

Krylov Subspace Methods for Big Data Analysis of Large Computational Electromagnetics Applications

Bruno CARPENTIERI ^{a,1}

^aFaculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bolzano, Italy

Abstract. In this paper we present some computational techniques based on the class of preconditioned Krylov subspace methods that enable us to carry out large-scale, big data simulations of Computational Electromagnetics applications modeled using integral equations. This analysis requires the solution of large linear systems that cannot be afforded by conventional direct methods (based on variants of the Gaussian elimination algorithm) due to their high memory costs. We show that, thanks to the development of efficient Krylov methods and suitable preconditioning techniques, nowadays the solution of realistic electromagnetic problems that involve tens of million (and sometimes even more) unknowns, has become feasible. However, the choice of the best class of methods for the selected computer hardware and the given geometry remains an open problem that requires further analysis.

Keywords. Computational Electromagnetics, Boundary Element Method, Surface Integral Equations, Krylov Subspace methods, Preconditioning, Big Data.

1. Introduction

Our society is characterized by an unprecedented ability to generate large volume of data that can be analysed by advanced computer algorithms to produce new knowledge, and use such information to simulate and reproduce the complex behaviour of real-world systems. The “big data” paradigm in science is driving the development of novel scientific methodologies that are opening up new frontiers for advanced discoveries in many fields of science; see e.g. [1,2,3,4,5,6] for some examples of big data simulations in computational science. In this paper, we discuss a case study in Computational Electromagnetics.

Many wave propagation phenomena in science are formulated mathematically in terms of integral equation models that are defined on the boundary of the pertinent computational domain. One of the most popular integral equation models for electromagnetic (EM) scattering applications is known as Electric Field Integral Equation (EFIE). It solves the integral equation below for the surface current \vec{j}

¹Corresponding Author: Bruno Carpentieri; E-mail: bcarpentieri@gmail.com.

$$\int_{\Gamma} \int_{\Gamma} G(|y-x|) \left(\vec{j}(x) \cdot \vec{j}^{\dagger}(y) - \frac{1}{k^2} \operatorname{div}_{\Gamma} \vec{j}(x) \cdot \operatorname{div}_{\Gamma} \vec{j}^{\dagger}(y) \right) dx dy = \frac{i}{k Z_0} \int_{\Gamma} \vec{E}_{inc}(x) \cdot \vec{j}^{\dagger}(x) dx,$$

where \vec{j}^{\dagger} 's denote suitable tangential test functions. By symbol $\operatorname{div}_{\Gamma} \vec{j}(x)$ we indicate the divergence operator, and $G(|y-x|) = \frac{e^{ik|y-x|}}{4\pi|y-x|}$ is the Green's function for scattering problems. The other symbols are defined as follows: Γ is the boundary of the domain of interest, by $Z_0 = \sqrt{\mu_0/\epsilon_0}$ the impedance of free space, and finally by k , ϵ and μ the wave number, the electric permittivity and the magnetic permeability of the medium, respectively. This model is applicable to generic objects, including those that have cavities and/or disconnected parts.

The numerical discretization of EFIE by the Method of Moments (MoM) [7] using a mesh with n edges yields an n -dimensional linear system

$$Ax = b. \quad (1)$$

In Eq. (1), the unknowns in vector x represent the vectorial flux of the surface electric current across the edges of the underlying mesh, A is a dense, complex, symmetric non-Hermitian matrix containing the contributions to the EFIE singular integrals, while the right-hand side vector b depends on the characteristics of the illuminating radiation such as the incidence angle. Scattering analysis may demand very large computer resources and highly efficient numerical algorithms. For example, accurate modeling of a perfectly conducting sphere of diameter of 1,800 wavelengths yields systems with more than 3 billion equations, whose storage requires 144,000 petabyte of data [8]. Systems of this size are not affordable using variants of the conventional Gaussian elimination algorithm. They can be solved only using matrix-free computational techniques such as the class of iterative Krylov subspace methods, that can overcome the memory bottlenecks of Gaussian elimination since they are based on matrix-vector and vector-vector operations.

