Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Accelerating the charge inversion algorithm with hierarchical matrices for gas insulated systems

To cite this article: F Lucchini and N Marconato 2021 *J. Phys.: Conf. Ser.* **2090** 012136

View the article online for updates and enhancements.

Accelerating the charge inversion algorithm with hierarchical matrices for gas insulated systems

F Lucchini[1] **and N Marconato**[2]

[1] Consorzio RFX (CNR, ENEA, INFN, Università degli Studi di Padova, Acciaierie Venete SpA), Corso Stati Uniti 4, 35127 Padova, Italy
[2] University of Padova, Department of Industrial Engineering, Italy

E-mail: francesco.lucchini@igi.cnr.it; nicolo.marconato@unipd.it

Abstract. Surface charges accumulating on dielectrics during long-time operation of Gas Insulated High Voltage Direct Current (HVDC-GIS) equipments may affect the stable operation and could possibly trigger surface flashovers. In industrial applications, to quantify and identify the location of the surface charge accumulation from experimental measurements, the surface potential distribution is evaluated using, e.g., electrostatic probes, then the charge density is determined by solving an electrostatic problem based on an inversion procedure known as Charge Inversion Algorithm. The major practical limitation of such procedure is the inversion and the storage of the fully dense matrix arising from the representation via Integral Equations of the electrostatic phenomenon, resulting in $\mathcal{O}(N^3)$ computational complexity and $\mathcal{O}(N^2)$ memory requirement. In this paper it is shown how hierarchical matrices can be efficiently used to accelerate the charge inversion algorithm and, more importantly, reduce the overall memory requirement.

1. Introduction

High Voltage Direct Current (HVDC) systems, because of their compact size, large-capacity transmission, and stable operation, have become of crucial importance in many applications [1, 2, 3]. Unlike High Voltage Alternating Current (HVAC) systems, for which the electric field is driven by the electrical permittivity of the materials, in HVDCs it depends both on permittivity and electrical conductivity [4, 5]. Moreover, for Gas Insulated HVDC (HVDC-GIS), the gas conductivity cannot be assigned a priori, but is affected by the motion of free gas ions due to electric field and diffusion [6].

A relevant aspect, specific of HVDC-GIS transmission, is the surface charge accumulating on dielectrics during the transition from the initial (capacitive) to the final (resistive) field distribution [4]. Charges accumulated on the dielectric surfaces are generally considered as the main factor that lowers the flashover voltage, by the local increase of the electric field [7, 8], and thus limits the practical applicability of HVDC-GIS systems.

Experimentally, the quantification of the charge density on dielectrics in HVDC-GIS is based on the measured surface potential distribution. Usually, electrostatic probes (e.g., Kelvin probes) are used to perform such measurements [9, 10]. This probing method does not require the physical contact with the measured surface, and the potential distribution is based on the compensation principle [10]. The *Charge Inversion Algorithm*, first introduced by Ootera *et. al.* [11], is usually adopted to retrieve the charge density from the potential measurements [12]. The main drawback of this method is the computational effort required for the storage and the inversion of the dense matrix which maps potentials to surface charges [13, 14]. However, as shown in this paper, this limitation can be overcome by exploiting *hierarchical matrices* ($\mathcal{H}$–matrices) [15, 16], which provide a data-sparse format to efficiently store and manipulate dense matrices arising from integral formulation [17].

In recent years, the interest on $\mathcal{H}$–matrices applied to Integral Equations (IE) methods [18, 19, 20, 21] for electromagnetic simulations has significantly increased. Indeed, as shown in [22, 23], $\mathcal{H}$–matrices allow for significantly reducing the overall computational effort required by IE methods and they have been used to alleviate the computational burden required by numerical studies in many different industrial applications, e.g., from wireless power transfer devices for electric vehicles [24, 25] to thermonuclear fusion applications [26, 27] or filters for electronic devices [28].

Despite the increasing use of $\mathcal{H}$–matrices in the context of IE methods, it seems that such techniques have not been adopted in HVDC-GIS research field yet. However, they would be very useful to solve, or at least alleviate, the computational issue arising from the need of storing and inverting the dense matrices arising in the Charge Inversion Algorithm.

This work aims to bridge the gap between these two reserach areas, with the aim of showing how $\mathcal{H}$–matrices coupled with low-rank compression techniques can be used to reduce the overall computational burden required by the Charge Inversion Algorithm in HVDC-GIS applications. In this article, different type of $\mathcal{H}$–matrices are used (i.e., $\mathcal{H}$ Off Diagonal Low Rank (HODLR), $\mathcal{H}$ Adaptive Low Rank (HALR) and $\mathcal{H}$–Semi–Separable (HSS) matrices). Moreover, depending on the case, different strategies are proposed, giving priority to the solution accuracy or to the reduction of the computational effort.

