This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



# Efficient Scalable Parallel Higher Order Direct MoM-SIE Method with Hierarchically Semiseparable Structures for 3D Scattering

Ana B. Manić, *Student Member, IEEE*, Aaron P. Smull, *Student Member*, *IEEE*, François-Henry Rouet, Xiaoye Sherry Li, and Branislav M. Notaroš, *Fellow, IEEE*

***Abstract*—A novel fast scalable parallel algorithm is proposed for the solution of large 3D scattering problems based on 1) the double (geometrical and current-approximation) higher order (DHO) method of moments (MoM) in the surface integral equation (SIE) formulation, and 2) a direct solver for dense linear systems utilizing hierarchically semiseparable (HSS) structures. Namely, an HSS matrix representation is used for compression, factorization, and solution of the system matrix. In addition, a rank-revealing QR decomposition for memory compression is used, with a stopping criterion in terms of the relative rank tolerance value. A method for geometrical preprocessing of the scatterers based on the cobblestone distance sorting technique is employed in order to enhance the HSS algorithm accuracy and parallelization. Numerical examples show how the accuracy of the DHO HSS-MoM-SIE method is easily controllable by using the relative tolerance for the matrix compression. Moreover, the examples demonstrate low memory consumption, as well as much faster simulation time, when compared to the direct LU decomposition. The method enables dramatically faster monostatic scattering computations than iterative solvers and reduced number of unknowns when compared to low-order discretizations. Finally, great scalability of the algorithm is demonstrated on more than thousand processes.**

***Index Terms*—Numerical algorithms, fast solvers, method of moments, surface integral equation, direct solvers, hierarchically semiseparable structures, multilevel matrix compression, low-rank matrix approximation, parallelization, scalability, higher order modeling, curved parametric elements, polynomial basis functions, scattering.**

## I. Introduction

Recent trends in computational electromagnetics (CEM), in applications that involve calculating, storing, and solving large and dense matrices, include applying fast, parallel (direct or iterative) solvers for the system of equations in conjunction with compressed storage of large matrices and their parts. Two general approaches emerge among fast methods attempting to reduce numerical and storage complexity: (i) the fast multipole method (FMM) [1] and (ii) $\mathcal{H}$-matrices [2]–[4]. The idea behind both of them is to approximate the integral kernel by a degenerate kernel using so-called functional skeletons. In the case of multipole methods, these functions have to be known explicitly for each kernel, which means that the method and its behavior depend heavily on the physics behind the exact problem to be solved. On the other hand, in the case of algebraic methods, such as $\mathcal{H}$-matrices, matrix blocks are approximated by low-rank matrices.

$\mathcal{H}$-matrix algorithms were first introduced by Hackbusch [2]–[4], with their $\mathcal{H}^2$-matrix version being introduced in [5], and have been used efficiently with fast LU based direct solvers or as preconditioners to fast iterative solvers. The $\mathcal{H}$-matrix methods are kernel independent so they are suitable for application to any type of integral equation (IE) based formulation. In the CEM community, several applications of $\mathcal{H}$-type direct solvers to tackle surface integral equation (SIE) problems are presented in [6]–[10].

Matrix compression solvers rely heavily on a type of the low-rank matrix approximation method such as singular value decomposition (SVD) [8], [11], rank-revealing QR (RRQR) decomposition [12], [13] or adaptive cross approximation (ACA), where ACA can be considered as rank-revealing LU (RRLU). ACA is well known and established method for fast matrix computation in CEM, introduced first to solve low-frequency (quasistatic) IE problems [14], and then combined with different matrix compression methods to solve systems of equations arising in high-frequency SIE methods [15]–[18].

In addition, semiseparable matrices, the ones that can easily be compressed and accurately approximated by their low-rank counterpart, and their application to Green's function integral kernels are discussed in [19]. In order to combine the beneficial features of semiseparable matrices and $\mathcal{H}$-matrix representation, hierarchically semiseparable (HSS) matrices were most recently introduced in [20], which exhibit great parallel performance for use in direct solvers when compared to their standard $\mathcal{H}$-matrix counterparts. The solution of problems in two-dimensional (2D) SIE method of moments

Manuscript received August 6, 2015; Revised April 25, 2016; Revised November 3, 2016. This work was supported by the National Science Foundation under grants ECCS-1002385, ECCS-1307863, and AGS-1344862, and by Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research.

A. B. Manić, A. P. Smull, and B. M. Notaroš are with Colorado State University, Department of Electrical and Computer Engineering, Fort Collins, CO 80523-1373 USA (e-mail: anamanic@engr.colostate.edu, asmull@rams.colostate.edu, notaros@colostate.edu; phone: 970-491-3537).

F.-H. Rouet and X. S. Li are with Lawrence Berkeley National Laboratory, Computational Research Division, Berkeley, CA 94720-8150, USA (e-mail: fhrouet@lbl.gov, xsli@lbl.gov).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



(MoM) using a hierarchically semiseparable compression algorithm is discussed in [21], where the authors comment on the possible extension of their work to the 3D case.

On the other hand, it is well known that by using higher order basis functions for current/field modeling in CEM, significant reductions in the number of unknowns, as well as faster system matrix computation/solution, can be achieved [22] when compared to the traditional low order modeling [23]. Tightly coupled with using higher order basis functions is higher order geometrical modeling [22], [24] and together they lay a foundation for double higher order (DHO) modeling. The DHO approach has been effectively used in both direct and iterative MoM-SIE solvers [25]–[29].

Besides developing fast algorithms to solve MoM-SIE equations, the CEM researchers have intensely investigated parallelization of the fast algorithms coupled with direct solvers in order to speed up the simulations of electrically large electromagnetic structures. DHO MoM-SIE system matrix filling followed by a computationally expensive LU decomposition on a full matrix was implemented into parallel out-of-core hybrid GPU/CPU algorithm [28], while the performance of a similar method using higher order basis functions was investigated on more than 4,000 CPU cores on a distributed memory system [29]. Similarly, a parallel $\mathcal{H}$-LU direct solver using hybrid MPI-OpenMP that builds on the ability to combine both shared and distributed memory programing was used to analyze 3D scattering problems with nearly 4 million unknowns [30]. Further, a parallel hierarchical ACA algorithm demonstrating an acceleration factor larger than 200 was presented in [31].