The Generalized Minimum Residual (GMRES) method introduced by Saad and Schultz in [9] is a very popular Krylov subspace algorithms. After k iterations it computes the approximate solution of Eq. (1) that minimizes the 2-norm of the residual over the Krylov space $K_k(A, r_0) = \operatorname{span}\{r_0, Ar_0, \dots, A^k r_0\}$ at the cost of $\mathcal{O}(nk)$ arithmetic operations and storage units. On the other hand, *nonoptimal* Krylov methods are developed upon three-term vector recurrences and have $\mathcal{O}(n)$ complexity in both time and space. The principal developments of nonoptimal methods include the Conjugate Gradients Squared (CGS) method by Sonneveld, the Quasi-Minimal Residual (QMR) and the Transpose-Free Quasi-Minimal Residual methods by Freund and Nachtigal, the Biconjugate Gradient STABILized (BiCGSTAB) method by van der Vorst, and others. See [10] for an overview of Krylov subspace methods. Recently, a new class of Krylov algorithms built upon the Lanczos A -orthonormalization procedure has shown competitive convergence rates for solving surface integral equations [11].

In many EM applications, including scattering analysis, microwave and millimeter-wave circuits design, antenna array simulations and others, the pertinent linear systems to solve have all the same coefficient matrix A and a set of different right-hand side vectors. In this circumstance, Eq. (1) writes in the form

$$AX = B, \quad (2)$$

where $B = [b_1, b_2, \dots, b_p] \in \mathbb{C}^{n \times p}$ is the matrix of the, say p , right-hand side vectors b_i , $i = 1, 2, \dots, p$ and $X \in \mathbb{C}^{n \times p}$ is the solution matrix to be computed. Block variants of Krylov subspace methods are significantly more robust than standard Krylov algorithms for solving Eq. (2) as they typically use much larger search spaces and a block implementation of the matrix-vector product operations that can better exploit the sophisticated memory hierarchy of modern computers [10].

2. Preconditioning boundary integral equations

The convergence of Krylov subspace methods is often slow and needs to be accelerated by a technique called *preconditioning*, which transforms the initial system $Ax = b$ into an equivalent system that has more favourable eigenvalues distribution, *i.e.* the vast majority of its eigenvalues are grouped close to point one of the spectrum. The new transformed system writes as $M^{-1}Ax = M^{-1}b$ if the *preconditioner matrix* M is applied from the left, or $AM^{-1}y = b$ (here $x = M^{-1}y$) if it is applied from the right. Preconditioning is necessary on EFIE as the number of Krylov iterations tends to increase as $\mathcal{O}(n^{0.5})$ when the number of unknowns, n , is related to the wavenumber. An effective preconditioner for solving surface integral equations should be cheap to compute and easy to combine with the data structure of fast integral equations solvers, *e.g.*, the Multilevel Fast Multipole Algorithm (MLFMA) [12], \mathcal{H} -matrices [13], wavelet techniques [14], panel clustering [15] and similar approaches, so that it can maintain overall $\mathcal{O}(n \log n)$ complexity. Another important requirement is that it should scale satisfactorily with the frequency of the problem and the number of processors, yielding robust convergence across a wide range of geometries and physical parameters. These requirements often contradict with each other. However, when an effective preconditioner is available, the selection of the Krylov algorithm to use is much less critical.

The need to solve large dense linear systems in big data EM simulations has led to the production of many efficient methods rather than to the specialization in one specific technology. For memory concerns, many efficient preconditioners for surface integral equations in EM are constructed from a sparse matrix S that approximates A and is much easier to invert than A . Initially, the boundary element matrix A is decomposed in the form

$$A = A_{near} + A_{far},$$

where A_{near} is the block diagonal and near-diagonal part of A coming from the interactions of nearby basis functions in the mesh, while A_{far} is the far-field part of A associated with interactions of distant basis functions. Then, it is natural to define $S = A_{diag} + A_{near}$. In the next sections, we identify some of the most important classes of preconditioning methods for Krylov subspace solvers constructed from such approximation S .