2. Charge Inversion Algorithm

In this section we briefly describe the Charge Inversion Algorithm which is used in HVDC-GIS applications to identify the charge distribution from measurements of the electric potential. The algorithm is based on the well-known electrostatic field equations [29]:

$$\nabla \cdot \mathbf{D} = \rho_{free}, \tag{1}$$

$$\mathbf{E} = -\nabla\varphi, \tag{2}$$

where φ is the electric potential, ρ_{free} is the free volumetric charge density [11] and $\mathbf{E}$ is the electric field. The electric displacement field $\mathbf{D}$ is given by

$$\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}, \tag{3}$$

where ε_0 is the vacuum electric permittivity and $\mathbf{P}$ is the polarization vector. Substituting (3) into (1) and defining the apparent charge density as

$$\rho_{app} = \rho_{free} + \rho_{bound}, \tag{4}$$

where $\rho_{bound} = -\nabla \cdot \mathbf{P}$, equation (1) can be rewritten as

$$\nabla \cdot \mathbf{E} = -\nabla^2\varphi = \frac{\rho_{app}}{\varepsilon_0}. \tag{5}$$

It is worth noting that the Charge Inversion Algorithm aims at quantifying the free charge density distribution, i.e., ρ_{free} (which is the main cause that lowers the surface flashover voltage and limits the practical applicability of HVDC-GIS systems), whereas ρ_{app} is introduced for computation purposes.

The Poisson equation (5) can then be reformulated in terms of Green function $G(\mathbf{r}, \mathbf{r}')$, i.e., the solution of:

$$\nabla^2 G(\mathbf{r}, \mathbf{r}') = -\delta(\mathbf{r} - \mathbf{r}'), \tag{6}$$

where $\mathbf{r}$ is the field point, $\mathbf{r}'$ is the source point, and

$$G(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi||\mathbf{r} - \mathbf{r}'||}. \tag{7}$$

By using (6) and (7) into (5), the integral solution of the Poisson equation is obtained:

$$\varphi(\mathbf{r}) = \int_\Omega \frac{\rho_{app}(\mathbf{r}')}{\varepsilon_0} G(\mathbf{r}, \mathbf{r}') d\Omega = \int_\Omega \frac{\rho_{app}(\mathbf{r}')}{4\pi\varepsilon_0||\mathbf{r} - \mathbf{r}'||} d\Omega, \tag{8}$$

where Ω is the integration domain. This integral equation is the starting point of the Charge Inversion Algorithm [13]. In practical situations, the charge density only stays on dielectric and metallic surfaces [12]; therefore, equation (8) is rewritten as:

$$\varphi(\mathbf{r}) = \int_\Gamma \frac{\sigma_{app}(\mathbf{r}')}{4\pi\varepsilon_0||\mathbf{r} - \mathbf{r}'||} d\Gamma, \tag{9}$$

where $\sigma_{app}(\mathbf{r})$ is the apparent surface charge density and Γ is the boundary of Ω [30]. The numerical solution of (9) requires the discretization of the computational domain Γ into, e.g., triangular elements and the expansion of the unknown $\sigma(\mathbf{r})$ in terms of Degrees of Freedom (DoFs) by using base functions w_k:

$$\sigma_{app}(\mathbf{r}) = \sum_{k=1}^{N} q_{app,k} w_k(\mathbf{r}), \tag{10}$$

where N is the number of triangular elements and $q_{app,k}$ is the apparent charge on the k-th one [31]. In this paper, piece-wise uniform bases function are considered, i.e., $w_k(\mathbf{r}) = 1/|S_k|$ when $\mathbf{r}$ is in the k-th element, and $w_k(\mathbf{r}) = 0$ when $\mathbf{r}$ is elsewhere, where $|S_k|$ is the area of the k-th triangular element [31]. However, higher order shape functions or different mesh elements can be also used. Using expansion (10), and projecting the resulting equations by adopting the same shape functions (i.e., by exploiting the Galerkin scheme), equation (9) becomes:

$$\phi_i = \int_\Gamma w_i(\mathbf{r}) \left(\sum_{j=1}^{N} \int_\Gamma \frac{w_j(\mathbf{r}') q_{app,j}}{4\pi\varepsilon_0||\mathbf{r}_i - \mathbf{r}_j'||} d\Gamma \right) d\Gamma, \tag{11}$$

where ϕ_i is the average electric potential in the i-th triangle. It is worth noting that in other works in the literature a collocation method is used instead of the Galerkin scheme here adopted [11], i.e. no averaging is performed on the electric potentials. In this paper, the Galerkin procedure (i.e., the averaging) is adopted since it allows for alleviating the issues related to the singularity of the Green's function in the evaluation of the integrals. However, all the discussions and considerations of the paper still hold if a collocation method is used for deriving (11), as in [11].

Then, equation (11) can be recasted in matrix form:

$$\mathbf{\Phi} = \mathbf{P}\mathbf{q}_{app}, \tag{12}$$

where $\mathbf{\Phi}$ and $\mathbf{q}_{app}$ are arrays storing the electric potential and apparent charges on each triangle. The coefficients of matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$ are given by:

$$P_{ij} = \frac{1}{|S_j||S_i|} \int_{S_i} \int_{S_j} \frac{1}{4\pi\varepsilon||\mathbf{r}_i - \mathbf{r}_j'||} d\Gamma' d\Gamma. \tag{13}$$

When the field and integration points overlap ($\mathbf{r}' \to \mathbf{r}$), the integrand (13) exhibits a singularity due to the Green's function. To properly evaluate matrix coefficients, singularity extraction techniques reported in the literature should be adopted [32]. On the contrary, the larger the distance between field and integration points, the smoother the integrand function [22], and therefore simple numerical techniques can be used to evaluate far-mutual coefficients without losing accuracy. Once the electric potentials are obtained from measurements, the apparent charge is computed by solving (12).