This paper proposes a novel fast scalable higher order parallel algorithm for large and complex scattering, radiation, and propagation problems in CEM based on the DHO MoM-SIE modeling in the frequency domain [22], [24], [32], [33] in conjunction with a direct solver for dense linear systems using HSS matrices [34], namely, the DHO HSS-MoM-SIE method. We are developing asymptotically fast higher order direct algorithms for MoM-SIE solutions which, in a nutshell, are an algebraic generalization to fast multipole methods. In addition to being fast, they offer a promise of being memory- and communication-efficient and amenable to extreme-scale parallel computing. The main advantage of the HSS algorithm is in the linear-complexity ULV-type factorizations (whereas the conventional LU decomposition has cubic complexity). The HSS algorithm is shown to have excellent parallel scalability, Our work uses the recently developed new, state-of-the-art, algorithms for solving dense and sparse linear systems of equations based on the HSS algorithm [34]. The new HSS algorithm has been demonstrated to have a dramatic advantage in terms of time and space complexity (e.g., ~70 times less memory for seismic imaging examples with matrix size 250,000×250,000) over the LU factorization algorithm, and to be extremely scalable. In addition, this paper employs an RRQR decomposition for the matrix (memory) compression. Its adaptive nature comes from the ability to use a stopping criterion, i.e., a relative rank tolerance value, which allows for the method to store only the low-rank approximation of the original matrix that satisfies a predefined accuracy. In order to take full advantage of the HSS algorithm, a method for geometrical preprocessing of scatterers based on the cobblestone distance sorting technique [15] is utilized. MoM unknowns are divided among mesh groups so that unknowns having spatial locality (belonging to the same mesh group) also exhibit locality in the matrix system of equations. In IE methods, these spatially close unknowns have much more numerically significant interactions, whereas interactions between mesh groups are of smaller numerical rank [15]. Ultimately, this spatial localization of the numerical interactions is what is exploited to have such effective coupling with the HSS algorithm. The basic theory and preliminary results of the DHO HSS-MoM-SIE analysis are presented in a summary form in [35], [36].

HSS construction is implemented in a multilevel fashion as described in [34] and, essentially, its multilevel compression can be considered comparable to the one used in the MLACA algorithm [17]. Furthermore, so-called multilevel "butterfly" algorithms [37]–[38], as well as the fast solver presented in [39], have a similar basis to the multilevel compression coupled with low-rank matrix representation. One of the main advantages the HSS algorithm has over methods like the MLACA algorithm is its strong parallel scalability. Furthermore, the possibility to adjust the numerical rank tolerance of the matrix or numerical rank tolerance of any of its sub-blocks gives more control over the method and its computational accuracy. Multilevel compressions utilize matrix rank reduction to the maximum extent and reduce the overall number of degrees of freedom (DoFs) describing the method, as well as the storage requirements, to their minimum. HSS's purely algebraic nature compresses the matrix of interest independently of the problem's geometry or electrical size. The direct ULV-type factorization also allows for a fast solution in the case of multiple excitation vectors (i.e., multiple right hand sides, a matrix of excitation vectors), whereas iterative solvers often require independent solving for each excitation. When coupled with the DHO approach to significantly reduce the number of unknowns in MoM-SIE equations, the result is a fast parallel solver with excellent algorithmic complexity, strong parallel scalability, and controllable accuracy.

Specifically in terms of the comparison with the method in [6], as a major representative of $\mathcal{H}$-matrix applications in CEM, the efficiency of the $\mathcal{H}^2$-matrix solver [6] stems from the fact that it compresses only the parts of the system matrix that correspond to the interactions of the mesh groups located far from each other. The HSS algorithm inherits this behavior. Rather than calculating admissible blocks, the HSS algorithm simplifies the $\mathcal{H}^2$-matrix implementation by considering a single predefined block per tree node to be admissible. The $\mathcal{H}$-matrix type of solver has not been shown to exhibit scalable behavior when parallelized. Further, it is more suitable for use with iterative solvers, because finding the inverse matrix is computationally more demanding than in the case of the HSS algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



Similarly, when specifically compared with the work in [40], as a major representative of the multiscale compressed block decomposition methods and the application of direct solvers in CEM, the matrix compression done in our HSS based method is obtained by fully utilizing hierarchically semiseparable structures, so the compressed system matrix has rather different form, storage and compression layout. In addition, work presented in [40] utilizes an inverse of a system matrix to find an appropriate solution, while in this present work a *ULV* factorization is obtained. Also, this present method is parallel, while the method in [40] is implemented in a sequential fashion.

We find from numerical experiments that the DHO MoM-SIE HSS method, which inherits the aforementioned benefits of HSS matrix representation, is able to compress the involved system matrix while maintaining accuracy of results and results in smaller computational and memory requirements than either DHO modeling alone, or an HSS algorithm-based MoM in a low-order modeling paradigm.

This paper is organized as follows. Section II gives an overview of the MoM-SIE methodology and the associated discretization using DHO modeling. In addition, geometrical preprocessing used to group surface quadrilaterals into mesh groups that achieves spatial-data locality in the system matrix is described. In Section III, the HSS algorithm, including matrix compression and the corresponding ULV factorization followed by the solution of the compressed matrix is discussed. The parallelization and communication between the processes in matrix filling and the HSS algorithm is outlined in Section IV. Section V provides numerical results and discussion, followed by the conclusions in Section VI.

## II. DHO MoM-SIE Modeling of Metallic Scatterers

One of the most general and best established approaches to solving scattering CEM problems is the one based on the method of moments in the surface integral equation formulation and the frequency domain (FD) [24]. Inherently, the MoM results in dense linear systems, so the compression and solver are applied to allow for fast and memory efficient execution.