2.1. Incomplete LU (ILU) factorization methods

Incomplete LU factorization preconditioners decompose approximately matrix S as $M = \tilde{L}\tilde{U} \approx S$, where the factors \tilde{L} , \tilde{U} are obtained by applying an incomplete Gaussian elimination procedure to S . Due to the indefiniteness of A , on EFIE it is likely to

encounter small pivots during the factorization, and to compute ill-conditioned triangular solves [16]. The problem of ill-conditioning can be remedied by reordering the entries of S before the factorization, by shifting the diagonal entries of S , or by using numerical pivoting. A recently developed family of multilevel incomplete LU factorization methods has yielded robust performance on this problems class (see e.g. some results in [17]). Incomplete LU factorization algorithms are not inherently parallel. However, good scalability can be obtained using domain decomposition techniques at the cost of moderate computational overhead.

2.2. Sparse approximate inverse methods

Considerable attention in the last years in EM have received computational methods that explicitly approximate and store the inverse of boundary integral equation matrices [18,19]. The approximate inverse matrix $M \approx S^{-1}$ is used as a preconditioner for Krylov methods. This approach is especially interesting for parallelism because applying M at every iteration simply requires one (or sometimes two) sparse M-V products that are easier to implement efficiently than conventional triangular systems solves on modern distributed memory multiprocessor computers and graphics processing units (GPUs). Admittedly, an efficient parallel implementation of these methods based on the Message Passing Interface and Open Multi-Processing paradigms with optimized parameter setting, runtime environment, load balance and minimised data movement on the given problem and dataset is a challenge in its own right [18,20]. Due to the highly localized coupling of the edges in the underlying mesh, boundary element matrices exhibit a good deal of regularity. As a consequence of this property, A can be effectively approximated by a very sparse matrix. Figure 1 shows that 1) the distribution of the large matrix entries in A and A^{-1} can be very similar because of the exponential decay of the Green's function, 2) a sparse matrix can approximate A^{-1} very effectively, and 3) the pattern of the sparsified matrix A can be a good choice for the nonzero structure of the approximate inverse.

The sparse approximate inverse known as SPAI computes the numerical values of the entries of the preconditioner M by minimizing the Frobenius-norm of the matrix $\|I - SM\|_F$ (in the case of right preconditioning), or $\|I - MS\|_F$ (in the case of left preconditioning). The computation reduces to solving n separate linear least-squares problems, one per column or row of M depending whether the preconditioner is computed from the right or from the left, according to the following equation

$$\|I - SM\|_F^2 = \sum_{j=1}^n \|e_j - Sm_{\bullet j}\|_2^2, \quad (3)$$

where we denote by e_j the j th canonical unit vector and $m_{\bullet j}$ is the j th column of M . For right preconditioning, an analogous relation to Eq. (3), that is

$$\|I - MS\|_F^2 = \|I - S^T M^T\|_F^2 = \sum_{j=1}^n \|e_j - S^T m_{j\bullet}\|_2^2$$

holds, where $m_{j\bullet}$ is the j th row of M . Successful experiments with approximate inverse methods are reported for the solution of both surface and surface-volume integral equations [18,19,21].

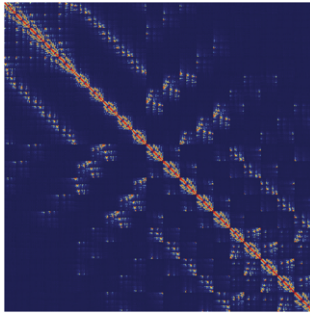
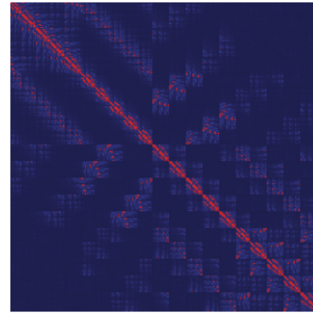
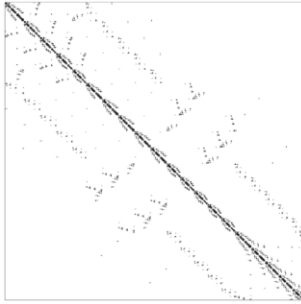
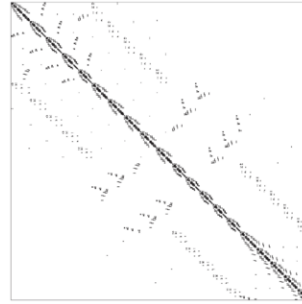
(a) Distribution of the entries values in A .(b) Distribution of the entries values in A^{-1} .(c) Pattern of sparsified A .(d) Pattern of sparsified A^{-1} .

