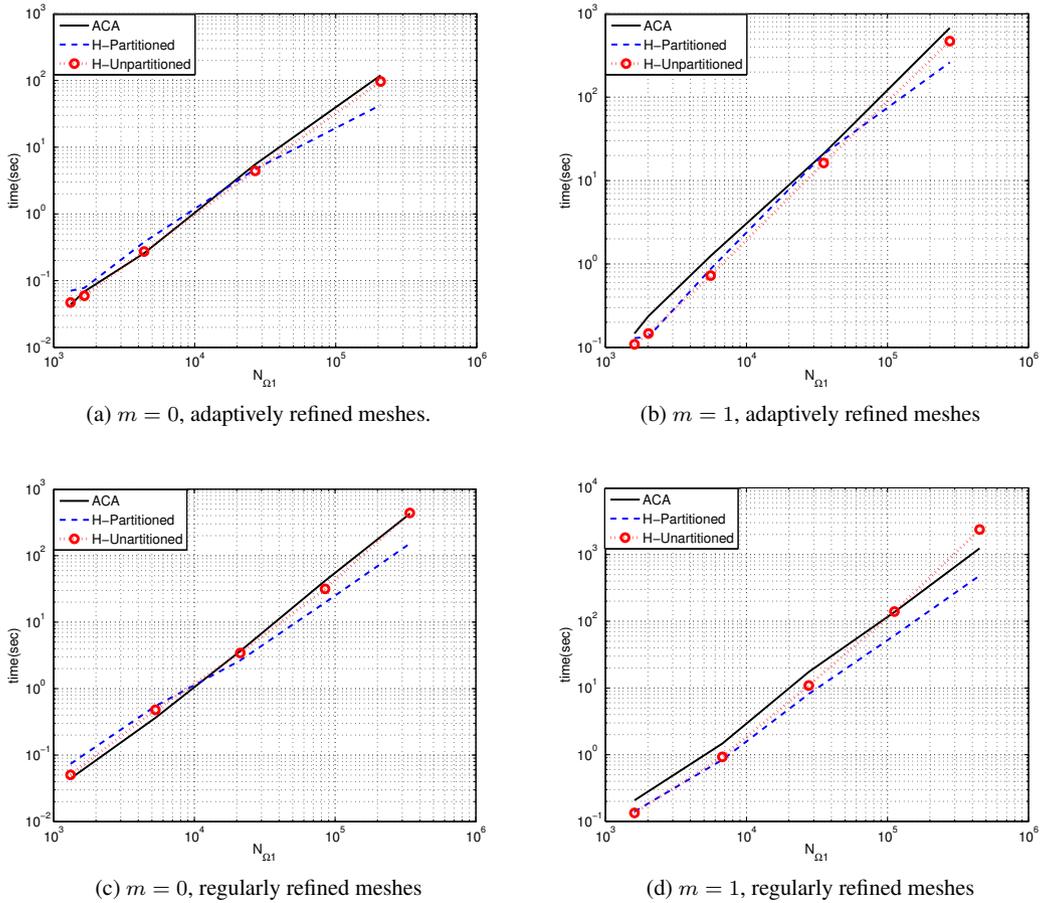
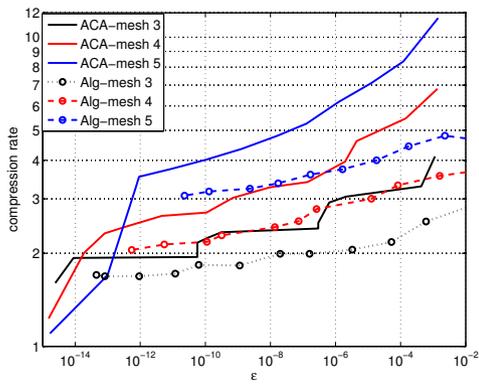
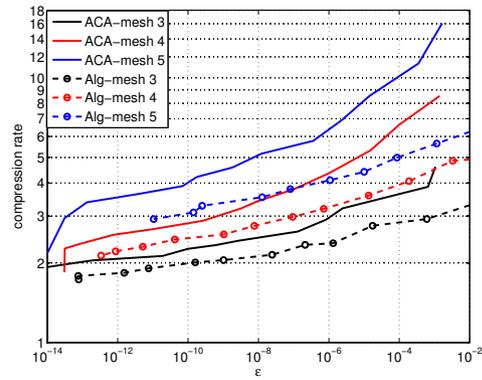


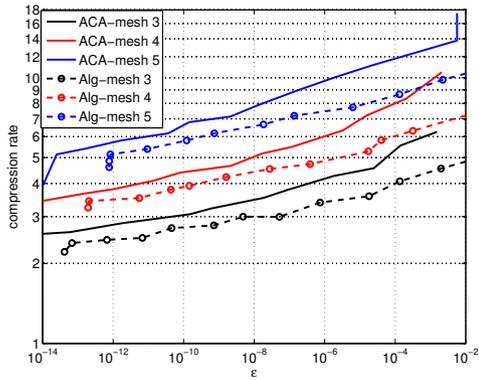
Figure 10. Time required to compute the \mathcal{H} -matrix approximation of \mathbf{S} . This figure illustrates the time required to compute entries of \mathbf{S} by MatLab sparse calculations (ACA), and the time required to calculate \mathcal{H} -matrix through \mathcal{H} -arithmetic approach when the rectangular matrices are partitioned (H-Partitioned) and not partitioned (H-Unpartitioned)



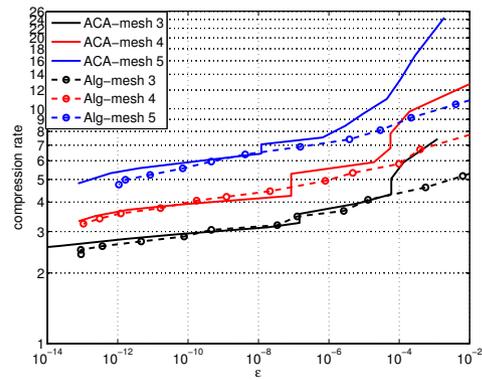
(a) $m = 0$, adaptively refined meshes.



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



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



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

Figure 11. Compression 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, ε .

Recent Research Reports

Nr.	Authors/Title
2015-16	V. Kazeev and Ch. Schwab Approximation of Singularities by Quantized-Tensor FEM
2015-17	P. Grohs and Z. Kereta Continuous Parabolic Molecules
2015-18	R. Hiptmair Maxwell's Equations: Continuous and Discrete
2015-19	X. Claeys and R. Hiptmair and E. Spindler Second-Kind Boundary Integral Equations for Scattering at Composite Partly Impenetrable Objects
2015-20	R. Hiptmair and A. Moiola and I. Perugia A Survey of Trefftz Methods for the Helmholtz Equation
2015-21	P. Chen and Ch. Schwab Sparse-Grid, Reduced-Basis Bayesian Inversion: Nonaffine-Parametric Nonlinear Equations
2015-22	F. Kuo and R. Scheichl and Ch. Schwab and I. Sloan and E. Ullmann Multilevel Quasi-Monte Carlo Methods for Lognormal Diffusion Problems
2015-23	C. Jerez-Hanckes and Ch. Schwab Electromagnetic Wave Scattering by Random Surfaces: Uncertainty Quantification via Sparse Tensor Boundary Elements
2015-24	V. Kazeev and Ch. Schwab Quantized tensor-structured finite elements for second-order elliptic PDEs in two dimensions