### A. Surface Integral Equation Formulation

The MoM-SIE methodology is applicable to the analysis of metallic and dielectric structures, where both electric and magnetic surface currents are introduced over boundary surfaces between homogeneous parts of the structure, and surface integral equations based on boundary conditions for both electric and magnetic field intensity vectors are solved with current densities as unknowns. This paper focuses on metallic structures only. However, extending this work to include problems involving dielectrics is straightforward based on [22].

If a structure made of a perfect electric conductor (PEC) is excited by a time-harmonic electromagnetic field of electric field intensity $\mathbf{E}^{inc}$ at the angular frequency ω, then the scattered field $\mathbf{E}^{scat}$ can be expressed in terms of the surface electric currents of density $\mathbf{J}_s$ using the boundary condition for the tangential fields at the surface $S$ of the structure as follows:

$$\left(\mathbf{E}^{scat}\left(\mathbf{J}_s\right)+\mathbf{E}^{inc}\right)_{\tan g}=0\,,\ \mathbf{E}^{scat}\left(\mathbf{J}_s\right)=-\mathrm{j}\omega\mathbf{A}-\nabla\Phi\,, \tag{1}$$

$$\mathbf{A}=\mu\int_S \mathbf{J}_s\, g dS,\ \Phi=\frac{\mathrm{j}}{\omega\varepsilon}\int_S \nabla_s\cdot\mathbf{J}_s\, g dS\,, \tag{2}$$

where $\mathbf{A}$ and $\Phi$ are the magnetic vector and electric scalar potentials, respectively, $g=\mathrm{e}^{-\mathrm{j}\omega\sqrt{\varepsilon\mu}R}/4\pi R$, is the Green's function for the unbounded homogeneous medium with parameters ε and μ, and R is the distance of the field point from the source point. Hence, (1) and (2) constitute an electric field integral equation (EFIE) for $\mathbf{J}_s$ as unknown quantity, which is discretized using the MoM.

### B. Double Higher Order Modeling

Double higher order modeling consists of meshing the geometry of the electromagnetic structure using DHO surface elements, which means that both geometry as well as the unknown variable (surface current) are discretized using higher order functions. In specific, surface of the structure is modeled using generalized curved quadrilaterals of arbitrary geometrical orders $K_u$ and $K_v$, shown in Fig. 1(a), and the current density $\mathbf{J}_s$ over quadrilaterals is approximated by means of hierarchical vector basis functions of arbitrarily high current-expansion orders $N_u$ and $N_v$ [22],

$$\mathbf{r}(u,v)=\sum_{k=0}^{K_u}\sum_{l=0}^{K_v}\mathbf{r}_{kl}L_k^{K_u}(u)L_l^{K_v}(v),\ -1\le u,v\le 1, \tag{3}$$

$$\mathbf{J}_s=\sum_{i=0}^{N_u}\sum_{j=0}^{N_v-1}\alpha_{ij}^{(u)}P_{ij}^{(u)}(u,v)\frac{\mathbf{a}_u}{\Im}+\sum_{i=0}^{N_u-1}\sum_{j=0}^{N_v}\alpha_{ij}^{(v)}P_{ij}^{(v)}(u,v)\frac{\mathbf{a}_v}{\Im} \tag{4}$$

arranged in a maximally orthogonalized fashion [41], [42] as illustrated in Fig. 1(b). Here, $L$ represent Lagrange interpolation polynomials, $\mathbf{r}_{kl}$ are position vectors of interpolation nodes, $P$ are divergence-conforming polynomial bases, $\Im=|\mathbf{a}_u\times\mathbf{a}_v|$ is the Jacobian of the covariant transformation, and $\mathbf{a}_u=\partial\mathbf{r}/\partial u$ and $\mathbf{a}_v=\partial\mathbf{r}/\partial v$ are unitary

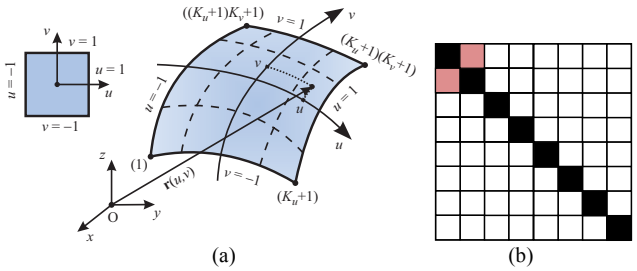



Fig. 1. (a) Generalized curved parametric quadrilateral patch for DHO MoM-SIE modeling [22]. (b) Sketch of the orthogonality factor for the first several maximally orthogonalized hierarchical basis functions of the eighth order [41]. The darkness of each square represents the magnitude of the inner products between pairs of basis functions. The set of basis functions is almost orthonormal.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



vectors along the parametric coordinates. The unknown current-distribution coefficients $\{\alpha\}$ in (4) are determined by solving the SIE in (1), employing Galerkin method. Double (geometrical and current) higher order modeling enables the use of large curved patches, which can greatly reduce the number of unknowns for a given problem and enhance the accuracy and efficiency of the computation.

### C. Geometrical Preprocessing Based on Cobblestone Distance Sorting Technique

The geometrical grouping of the quadrilaterals in the object mesh based on their spatial locality, as shown in Fig. 2, is done by applying the distance sorting technique [15] as outlined as follows:

To divide a number of unknowns into $N^g$ groups we first calculate how many unknowns $N^k$ should be in each group so as to have an even division. A box is created surrounding all remaining mesh patches. We calculate the projection of all boxed patches' centers onto the vector forming the diagonal of the box. The patch with the smallest projection is defined as the zero point for our current mesh group. All unknowns are then sorted by their distance from the zero point, and the first $N^k$ from this sort are assigned to that particular mesh group. These patches are then removed from the mesh and the entire process is repeated until there are no remaining unknowns to be sorted. The groupings generated from this process are illustrated in Fig. 2 for two different geometric shapes.