Figure 1. Above: typical pattern of the large entries of the coefficient matrix A (on the left) and of its inverse A^{-1} (on the right) for boundary element matrices. We depict large to small entries using different colors, from red to green, yellow and blue. The model problem is a sphere. Below: nonzero structure of A (left) and of A^{-1} (right) after thresholding all the entries having relative magnitude less than 5.0×10^{-2} .

2.3. Multilevel methods

Due to the sparsity of the local approximation S used to compute the preconditioner M , some strategies need to be developed to make the preconditioner more robust on large problems. In the inner-outer two-levels iterative method proposed in [18], for the preconditioning operation we carry out some iterations of an inner Krylov method, to balance the locality of the preconditioner with the use of the MLFMA matrix arising from the discretization. The effectiveness of this approach depends on three main ingredients: 1) the outer solver must accommodate variable preconditioners (e.g., FGMRES [22] and GMRES* [23, p. 91] are two possibilities), 2) the inner solver needs to be preconditioned to ensure a significant reduction of the inner residuals in a few iterations, 3) the matrix-vector operations in the inner solver can be less accurate as they are used for the preconditioning operation. Another attempt to improve conventional preconditioners for boundary integral equations “removes” the negative effects that very small eigenvalues of the preconditioned matrix can have on the convergence [24].

By using a three-levels inner-outer schemes, we solved a rectangular cavity problem discretized with 12,697,120 unknowns in 58 minutes. The construction of the preconditioner took 30 minutes on 36 processors, and demanded 96GB memory in total. On a computer with 120 cores of a computer cluster equipped with 6-core processors,

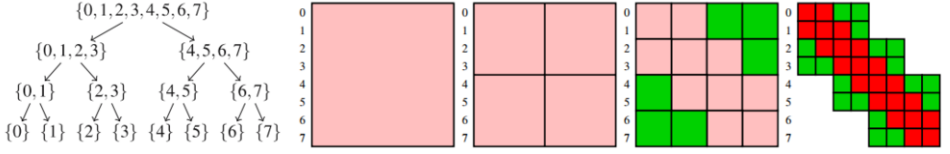


Figure 2. Left. A cluster tree in the \mathcal{H} -matrix approach. Right. Four levels block cluster tree: nodes to be refined are coloured light red, admissible leaves are green, inadmissible ones are red.

2.66GHz clock rate, 10 nodes with 48GB of memory each, a 40Gb/s Infiniband network, we solved a scattering problem from a Cobra geometry modeled with 21,682,980 unknowns in 38 minutes of CPU elapsed time, and a tank with 10,768,581 unknowns in only 12 minutes.

2.4. \mathcal{H} -matrix-based solvers

A more recent approach to build low-complexity data-sparse matrix solvers for EFIE linear systems is based upon the hierarchical \mathcal{H} -matrix representation of A , which replaces dense blocks of A by low-rank approximants computed without any knowledge of the underlying kernel. In the original \mathcal{H} -matrix formulation [13], the blocks that are admitted a low-rank representation satisfy the *admissibility criterion*

$$\min(\text{diam}(B_s), \text{diam}(B_t)) \leq \eta \cdot \text{dist}(B_s, B_t) \quad (4)$$

for some $0 < \eta$. In (4), B_s and B_t are rectangular boxes surrounding two clusters s and t of nodes in the mesh, whereas the distances (dist) between s and t and the diameters (diam) of these clusters are computed in terms of the Euclidean norm from the center of gravity of the box. Upon recursive partitioning of the bounding box into smaller boxes until the admissibility condition (4) is not satisfied, a cluster tree data structure is obtained, similar to the left Figure 2. Then, a hierarchical \mathcal{H} -matrix block partitioning of A is produced by associating the matrix block $(A_{ij})_{i \in s, j \in t}$ to the cartesian product $s \times t$, as in the right Figure 2. It has been shown that by means of fast low-rank (k) compression of the admissible blocks, an almost optimal $\mathcal{O}(nk \log n)$ memory complexity and $\mathcal{O}(nk^2 \log^2 n)$ arithmetic costs can be achieved for the matrix vector multiply, factorization and inverse operations using \mathcal{H} -matrices [25].