As previously discussed, the aim of the Charge Inversion Algorithm is to find the free surface charge density distribution, and not the apparent one. The free surface charge $\mathbf{q}$ is derived by using the dielectric interface boundary condition [11, 12, 13, 14]. Numerically, this procedure is performed by multiplying $\mathbf{q}_{app}$ with an integral matrix operator, whose coefficients depends on the relative permittivity discontinuities. Such operator, usually indicated as $\mathbf{F}$, is given in, e.g., [11, 14]. Therefore, $\mathbf{q}$ (i.e., the array storing the charges in each triangle) is obtained by using $\mathbf{F}$, i.e.,

$$\mathbf{q} = \mathbf{F}\mathbf{q}_{app}. \tag{14}$$

Once $\mathbf{q}$ is obtained, the free surface charge density, σ, is given by

$$\sigma(\mathbf{r}) = \sum_{k=1}^{N} q_k w_k(\mathbf{r}). \tag{15}$$

Nevertheless, the bottleneck of the surface Charge Inversion Algorithm is related to the computational effort required to solve equation (12), due to the fact that $\mathbf{P}$ is a full matrix.

3. Data-Sparse Representation

Generally, a large number of mesh elements is required to reach good accuracy with the Charge Inversion Algorithm. However, this leads to a high computational effort since the storage and assembly cost for generating $\mathbf{P}$ grows with N^2, whereas the direct solution of (12) has $\mathcal{O}(N^3)$ complexity. Fortunately, due to the smoothness of the Green's function for distant field and integration points, the fully populated matrix $\mathbf{P}$ contains low-rank blocks. By applying a reordering of the triangles (e.g., following a geometric criterion), off-diagonal matrix blocks representing interaction between far-field points, can be represented with low–rank approximation techniques making $\mathbf{P}$ compressible [33].

There exists a variety of *data-sparse* representations based on $\mathcal{H}$-matrices: Hierarchical Off Diagonal Low Rank (HODLR) [15], Hierarchical Semi Separable (HSS) [34], Hierarchical Adaptive Low Rank (HALR) [35], and $\mathcal{H}^2$ matrices [16], which can be used to represent $\mathbf{P}$ in a data-sparse format, while keeping a reasonably high accuracy. In the next sections, the steps required to generate such sparse representation of $\mathbf{P}$ are discussed.

3.1. Geometric Partitioning and Cluster Tree

This section describes the geometric procedures at the base of the hierarchical subdivision of the triangles which discretize the boundary Γ.

Starting from ordered sets I, J representing row and column indices of matrix $\mathbf{P}$, corresponding to the triangles, a bisection of each set is generated, e.g., I is partitioned into two clusters: $\tau_I = \{1, \ldots, \lfloor N/2 \rfloor\}$ and $\sigma_I = \{\lfloor N/2 \rfloor + 1, \ldots, N\}$. The same operation is performed for J, leading to the clusters τ_J and σ_J. Then, each of the four cluster pairs representing matrix blocks $\mathbf{P}_{(\tau_I,\tau_J)}$, $\mathbf{P}_{(\tau_I,\sigma_J)}$,$\mathbf{P}_{(\sigma_I,\tau_J)}$,$\mathbf{P}_{(\sigma_I,\sigma_J)}$ are tested against an admissibility criterion which identifies if the cluster is a good candidate for applying low-rank approximation or if it must be further partitioned [27]. Such admissibility check is given by

$$\min\{diam(\tau), diam(\sigma)\} \leq \eta \; dist(\tau, \sigma), \tag{16}$$

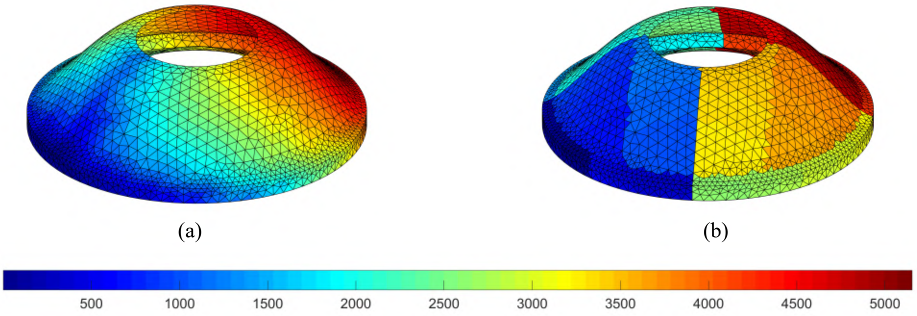

Figure 1. Example of clustering performed with the geometric bisection algorithm, on conical spacer surface. Originial (a) and reordered (b). For the latter case, triangles $\{1, ..., N\}$ were reordered so that near ones have approximately the same colour.

where $\eta > 0$ is the admissibility parameter chosen by the user (e.g., $\eta = 2$), *diam* is the cluster diameter and *dist* is the distance function between clusters (see [22, 33] for more details).