The distance sorting technique is adapted to more appropriately fit a hierarchical method like the HSS algorithm. Instead of choosing $N^k$ subgroups, the cobblestone method is used to sort the initial mesh into just 2 subgroups. Afterwards, each of these mesh groups is treated as an independent mesh, and individually sorted into 2 subgroups among itself. This process is applied recursively in a binary fashion, until the number of mesh groups is equal to the number of leafs in the desired HSS tree (or the number of processes), as described in Section III. If there are multiple processes per leaf, each mesh group corresponding to a leaf is then sorted appropriately by itself among the remaining processes. This hierarchical division of the mesh matches exactly with the desired partition defined by the HSS tree (Section III).

This grouping technique ensures data locality in the dense system matrix, which is greatly beneficial for achieving the properties needed for the HSS algorithm. Specifically, the cobblestone distance sorting technique divides the mesh into $N^g$ mesh groups. Each matrix sub-block determined by the coordinates $(i,j)$ stores the interactions between MoM unknowns belonging to the $i^{th}$ and $j^{th}$ mesh groups. The mesh size of each mesh group (i.e., the number of unknowns) is predetermined by the number of processes, geometry, and other simulation specifics and is given as an input to the geometrical preprocessor. The outcome of the preprocessor is that MoM unknowns belonging to the same mesh group, along with having their spatial locality, exhibit data locality in the system matrix: self-interaction blocks are on the diagonal, while near interactions tend to be closer to the diagonal and far interactions are further away from the diagonal in the system matrix. It is well known that the numerical rank of the matrix block describing the interactions between two groups decreases with an increase in distance between the groups [15], [43]. This matrix property plays a significant role in matrix compression during the HSS construction step.

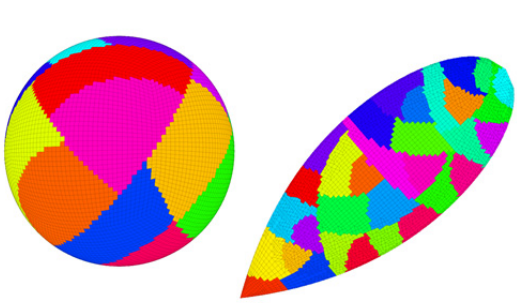


Fig. 2 Illustration of geometrical preprocessing of objects based on the cobblestone distance sorting technique [15].

## III. HSS Theory

This section contains an overview of hierarchically semiseparable (HSS) structures, their construction, factorization and solution, defined and explained in more detail in [34].

### A. HSS Structures

The HSS form of a general $N \times N$ matrix $A$ relies on something called an HSS tree, denoted by $\mathcal{T}$, which defines a hierarchical partitioning of the set of indices of the matrix, $\mathcal{I} = \{1,2,\ldots,N\}$. A binary cluster tree is a binary tree such that every node $i$ has associated with it an index subset, $t_i \subset \mathcal{I}$. We denote the left and right child nodes of a particular node as $c1$ and $c2$, respectively. An HSS tree is a full binary cluster tree such that any non-leaf node $i$'s children $c1$ and $c2$ have the property that $t_{c1} \cap t_{c2} = \emptyset$ and $t_{c1} \cup t_{c2} = t_i$, i.e., a node's children define an even finer partition of that node's index set. Consequently, $A_{|t_i \times t_j}$ denotes the submatrix of $A$ formed by the columns with indices in $t_i$ and the rows with indices in $t_j$

We say that an HSS tree $\mathcal{T}$ is postordered if the nodes in the tree are enumerated in such a way that for every nonleaf node $i \in T$, its children $c1$ and $c2$ satisfy $c1 < c2 < i$. Following this ordering scheme, a full $L$-level postordered HSS tree will consist of $2k$-1 (where $k = 2^{L-1}$) ordered nodes, in which the root node is always numbered as $2k$-1. The index set associated with the root node is $t_{2k-1} = \mathcal{I}$, the full index set. As we traverse down the HSS tree starting from the root, the partitioning of the index set described by the nodes on each level becomes finer and finer. We also number the levels of the tree with the leaf level being 1, and the level number increasing as we get closer to the root. A visualization of a four-level HSS tree and corresponding index partitioning is given in Fig. 3.

With the structure of an HSS tree firmly in place, the HSS compressed representation $\hat{A}$ of a matrix $A$ can be defined from the bottom up by defining a set of so-called generator matrices, $D_i$, $U_i$, $R_i$, $B_i$, $W_i$, and $V_i$, at the $i^{th}$ node of the HSS tree (we discuss HSS construction and corresponding computation of these generator matrices in Section III B.) At

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



the lowest (leaf) level, a matrix $D_i$ corresponds to the block of the original matrix on the diagonal at the finest partition, $A_{|t_i \times t_i}$. With the unknown ordering scheme described in Section II, these diagonal blocks consist primarily of the interactions between basis functions which are spatially close to each other in the MoM-SIE mesh. Because of the strong 1/R spatial dependence of the electrodynamic Green's function, blocks close to the main diagonal of the system matrix are numerically much more significant (higher rank) than the off diagonal blocks [15],[43]. This numerical structure of the system matrix is what allows us to exploit the compressibility of the system matrix during HSS construction. Thus, the HSS algorithm is perfectly suited for its application in many computational electromagnetics problems.

At the higher (non-leaf) levels, a subset of the generator matrices of a node $i$ are block-wise defined in terms of the generators of $i$'s children, $c1$ and $c2$, as so:

$$D_i = \hat{A}_{|t_i \times t_i} = \begin{bmatrix} D_{c1} & U_{c1} B_{c1} V_{c2}^H \\ U_{c2} B_{c2} V_{c1}^H & D_{c2} \end{bmatrix}, \tag{5}$$

$$U_i = \begin{bmatrix} U_{c1} R_{c1} \\ U_{c2} R_{c2} \end{bmatrix}, \quad V_i = \begin{bmatrix} V_{c1} W_{c1} \\ V_{c2} W_{c2} \end{bmatrix}, \tag{6}$$

where the superscript "$H$" denotes the Hermitian transpose. The remaining generator matrices, $R_i$, $B_i$, and $W_i$ are stored at each level. With this structure in place, at the root node $2k$-1, we have that $D_{2k-1} = \hat{A}$, the full HSS compressed representation of $A$. Fig. 4 depicts a block example of the 8 × 8 (4-level) HSS representation of a matrix defined on the corresponding HSS tree given in the Fig. 3. Leaf level $D$ matrices are calculated and stored in a fully dense manner, while other matrices will be calculated and saved in a compressed form obtained by the rank-revealing QR (RRQR) decomposition.