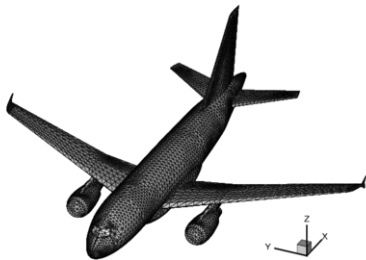
The so-called hierarchical \mathcal{H}^2 -matrix representation can significantly improve the efficiency of \mathcal{H} -matrices by computing nested low rank factorizations of entire collections of blocks instead of decomposing each admissible block separately [26]. An admissible block A_{st} associated to the cluster of nodes s and t satisfies the admissibility condition

$$\max(\text{diam}(B_s), \text{diam}(B_t)) \leq \eta \cdot \text{dist}(B_s, B_t)$$

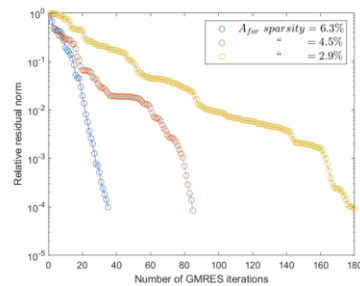
which replaces condition (4) in the theory of \mathcal{H}^2 -matrices, and is represented mathematically by the rank- k factorization

$$\tilde{A}_{st} = V_s S_k V_t^H,$$

where V_s of size $\#s \times k$ and V_t of size $k \times \#t$ (here symbol $\#$ denotes the set cardinality) are called *cluster bases* of clusters s and t , while S_k of size $k \times k$ is called *coupling matrix*. The \mathcal{H}^2 -matrices variant of \mathcal{H} -matrices achieves $\mathcal{O}(nk)$ storage complexity for a $n \times n$ boundary element matrix, instead of the $\mathcal{O}(nk \log n)$ memory complexity of standard \mathcal{H} -matrices, at the cost of only slightly larger error [26]. In Figure 3 we show some convergence results obtained by the author for solving a radar cross-section calculation problem on an Airbus aircraft prototype mesh discretized with $n = 23676$ nodes using an \mathcal{H}^2 -matrices-based solver. We can see that fast convergence can be achieved on this difficult problem using low to moderate sparsity levels for the matrix A_{near} used to construct the preconditioner.



(a) Aircraft prototype mesh (23676 dofs).
Courtesy of EADS-CCR Toulouse.



(b) Convergence histories on the Airbus aircraft prototype mesh.

Figure 3. GMRES convergence histories with \mathcal{H} -matrix-based solvers on the Airbus aircraft problem.

3. Conclusions

In this paper we have discussed large-scale simulations of Computational Electromagnetics applications modeled using integral equations. We have shown that their rigorous numerical solution requires to process large volumes of data, and thus it is highly demanding of innovative algorithms. An essential ingredient of the big data analysis is the numerical solution of high-dimensional linear systems that cannot be solved by the standard Gaussian elimination method. We have presented low-complexity iterative solution techniques based on Krylov subspace methods, incomplete factorizations, sparse approximate inverses, multilevel schemes and \mathcal{H} -matrix-based solvers for this class of problems. An iterative method can solve an $n \times n$ dense linear system arising from the boundary element discretization of integral equations in $\mathcal{O}(n \log n)$ arithmetic operations, that is a dramatic improvement compared to the $\mathcal{O}(n^3)$ work required by a direct method. Effective preconditioners are mandatory to use to decrease the total number of iterations from $\mathcal{O}(n)$ to $\mathcal{O}(1)$. By using Krylov subspace methods and robust preconditioners, big data analysis of extremely large Computational Electromagnetics applications with tens of million unknowns, and even more, is becoming feasible. Some of these techniques can be used in other electromagnetic simulations, e.g. in fusion energy research [27]. However, the choice of the best class

of methods for the selected computer architecture and given geometry remains an open problem that requires further analysis.

Acknowledgements

The author is a member of the *Gruppo Nazionale per il Calcolo Scientifico* (GNCS) of the Istituto Nazionale di Alta Matematica (INdAM) and this work was partially supported by INdAM-GNCS under Progetti di Ricerca 2020. Support from the Research Südtirol/Alto Adige 2019 grant (Provincia autonoma di Bolzano/Alto Adige – Ripartizione Innovazione, Ricerca, Università e Musei, contratto no. 19/34) is gratefully acknowledged.