The admissible cluster pairs (i.e., the ones which satisfy (16)), whose corresponding matrix block belongs to the far-field interaction, are stored efficiently with low-rank techniques (as described in the next section); otherwise, the clusters are halved and the procedure is applied recursively until the admissibility check is satisfied or the number of elements (i.e., unknowns) related to the matrix block reaches a specified minimum threshold ($blocksize^2$) [27]. This recursive procedure generates the so called *block cluster tree* [33, 36].

It is worth noting that a proper reordering of the triangles is of crucial importance in order to guarantee that off-diagonal blocks are actually low-rank. In fact, the index set $\{1, ..., N\}$ should be reordered so that close indices in the set correspond to near geometric entities [22]. Only in this condition the far-field interactions (i.e., the off-diagonal blocks of $\mathbf{P}$) can be represented by using low-rank approximation with good accuracy. Such reordering can be performed by clustering the triangles using, e.g., a geometric bisection algorithm. This bisection algorithm constructs a binary *cluster tree* of index set $\{1, ..., N\}$ with a prescribed number of subdivisions (levels). Clearly, increasing the number of levels, the reordering procedure is more accurate. An example of clustering with the geometric bisection is reported in figure 1.

For reordered index sets, the off-diagonal blocks of $\mathbf{P}$ correspond to far-field interactions for which the Green's function is smooth. Therefore, such blocks are actually low-rank and can be efficiently represented by using low-rank compression techniques.

3.2. Low-Rank Approximation

In this section, for the sake of completeness, we briefly describe the low-rank approximation of far-field (admissible) off-diagonal blocks. Consider a matrix block $\mathbf{A} \in \mathbb{R}^{m\times n}$, related to far-field interactions of triangles in τ and σ. Matrix $\mathbf{A}$ has a low-rank representation if exists matrices $\mathbf{U} \in \mathbb{R}^{m\times k}$ and $\mathbf{V} \in \mathbb{R}^{n\times k}$ such that [36]:

$$\mathbf{A} = \mathbf{U}\mathbf{V}^T, \tag{17}$$

with $k(m + n) < mn$, where k is the *effective rank*. Since this is not known a priori, the rank-revealing factorization (17) is computed within a prescribed tolerance ε [37]:

$$\frac{||\mathbf{A} - \mathbf{U}_{k'}\mathbf{V}_{k'}^T||_F}{||\mathbf{A}||_F} < \varepsilon. \tag{18}$$

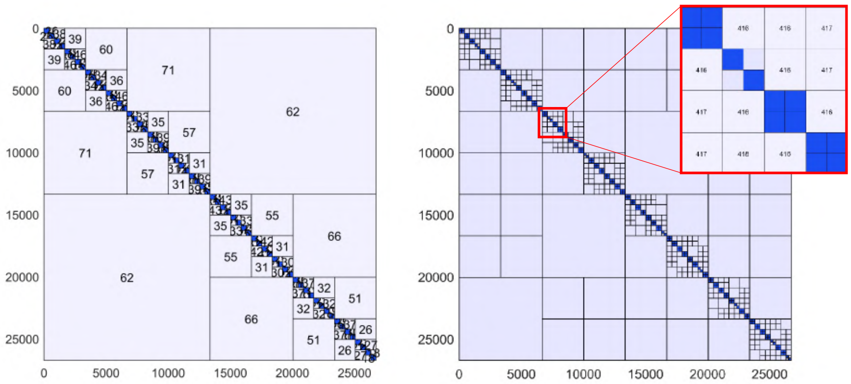

Figure 2. Illustration of block cluster trees for HSS (left) and HALR (right) hierarchical matrices. The number in each block represents the numerical rank of the approximation.

where $k' < k$ is the *numerical rank*. The decomposition (18) can be conveniently obtained by using the Adaptive Cross Approximation (ACA) algorithm [38, 39]. It is a completely *matrix-free* approach which adaptively computes the columns of $\mathbf{U}_{k'}$ and $\mathbf{V}_{k'}$, without previously storing the entire matrix $\mathbf{A}$. The algorithm requires only the capability of computing the generic entry (13) for $i \in \tau, j \in \sigma$. Since only few entries of the original matrix must be computed, it can be proven that the computational cost required by the ACA approximation have a linear-logarithmic complexity [27].

3.3. HALR, HODLR and HSS matrices

The data-sparse representations mentioned in Section 3 differs from the way the block cluster tree is built. For the HALR representation, the cluster tree can be specified by the user allowing for greater flexibility. The HODLR case correspond to $\mathcal{H}$-matrices with partitioning recursively done in 2 x 2 blocks [40], while the HSS format, using the same partitioning, correspond to a $\mathcal{H}^2$-matrix [34] performing a re-compression of blocks. Examples of block cluster tree for HSS and HALR $\mathcal{H}$-matrices are shown in figure 2.