The maximum numerical rank $r$ for a given rank tolerance τ of all compressed blocks is called the HSS rank of matrix $A$. We say that a matrix $A$ has the low-rank property and can be efficiently compressed and solved using the HSS algorithm if $r$ is small comparing to the matrix size.

### B. HSS Construction

HSS construction is done in two stages: first a row compression is applied, followed by a column compression. We utilize RRQR decompositions [13] for matrix compression

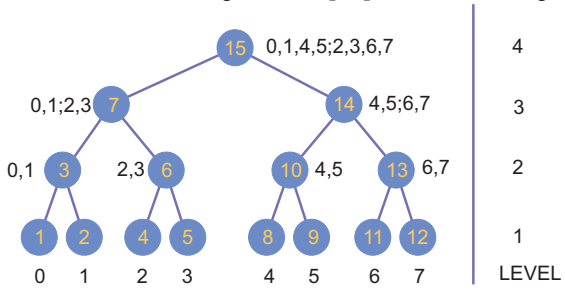



Fig. 3. A four-level postordered HSS tree, along with the associated index sets at each level. Here, the full index set $\mathcal{I} = \{0,1,\ldots,7\}$.

at all stages. The RRQR decomposition is a modified Graham-Schmidt algorithm designed to terminate after a certain rank tolerance is reached.

The RRQR algorithm is as follows: We may compress and approximate a general $M \times N$ matrix $A \approx QT$ with a rank tolerance τ by iterating through the columns (or the first $M$ columns if $M < N$) of the matrix and doing the following at each iteration $i$. We will use the notation $a_i$ to designate the $i^{th}$ column of our current matrix $A$.

1. Find the column of the current matrix $A$ with the largest 2-norm. We call this the $j^{th}$ column, $a_j$.
2. Swap $a_i$ and $a_j$.
3. Set the $i^{th}$ diagonal element of $T$ to be the 2-norm of $a_i$, $t_{ii} = \|a_i\|_2$. If $t_{ii}/t_{11}$ is below the selected tolerance τ, we terminate the algorithm, and we define the rank $r$ of $A$ to be $i$.
4. Calculate the $i^{th}$ column of $Q$ as $q_i = a_i/t_{ii}$.
5. Calculate the remainder of the $i^{th}$ row of $T$ as $t_i^H = q_i^H [a_{i+1}, a_{i+2,\ldots}, a_N]$
6. Update the remaining columns of $A$ as $[a_{i+1}, a_{i+2}, \ldots, a_N] = [a_{i+1}, a_{i+2}, \ldots, a_N] - q_i t_i^H$

The matrix $A$ is now compressed and stored as a low-rank approximation given by the product of an $M \times r$ orthogonal matrix $Q$ and an $r \times N$ upper trapezoidal matrix $T$.

Row compression is applied in an upward sweep along the HSS tree, beginning at the leaf level, where at each node $i$, a local row block of the global system matrix, $A_{|t_i \times I}$, is filled in full as explained in Section II. To this row block, an initial RRQR decomposition is applied to the portion which does not lie on the diagonal block, $A_{|t_i \times I \setminus t_i}$ (the notation $I \setminus t_i$ refers to the set of all indices which are in $I$, but not in $t_i$). We approximate this block via an RRQR decomposition as

$$A_{|t_i \times (I \setminus t_i)} \approx U_i A_{|\hat{t}_i \times (I \setminus t_i)} \tag{7}$$

which defines the column block HSS generators $U_i$ at the leaf

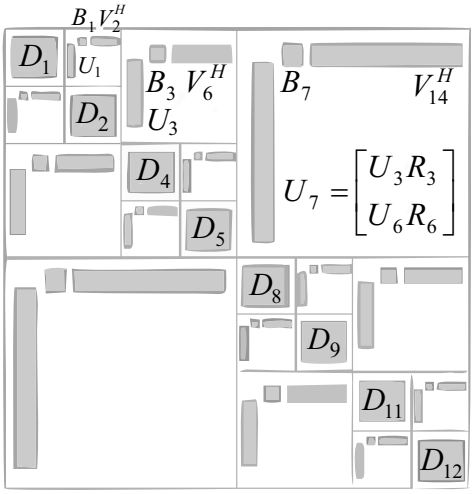



Fig. 4. Illustration of a corresponding HSS form. Diagonal blocks of the matrix are stored in full, while off-diagonal blocks are stored as low-rank approximations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



level nodes. We use the notation $\hat{t}_i$ to express that the new row index set is no longer a part of the global index set, but rather a set of indices corresponding to the new row compressed form of that matrix block.

At every non-leaf level $i$, row compression is done by inheriting submatrices of $i$'s children's row compressed blocks, $A_{|t_{c1}\times I\setminus t_i}$ and $A_{|t_{c2}\times I\setminus t_i}$. Again these correspond to the portions of the compressed matrices which do not lie on the diagonal block at the next level up the HSS tree. These two forwarded blocks are concatenated vertically and then compressed again as:

$$\begin{pmatrix} A_{|\hat{t}_{c1}\times(I\setminus t_i)} \\ A_{|\hat{t}_{c2}\times(I\setminus t_i)} \end{pmatrix} \approx \begin{pmatrix} R_{c1} \\ R_{c2} \end{pmatrix} A_{|\hat{t}_i\times(I\setminus t_i)}. \tag{8}$$

The remaining portions of $i$'s children's corresponding row compressed matrices are left as they were:

$$A_{|t_{c1}\times t_{c2}} \approx U_{c1} A_{|\hat{t}_{c1}\times t_{c2}},\ A_{|t_{c2}\times t_{c1}} \approx U_{c2} A_{|\hat{t}_{c2}\times t_{c1}}, \tag{9}$$

Recursively applying this algorithm upward along the HSS tree until the root node is reached defines all of the necessary row generators $U_i$ and $R_i$. It also defines all row-compressed matrices of the form $A_{|\hat{t}_i\times t_j}$ where $i$ and $j$ are sibling nodes (i.e., nodes which are both children of the same node).