References

- [1] J. N. Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.
- [2] S. Velamparambil, W. C. Chew, and J. Song. 10 million unknowns: is it that big? [computational electromagnetics]. *IEEE Antennas and Propagation Magazine*, 45(2):43–58, 2003.
- [3] H. Hangan, M. Refan, C. Jubayer, D. Parvu, and R. Kilpatrick. Big data from big experiments. the windeee dome. In *Whither Turbulence and Big Data in the 21st Century?*, pages 215–230. Springer, 2017.
- [4] J. H. Faghmous and V. Kumar. A big data guide to understanding climate change: The case for theory-guided data science. *Big data*, 2(3):155–163, 2014.
- [5] Md Altaf-Ul-Amin, F. M. Afendi, S. K. Kiboi, and S. Kanaya. Systems biology in the context of big data and networks. *BioMed research international*, 2014, 2014.
- [6] S. Pal, S. Mondal, G. Das, S. Khatua, and Z. Ghosh. Big data in biology: The hope and present-day challenges in it. *Gene Reports*, 21:100869, 2020.
- [7] W.C. Gibson. *The method of moments in electromagnetics*. Boca Raton, FL: Chapman & Hall/CRC. xv, 272 p. \$ 119.95 , 2008.
- [8] B. Michiels, J. Fostier, I. Bogaert, and D. De Zutter. Full-wave simulations of electromagnetic scattering problems with billions of unknowns. 63(2), 2015.
- [9] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Scientific and Statistical Computing*, 7:856–869, 1986.
- [10] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, 2nd edition, 2003.
- [11] Y.-F. Jing, B. Carpentieri, and T.-Z. Huang. Experiments with Lanczos biconjugate A-orthonormalization methods for MoM discretizations of Maxwell’s equations. *Progress In Electromagnetics Research*, 99, pages 427–451, 2009.
- [12] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.
- [13] W. Hackbush. A sparse matrix arithmetic based on \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
- [14] B.K. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM J. Scientific and Statistical Computing*, 14:159–184, 1993.
- [15] W. Hackbush and Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54(4):463–491, 1989.
- [16] B. Carpentieri, I.S. Duff, L. Giraud, and M. Magolu monga Made. Sparse symmetric preconditioners for dense linear systems in electromagnetism. *Numerical Linear Algebra with Applications*, 11(8-9):753–771, 2004.
- [17] B. Carpentieri and M. Bollhöfer. Symmetric inverse-based multilevel ILU preconditioning for solving dense complex non-hermitian systems in electromagnetics. *Progress In Electromagnetics Research*, 128:55–74, 2012.
- [18] B. Carpentieri, I.S. Duff, L. Giraud, and G. Sylvand. Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations. *SIAM J. Scientific Computing*, 27(3):774–792, 2005.

- [19] T. Malas and L. Gürel. Accelerating the multilevel fast multipole algorithm with the sparse-approximate-inverse (SAI) preconditioning. *SIAM Journal on Scientific Computing*, 31(3):1968–1984, 2009.
- [20] C. Delgado, E. García, Á. Somolinos, and M.F. Cátedra. Hybrid parallelisation scheme for the application of distributed near-field sparse approximate inverse preconditioners on high-performance computing clusters. *IET Microwaves, Antennas & Propagation*, 14(4):320–328, 2020.
- [21] J. Lee, C.-C. Lu, and J. Zhang. Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 52(9):2277–2287, 2004.
- [22] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Scientific and Statistical Computing*, 14:461–469, 1993.
- [23] H.A. Van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge, UK, 2003.
- [24] B. Carpentieri. A matrix-free two-grid preconditioner for boundary integral equations in electromagnetism. *Computing*, 77(3):275–296, 2006.
- [25] M. Bebendorf. *Hierarchical Matrices*. Springer Berlin Heidelberg, 2008.
- [26] S. Börm. *Efficient Numerical Methods for Non-local Operators*. European Mathematical Society Publishing House, December 2010.
- [27] M. Bonotto, P. Bettini, and A. Cenedese. Model-order reduction of large-scale state-space models in fusion machines via Krylov methods. *IEEE Transactions on Magnetics*, 53(6):1–4, 2017.