Once the $\mathcal{H}$-matrix representation of $\mathbf{P}$ is obtained, the $\mathcal{H}$-matrix arithmetic allows for standard algebraic operations such as, matrix summation, matrix-vector product and, more importantly, matrix inversion and LU-factorization. Obviously, by exploiting the $\mathcal{H}$-matrix arithmetic, all the operations are significantly accelerated and require less memory with respect to standard arithmetic applied to the corresponding dense matrices.

4. Numerical Procedure

The code used for the numerical experiments has been written in MATLAB® by using parallel MEX–FORTRAN functions based on OpenMP libraries for the computation of the matrix coefficients. The low-rank approximation of the dense matrices was realized with the MATLAB® library hm-toolbox [40] which allows for using different techniques for the data-sparse representation of $\mathbf{P}$ and the solution of the inverse problem.

In particular, depending on the user requirements, we describe here two alternative procedures where $\mathcal{H}$–matrices are used to reduce the computational cost of the Charge Inversion Algorithm:

(i) If the storage of the full matrix $\mathbf{P}$ is not the main problem, the data-sparse representation can be performed by directly using the entire matrix $\mathbf{P}$. Then, a data-sparse representation

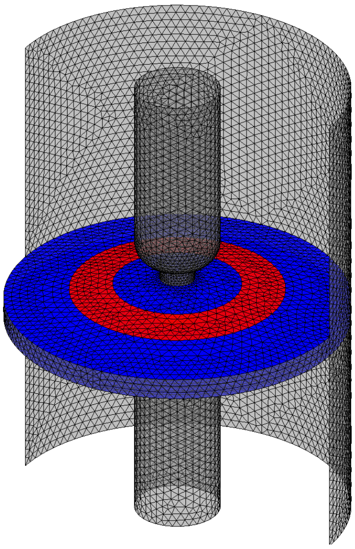

Figure 3. 3D view of conical spacer (blue) with deposited annular charge density (red).

of **P**, i.e., $\mathcal{H}$-**P**, is generated by using, e.g., HODLR, HALR or HSS format. By using the algebra of $\mathcal{H}$-matrices implemented in the hm-toolbox, a data-sparse $\mathcal{H}$-*LU* factorization of $\mathcal{H}$-**P** is efficiently computed and the solution of the linear system (12) is performed with the GMRES iterative solver using the $\mathcal{H}$-*LU* of $\mathcal{H}$-**P** as preconditiner. Since the $\mathcal{H}$-*LU* is a very good preconditioner, GMRES will require only a few iterations to converge, and no factorization of the full matrix **P** is performed.

(ii) If the storage of the full matrix **P** is an issue, the data-spare structure of **P** can be generated by using a matrix-free approach. In this case, only an operator which evaluates the coefficients of **P** for a generic set of indices is required. Thus, the data-sparse representation of **P**, $\mathcal{H}$-**P**, can be generated by using, e.g., HODLR, HALR or HSS format and the corresponding $\mathcal{H}$-*LU* factorization is obtained. Finally, a GMRES solver can be adopted to solve the problem by using $\mathcal{H}$-**P** as the system matrix and the $\mathcal{H}$-*LU* as preconditioner.

As a preprocessing operation, the reordering of triangles was performed with the geometric bisection algorithm.

5. Numerical Results

The accuracy analysis of the charge reconstruction in a conical spacer [41] was taken as test case to show the performances of the data-sparse method. As shown in figure 3, a charge density $\sigma_{imposed} = 1$ C/m^2 was deposited on an annular area of the upper surface of the insulator and the corresponding electric potential $\mathbf{\Phi}$ was calculated with (12). The computational domain was discretized with different numbers of triangular elements: N = 26724 and N = 42852 to check the scalability of the proposed method. For the block cluster tree construction, *blocksize* = 64 was adopted. A summary of the computational performances for the test case is listed in table 1 and table 2, including the following data:

- Memory Ratio: data-sparse compression percentage, defined as:

$$\text{Memory Ratio} = 100 \cdot \frac{MemorySize(\mathcal{H})}{MemorySize(P)}, \tag{19}$$

where *MemorySize(·)* gives the computer memory required by the object;

Table 1. Results for N = 26724 triangular elements following approach (ii) of Section 4.

-	Memory ratio [%]	Relative error [%]	Construction [s]	$\mathcal{H}$-*LU* [s]	Solving [s]
HODLR ($\varepsilon = 10^{-2}$)	10.4	10.6	11	20	0.7
HODLR ($\varepsilon = 10^{-3}$)	14.1	2.4	9	25	0.6
HODLR ($\varepsilon = 10^{-4}$)	18.7	0.3	14	36	0.5
HODLR ($\varepsilon = 10^{-5}$)	23.3	0.1	26	59	0.5
HSS ($\varepsilon = 10^{-3}$)	2.8	5.9	189	—	2.5
HSS ($\varepsilon = 10^{-4}$)	7.6	0.8	403	—	3.8
HSS ($\varepsilon = 10^{-5}$)	12.6	0.1	422	—	6.5
HALR ($\varepsilon = 10^{-3}$)	23.0	1.2	21	34	1.0
HALR ($\varepsilon = 10^{-4}$)	25.6	0.3	20	62	1.9
HALR ($\varepsilon = 10^{-5}$)	28.9	0.01	16	51	0.8
Dense	100.0	0.0	39	—	26.5

Table 2. Results for N = 129384 triangular elements following approach (ii) of Section 4. The low-rank accuracy ε was set to 10^{-4}.