Column compression is applied in a similar fashion via an upward sweep along the HSS tree. We define at every leaf node $i$ a new row index set $\bar{t}_i$, which consists of all the rows in all the row-compressed matrices for which there exist entries in the columns indexed by $t_i$. Formally, this may be expressed as

$$\bar{t}_i = \bigcup_{l=1}^{L-1} \hat{t}_{\mathrm{sib}(\mathrm{ances}(i,l))} \tag{10}$$

where $sib(i)$ refers to the sibling node of $i$ and $ances(i,l)$ refers to $i$'s ancestor node at the $l^{\text{th}}$ level of the HSS tree (note that $ances(i,1) = i$).

Again, RRQR decompositions are performed hierarchically to blocks of the row-compressed matrices to generate the final, fully compressed matrices. The column compression of the local matrix at the leaf level is given by

$$A_{|\bar{t}_i\times t_i} \approx A_{|\bar{t}_i\times\tilde{t}_i} V_i^H; \tag{11}$$

Blocks of these leaf level matrices are again inherited up the tree, just as in the row compression step given by (9). For a non-leaf node $i$, we define the horizontal compression in terms of submatrices of $i$'s children's compressed matrices. We concatenate blocks of the column compressed matrices from each child, and perform the last series of RRQR decompositions which define the generator matrices $W_{c1}$ and $W_{c2}$.

$$\begin{pmatrix} A_{|\bar{t}_i\times\tilde{t}_{c1}} & A_{|\bar{t}_i\times\tilde{t}_{c2}} \end{pmatrix} = A_{|\bar{t}_i\times\tilde{t}_i} \begin{pmatrix} W_{c1} \\ W_{c2} \end{pmatrix}^H, \tag{12}$$

The portions of the compressed matrices from the children which aren't forwarded become the final generator matrices, $B_i$ for the child nodes:

$$B_{c1} = A_{|\hat{t}_{c1}\times\tilde{t}_{c2}},\ B_{c2} = A_{|\hat{t}_{c2}\times\tilde{t}_{c1}}. \tag{13}$$

Construction of the matrix $A$ in its HSS form and all generators corresponding to the non-root nodes are obtained by the completion of the column compression stage. Note that the compression of the right-hand side is done in the same fashion as the compression of the system matrix.

### C. HSS ULV Factorization and Solution

Once the HSS matrix form has been constructed, and all relevant generator matrices have been calculated, an HSS *ULV* factorization is systematically applied to the HSS compressed form of the matrix $A$ in (5) in order to find orthogonal $(U,V)$ and triangular $(L)$ matrices. The ULV factorization is done beginning at the lowest non-leaf level node $i$ by multiplying the local matrix $D_i$ given in (5) by specially constructed orthogonal matrices $Q$ and $P$, calculated from $i$'s children's generators,

$$\begin{bmatrix} Q_{c1}^H & 0 \\ 0 & Q_{c2}^H \end{bmatrix} \begin{bmatrix} D_{c1} & U_{c1}B_{c1}V_{c2}^H \\ U_{c2}B_{c2}V_{c1}^H & D_{c2} \end{bmatrix} \begin{bmatrix} P_{c1}^H & 0 \\ 0 & P_{c2}^H \end{bmatrix}. \tag{14}$$

The $Q$ matrices are formed by standard Gram-Schmidt $QL$ factorizations of the children's column block generator matrices, $U_{c1}$ and $U_{c2,}$ in order to introduce zeros on their off-diagonal row blocks:

$$U_{c1} = Q_{c1}\begin{bmatrix} 0 \\ \tilde{U}_{c1} \end{bmatrix},\ U_{c2} = Q_{c2}\begin{bmatrix} 0 \\ \tilde{U}_{c2} \end{bmatrix}, \tag{15}$$

Now, $\tilde{U}_{c1}$ and $\tilde{U}_{c2}$ are lower square triangular matrices of size $r_{c1}$ and $r_{c2}$, respectively, with $r_{ck}$ being the rank of the matrix $U_{ck}$. The obtained $Q$ matrices are used to further define $\hat{D}_{ck} = Q_{ck}^H D_{ck}, k = 1,2,$ which are partitioned as:

$$\hat{D}_{ck} = \begin{bmatrix} \hat{D}_{ck;1,1} & \hat{D}_{ck;1,2} \\ \hat{D}_{ck;2,1} & \hat{D}_{ck:2,2} \end{bmatrix}, k = 1,2., \tag{16}$$

so that $\hat{D}_{ck:2,2}$ is a square matrix of size equal to $r_{ck}$. Finally, the $P$ matrices are defined by the following $LQ$ factorization:

$$\begin{pmatrix} \hat{D}_{ck;1,1} & \hat{D}_{ck;1,2} \end{pmatrix} = \begin{pmatrix} \tilde{D}_{ck;1,1} & 0 \end{pmatrix} P_{ck}. \tag{17}$$

Following the steps given in (14)-(17) and applying the orthogonality properties of the involved $P$ and $Q$ matrices, (14) may be expressed in a new, more convenient form:

$$\begin{bmatrix} \begin{bmatrix} \tilde{D}_{c1;1,1} & 0 \\ \tilde{D}_{c1;2,1} & \tilde{D}_{c1;2,2} \end{bmatrix} & \begin{bmatrix} 0 \\ \tilde{U}_{c1}B_{c1}\begin{bmatrix} \tilde{V}_{c2;1}^H & \tilde{V}_{c2;2}^H \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} 0 \\ \tilde{U}_{c2}B_{c2}\begin{bmatrix} \tilde{V}_{c1;1}^H & \tilde{V}_{c1;2}^H \end{bmatrix} \end{bmatrix} & \begin{bmatrix} \tilde{D}_{c2;1,1} & 0 \\ \tilde{D}_{c2;2,1} & \tilde{D}_{c2;2,2} \end{bmatrix} \end{bmatrix}. \tag{18}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



It is important to note that all non-zero off-diagonal blocks in (18) are of small dimensions relative to the full matrix. At this stage, it is easy to redefine the generators at the next level up node using only a small part of its children's factorized generators:

$$D_i = \begin{bmatrix} \widetilde{D}_{c1;2,2} & \widetilde{U}_{c1} B_{c1} \widetilde{V}^H_{c2;2} \\ \widetilde{U}_{c2} B_{c2} \widetilde{V}^H_{c1;2} & \widetilde{D}_{c2;2,2} \end{bmatrix}, U_i = \begin{bmatrix} \widetilde{U}_{c1} R_{c1} \\ \widetilde{U}_{c2} R_{c2} \end{bmatrix}, V_i = \begin{bmatrix} \widetilde{V}_{c1;2} W_{c1} \\ \widetilde{V}_{c2;2} W_{c2} \end{bmatrix}. \quad (19)$$

Note that, for example, a square matrix $D_i$ is still saved in a compressed fashion as before, but its new dimension is only $r_{c1} + r_{c2}$, while immediately after the HSS construction stage, it was the size of $t_{c1} \cup t_{c2}$.

The algorithm described in (14)-(19) is then performed in an upward sweep along the HSS tree. When the root node is reached, a dense LU factorization with partial pivoting is performed on a square matrix of dimensions far smaller than those of the starting matrix Equation (19) along with the corresponding HSS compressed form of the excitation matrix, defines the matrix system at the $i^{\text{th}}$ node – these unknowns do not correspond to the final solution, but a partial factorization. Once this is solved on the parent node, it is straightforward to perform elimination on (18) and calculate the solutions corresponding to the children nodes. Solving the matrix is thus done in a downward sweep along the HSS tree – the final solution is obtained once the leaf level is reached.

## IV. Parallelization Strategy

The parallelization strategy of the HSS-MoM-SIE method is adapted from [34] for construction, factorization, and solution, and from [44] for MoM-SIE matrix filling. The communication methods between the processes are designed to fit with the communication layer of the state-of-the-art dense linear algebra libraries ScaLAPACK [45] and BLACS (Basic Linear Algebra Communication Subprograms) [46]. Both libraries are utilized extensively for the execution of algebraic computations on dense matrix blocks.

MoM-SIE matrix filling is done as described in Section II, where the parallel matrix filling builds on top of the geometrical processing as modified from [44]. If the total number of processes running the simulation is $N_{\text{procs}}$, then the number of mesh groups used to partition the mesh during preprocessing is $N^g = \sqrt{N_{procs}}$. The processes form a block partitioning of the matrix, where each matrix block may be described by its coordinates ($p$, $q$), corresponding to the interactions between the $p$-th and $q$-th mesh groups.

Matrix filling is done at the leaf level, where the 2D process grid is divided into $k$ (the number of leaf nodes) contexts, where each context defines a process subgrid of size $N^g/k \times N^g$, as illustrated in Fig. 5 for the case of 16 processes. Each of these subgrids consists of multiple processes operating blockwise on a row chunk of the full matrix. A set of processes operating on a matrix block together like this is known as a context. The context corresponding to

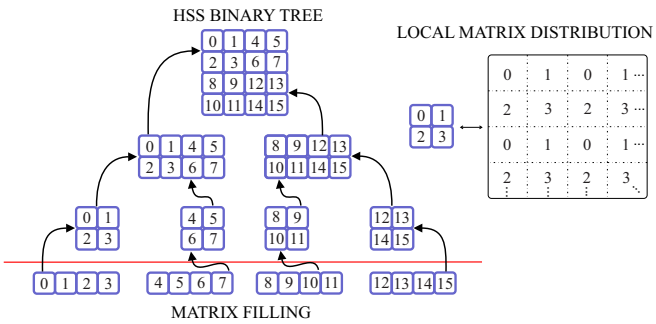



Fig. 5. Illustration of a process context change throughout matrix filling and HSS construction/solution on a level 3 HSS tree using 16 processes. Matrix filling is done by partitioning the matrix into row chunks, while all matrix operations during HSS construction/solution are done utilizing 2D block cyclic distributions. The process grid arrangements in the binary tree represent the block cyclic distribution patterns for the local matrices at each tree node during HSS construction and factorization.

the leaf-level node $i$ is used to calculate the chunk of the MoM-SIE system matrix, $A_{|t_i \times I}$, as well as the corresponding right-hand side vector set.

For optimal communication and computation, dense matrix blocks shared by multiple processes on a context are stored in a distributed 2D block cyclic fashion in accordance with the ScaLAPACK library routines [45]. Any overlap in computation (i.e., elements whose unknowns may be split between processes), may be avoided by effective use of the BLACS communication routines.

Matrix distribution and process grouping at each level of the HSS algorithm follows the basic HSS tree structure. The processes are utilized in such a way that they are combined and split into hierarchical groupings (known as contexts) as the algorithm traverses up and down the tree. At each node of the HSS tree, the process grid (and hence the block cyclic distribution) is redefined in terms of its child nodes' contexts. It inherits the contexts from both nodes, and their process grid is concatenated in order to form a new process grid. This is done hierarchically in a way that preserves as close to a square process grid as possible, as seen in Fig. 5.

During the construction and factorization stages, intercontext communication can largely be avoided, relying largely on only intracontext communication. An example of this communication process can be visualized as in Fig. 6. The communication in HSS construction is visualized here as performed with 8 processes, each corresponding to one row block of the matrix at the row level; immediately following the leaf level row compression, blocks of the row compressed matrices $A_{|\hat{t}_i \times (I \setminus t_i)}$ are pairwise exchanged between pairs of processes as follows: 0 ↔ 1, 2 ↔ 3, 4 ↔ 5, and 6 ↔ 7, as the pairs of processes are merged into new contexts ({0, 1}, {2, 3}, {4, 5}, and {6, 7}). This sets up the matrix in the proper block cyclic distribution for the next level of row compression. At every level during the row compression, to maintain the proper distribution during context merging, all that is required are similar pairwise exchanges, which may all be done in parallel.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



More details behind the communication at each step in the parallel HSS solver can be found in [34]. An example of intracontext communication throughout the computation phase are the following: In the RRQR algorithm, when finding the norm of each column, the norm of each local column is determined and then the BLACS function is used for fast summing in a column-wise fashion for norm calculations on a 2D process grid.