-	Memory ratio [%]	Relative error [%]	Construction [s]	$\mathcal{H}$-*LU* [s]	Solving [s]
HODLR	7.8	0.5	210	492	4
HSS	1.4	1.2	9423	—	25
HALR	6.3	0.3	142	281	5
Dense	100.0	0.0	889	—	1941

- Relative Error: relative percentage error as:

$$\text{Relative Error} = 100 \cdot \frac{||\sigma - \sigma_{\textbf{imposed}}||}{||\sigma_{\textbf{imposed}}||}, \tag{20}$$

  with σ the solution of the problem (12) obtained from (15);
- Construction: time spent to assemble data-sparse format;
- LU: time spent to compute the $\mathcal{H}$-*LU* factorization;
- Solving: time spent to solve the linear system with GMRES (tol = 10^{-6}).

It it worth nothing that the memory required to store the full **P** matrix for N=26724 and N=129384 is 5.7 GB and 133.9 GB, respectively. However, using e.g., the HSS ($\varepsilon = 10^{-4}$), the memory requirement drastically reduces to 0.43 GB and 1.86 GB.

For the HODLR cases of table 1, figure 4 shows the error map determined as the absolute difference between the imposed and the reconstructed charge density, on the upper part of the conical insulator, on each triangular element.

Also the approach (i) of Section 4 is tested against N = 129384 triangles and the results are reported in table 3. We remind that in this case the full (dense) matrix **P** was used for the matrix-vector products required by GMRES iterations and the $\mathcal{H}$-*LU* factors of $\mathcal{H}$-*P* (computed with $\varepsilon = 10^{-2}$) were used for the preconditioning.

Table 3. Results for N = 129384 triangular elements following approach (i) of Section 4. The low-rank accuracy ε was set to 10^{-2} to lower the computational effort for $\mathcal{H}$-*LU* computation.

-	Memory ratio	Relative error	Construction	$\mathcal{H}$-*LU* [s]	Solving
	[%]	[%]	[s]	[s]	[s]
HODLR	4.5	$8{\cdot}10^{-5}$	114	197	41
HSS	0.2	$1.7{\cdot}10^{-4}$	1798	—	109
HALR	4.3	$5{\cdot}10^{-5}$	80	123	43
Dense	100.0	0.0	889	—	1941

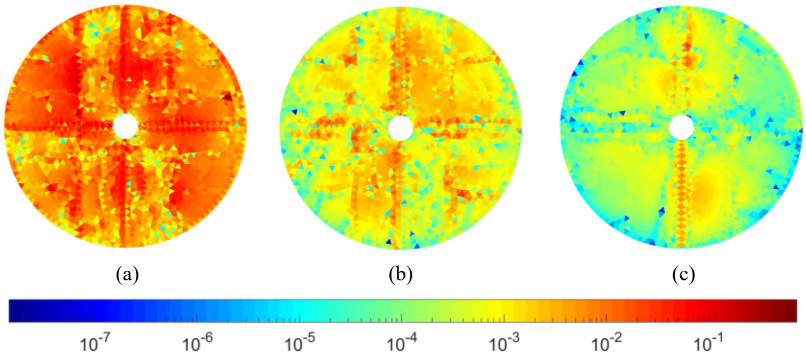

Figure 4. Reconstruction error map using HODLR representation, with low-rank approximation parameter ε=10^{-2} (a), ε=10^{-3} (b), ε=10^{-4} (c), from the left to the right.

The overall results reported in the figures and tables show that $\mathcal{H}$-matrices are a valid tool to significantly reduce the computational burden required for the solution of (12), both in terms of memory requirement and computation time. In particular, the computational performances of HALR are slightly better than HODLR when large problems are considered (see table 2). Indeed, thanks to the higher generality of the cluster tree (see figure 2), HALR matrices allow for low memory ratio and high accuracy. HSS matrices allow for very low memory ratio and good accuracy, however they also require a long computation time for the construction of the HSS matrix representation (see table 2 and table 3). Indeed, they require the generation of all the coefficients of **P** (several times) and the re-compression require a non-negligible computation time. Therefore, HSS should be used when the memory must be kept very low wheres the computation time is not the main issue.

In this paper, we have used the hm-toolbox [40], which provides HODLR, HALR, and HSS matrix implementations that can be easily applied to a large class of problems. However, it is worth noting that better performances (even in terms of orders of magnitude for the computation time) could be achieved by using the highly optimized $\mathcal{H}$LibPro library [42] for HALR matrices (which exploits a lower-level programming language with respect to hm-toolbox), as shown by many works where such library was used in IE methods [31, 22, 17], or STRUMPACK library [43] for HSS matrices.

6. Conclusions

This paper describes the application of an improved Charge Inversion Algorithm for high voltage gas insulated systems. The bottleneck of the method was highlighted and an efficient solution for reducing the storage and inversion requirements of $\mathbf{P}$ matrix, with data-sparse (hierarchical) techniques, has been presented. The effectiveness of the data-sparse representation was verified evaluating the capability of reconstruction of the apparent charge deposited on a ring surface of a conical insulator.

The performances in term of storage requirement, solution relative error and construction times were analyzed in terms of different data-sparse representations and low-rank approximation tolerances. The data-sparse format, due to its logarithmic scaling, becomes more and more efficient for increasing number of discretization elements. This aspect, together with the matrix-free approach, sets up the possibility of using a huge number of elements, providing better numerical accuracy.

7. References

[1] M. P. Bahrman and B. K. Johnson. The ABCs of HVDC transmission technologies. *IEEE Power and Energy Magazine*, 5(2):32–44, March 2007.

[2] M. Boldrin, M. Valente, S. Dal Bello, L. Grando, V. Toigo, P. Zaccaria, H. Decamps, H. Tobari, M. Simon, G. E. Gomez, and A. Garbuglia. 1 MV power supplies integration issues in MITICA experiment, the ITER heating neutral beam injector prototype. *Fusion Engineering and Design*, 164:112170, 2021.

[3] M. Kosse, K. Juhre, M. Kuschel, and D. Li. Overview of development, design, testing and application of compact gas-insulated dc systems up to ±550 kv. *Global Energy Interconnection*, 2(6):567–577, 2019.

[4] A. Winter and J. Kindersberger. Transient field distribution in gas-solid insulation systems under DC voltages. *IEEE Transactions on Dielectrics and Electrical Insulation*, 21(1):116–128, February 2014.

[5] P. Bettini, R. Specogna, and F. Trevisan. Electroquasistatic analysis of the gas insulated line for the iter neutral beam injector. *IEEE Transactions on Magnetics*, 45(3):996–999, 2009.

[6] G.-M. Ma, H.-Y. Zhou, C.-R. Li, J. Jiang, and X.-W. Chen. Designing epoxy insulators in SF6-filled DC-GIL with simulations of ionic conduction and surface charging. *IEEE Transactions on Dielectrics and Electrical Insulation*, 22(6):3312–3320, December 2015.

[7] E. Volpov. Dielectric strength coordination and generalized spacer design rules for HVAC/DC SF6 gas insulated systems. *IEEE Transactions on Dielectrics and Electrical Insulation*, 11(6):949–963, Dec 2004.

[8] C. Li, J. Hu, C. Lin, B. Zhang, G. Zhang, and J. He. Surface charge migration and dc surface flashover of surface-modified epoxy-based insulators. *Journal of Physics D: Applied Physics*, 50(6):065301, jan 2017.

[9] M. Schueller, U. Straumann, and C. M. Franck. Role of ion sources for spacer charging in SF6 gas insulated hvdc systems. *IEEE Transactions on Dielectrics and Electrical Insulation*, 21(1):352–359, February 2014.

[10] H.-Y. Zhou, G.-M. Ma, Y. Wang, C.-R. Li, Y.-P. Tu, S.-P. Ye, B. Zhang, X.-F. Guo, and X.-L. Yan. Surface charge accumulation on 500kV cone-type GIS spacer under residual DC voltage. *IEEE Transactions on Dielectrics and Electrical Insulation*, 25(4):1230–1237, August 2018.

[11] H. Ootera and K. Nakanishi. Analytical method for evaluating surface charge distribution on a dielectric from capacitive probe measurement-application to a cone-type spacer in +or-500 kV DC-GIS. *IEEE Transactions on Power Delivery*, 3(1):165–172, Jan 1988.

[12] Q. Wang, G. Zhang, and X. Wang. Characteristics and mechanisms of surface charge accumulation on a cone-type insulator under dc voltage. *IEEE Transactions on Dielectrics and Electrical Insulation*, 19(1):150–155, February 2012.

[13] C. Lin, C. Li, J. He, J. Hu, and B. Zhang. Surface charge inversion algorithm based on bilateral surface potential measurements of cone-type spacer. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(3):1905–1912, June 2017.

[14] F. Wang, F. Liang, S. Chen, Y. Tan, L. Zhong, and Q. Sun. Surface charge inversion method on cylindrical insulators based on surface potentials measured online. *IEEE Transactions on Dielectrics and Electrical Insulation*, 28(1):192–197, February 2021.

[15] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405–422, 2003. Large scale problems using BEM.

[16] D. Kressner, S. Massei, and L. Robol. Low-rank updates and a divide-and-conquer method for linear matrix equations. *SIAM Journal on Scientific Computing*, 41(2):A848–A876, 2019.

[17] R. Torchio, V. Cirimele, P. Alotto, and F. Freschi. Modelling of road–embedded transmitting coils for wireless power transfer. *Computers & Electrical Engineering*, 88:106850, 2020.

[18] M. Passarotto, R. Specogna, and C. Geuzaine. Fast iterative schemes for the solution of eddy-current problems featuring multiple conductors by integral formulations. *IEEE Transactions on Magnetics*, 56(3):1–4, 2020.

[19] P. Baumgartner, T. Bauernfeind, O. Biro, C. Magele, W. Renhart, and R. Torchio. Synthesis of NFC antenna structure under multi-card condition. *Applied Computational Electromagnetics Society Journal*, 33(10):1161–1163, 2018.

[20] G. Meunier, Q.-A. Phan, O. Chadebec, J.-M. Guichon, B. Bannwarth, and R. Torchio. Unstructured PEEC method with the use of surface impedance boundary condition. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, 39(5):1017–1030, Jan 2020.

[21] R. Torchio. A volume PEEC formulation based on the cell method for electromagnetic problems from low to high frequency. *IEEE Transactions on Antennas and Propagation*, 67(12):7452–7465, 2019.

[22] D. Voltolina, R. Torchio, P. Bettini, R. Specogna, and P. Alotto. Optimized cycle basis in volume integral formulations for large scale eddy-current problems. *Computer Physics Communications*, 265:108004, 2021.

[23] R. Torchio, L. Codecasa, L. Di Rienzo, and F. Moro. Fast uncertainty quantification in low frequency electromagnetic problems by an integral equation method based on hierarchical matrix compression. *IEEE Access*, 7:163919–163932, 2019.

[24] V. Cirimele, R. Torchio, A. Virgillito, F. Freschi, and P. Alotto. Challenges in the electromagnetic modeling of road embedded wireless power transfer. *Energies*, 12(14), 2019.

[25] V. Cirimele, R. Torchio, J. L. Villa, F. Freschi, P. Alotto, L. Codecasa, and L. Di Rienzo. Uncertainty quantification for SAE J2954 compliant static wireless charge components. *IEEE Access*, 8:171489–171501, 2020.

[26] P. Bettini, M. Passarotto, and R. Specogna. Coupling volume and surface integral formulations for eddy-current problems on general meshes. *IEEE Transactions on Magnetics*, 54(3):1–4, 2018.

[27] R. Torchio, P. Bettini, and P. Alotto. PEEC-based analysis of complex fusion magnets during fast voltage transients with H-matrix compression. *IEEE Transactions on Magnetics*, 53(6):1–4, 2017.

[28] D. Romano and G. Antonini. Quasi-static partial element equivalent circuit models of magneto-dielectric materials. *IET Microwaves, Antennas & Propagation*, 11(6):915–922, 2017.

[29] P. Bettini and F. Trevisan. Electrostatic analysis for plane problems with finite formulation. *IEEE Transactions on Magnetics*, 39(3):1127–1130, 2003.

[30] R. Torchio, P. Alotto, P. Bettini, D. Voltolina, and F. Moro. A 3-D PEEC formulation based on the cell method for full-wave analyses with conductive, dielectric, and magnetic media. *IEEE Transactions on Magnetics*, 54(3):1–4, 2018.

[31] R. Torchio, F. Moro, G. Meunier, J.-M. Guichon, and O. Chadebec. An extension of unstructured-PEEC method to magnetic media. *IEEE Transactions on Magnetics*, 55(6):1–4, 2019.

[32] S. Järvenpää, M. Taskinen, and P. Ylä-Oijala. Singularity extraction technique for integral equation methods with higher order basis functions on plane triangles and tetrahedra. *International Journal for Numerical Methods in Engineering*, 58(8):1149–1165, 2003.

[33] D. Voltolina, P. Bettini, P. Alotto, F. Moro, and R. Torchio. High-performance PEEC analysis of electromagnetic scatterers. *IEEE Transactions on Magnetics*, 55(6):1–4, 2019.

[34] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numerical Linear Algebra with Applications*, 17(6):953–976, 2010.

[35] S. Massei, L. Robol, and D. Kressner. Hierarchical adaptive low-rank format with applications to discretized PDEs, 2021.

[36] M. Bebendorf. *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*. Lecture Notes in Computational Science and Engineering. Springer-Verlag Berlin Heidelberg, 2008.

[37] P. Alotto, P. Bettini, and R. Specogna. Sparsification of BEM matrices for large-scale eddy current problems. *IEEE Transactions on Magnetics*, 52(3):1–4, 2016.

[38] Z. Kezhong, M. N. Vouvakis, and L. Jin-Fa. The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems. *IEEE Transactions on Electromagnetic Compatibility*, 47(4):763–773, 2005.

[39] G. Antonini and D. Romano. Efficient frequency-domain analysis of peec circuits through multiscale compressed decomposition. *IEEE Transactions on Electromagnetic Compatibility*, 56(2):454–465, 2014.

[40] S. Massei, L. Robol, and D. Kressner. hm-toolbox: Matlab software for hodlr and hss matrices. *SIAM Journal on Scientific Computing*, 42(2):C43–C68, 2020.

[41] B. Zhang, W. Gao, Z. Qi, Q. Wang, and G. Zhang. Inversion algorithm to calculate charge density on solid dielectric surface based on surface potential measurement. *IEEE Transactions on Instrumentation and Measurement*, 66(12):3316–3326, 2017.

[42] R. Kriemann. HLibPro library, https://www.hlibpro.com/index.html.

[43] F. H. Rouet et al. STRUMPACK library, http://portal.nersc.gov/project/sparse/strumpack/.