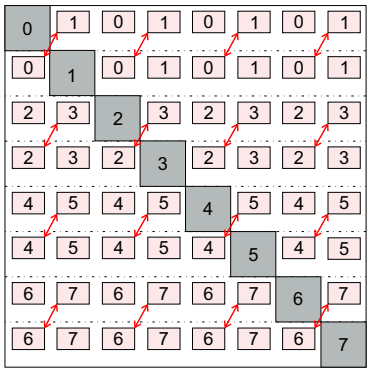


Fig. 6. Illustration of pairwise communication, visualized on 8 processes (one per node of the HSS tree). Immediately after the first row compression stage, the processes have exchanged matrix blocks according to the red arrows. Note that when, for example, processes 6 and 7 merge into one context, the data is in the proper block cyclic distribution for the next level of row compression

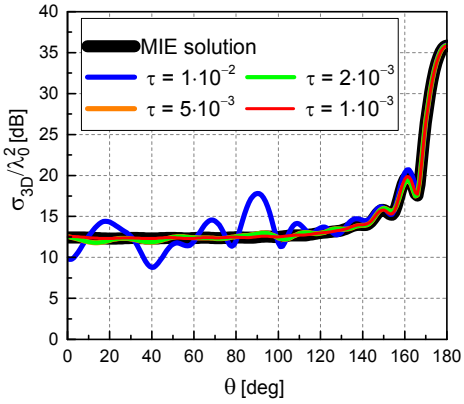
## V. Numerical Results and Discussion

This section provides numerical results obtained by the DHO HSS-MoM-SIE analysis. All simulations were run on the TACC Stampede supercomputer, made accessible through The Extreme Science and Engineering Discovery Environment (XSEDE) [47]. Stampede has 6400 compute nodes where each node contains two Xeon Intel 8-Core 64-bit E5-processors (16 cores on each node). The core frequency is 2.7 GHz and supports 8 floating-point operations per clock period with a peak performance of 21.6 GFLOPS/core or 346 GFLOPS/node. Each node contains 32 GB of memory (2 GB/core). Nodes are interconnected with Mellanox FDR InfiniBand technology in a 2-level fat-tree topology [48]. The first example is a spherical scatterer for comparison with the exact Mie series solutions. This example is well suited for the demonstration of the effectiveness of using higher-order modeling compared to low-order modeling, as well as a strong parallel scalability. The second example is a NASA almond scatterer to show that this method performs well for electrically large problems, as well as objects with complex shapes.

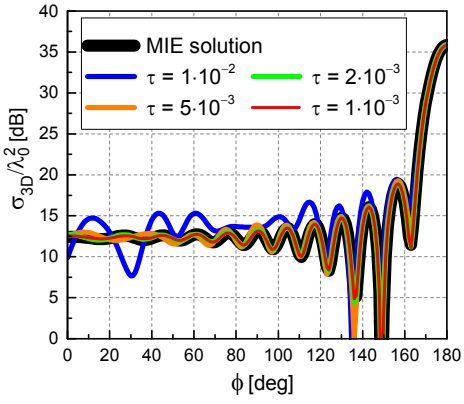
### A. Example 1: Spherical Scatterer

As the first example of our application as well as a validation of the HSS-MoM-SIE method, we consider the scattering analysis of a PEC sphere of diameter $d = 4.666\lambda_0$, where $\lambda_0$ is the free-space wavelength. First, consider a higher-order mesh of the scatterer, of which the allowed size for a single patch is to be less than or equal to a wavelength. After the meshing procedure, the patch size $s$ is approximately $0.9\lambda_0 \le s \le 0.95\lambda_0$ for all 96 geometrically second-order ($K_u = K_v = 2$) curvilinear quadrilateral patches. The adopted current approximation orders are $N_u = N_v = 4$, which results in a total of $N = 3{,}072$ MoM-SIE unknowns. Fig. 7 shows the normalized bistatic radar cross section (RCS), $\sigma_{3D}/\lambda_0^2$, as a function of the scattered angle, in two characteristic plane cuts. The exciting plane wave is incident from the direction defined by $(\theta_{inc}, \phi_{inc}) = (90°,0)$ in spherical coordinates. For the set of results given in Fig. 7, the number of levels in the full postordered HSS tree is chosen to be 5, which results in 16 leaf nodes, while the number of processes used in the parallel simulation is 64. In the same figure, we observe the convergence of results to the analytical Mie solution with the decrease of $\tau$, the user selected RRQR relative tolerance (used in the matrix compressions during the construction step).

In addition, Table I provides information on the average error, maximal rank, memory consumption, and total simulation time (including matrix filling, HSS construction, factorization, and solution times) for different simulations given in Fig. 7. The average relative error is obtained as the average of the absolute error between the normalized bistatic RCS calculated by the numerical method and by the analytical Mie's series, respectively. The averaging is done by taking into account the error in a number ($N^{dir}$) of directions describing the bistatic RCS plane,




(a)




(b)

Fig. 7. Normalized bistatic radar cross section of a spherical PEC scatterer computed by the HSS-MoM-SIE method using preprocessed mesh given in the Fig. 2 and by the Mie's series: (a) $\phi = 0$ cut and (b) $\theta = 90°$ cut.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



$$\zeta = 10\log\frac{\left(\sum_{i=0}^{N^{\mathrm{dir}}}\left|\sigma_i^{\mathrm{num}} - \sigma_i^{\mathrm{MIE}}\right|\right)}{\left(\sum_{i=0}^{N^{\mathrm{dir}}}\left|\sigma_i^{\mathrm{MIE}}\right|\right)} \tag{20}$$

Based on the convergence of different graphs given in Fig. 7, as well as the average errors given in Table I, it can be concluded that the accuracy of the results is easy to control by the relative tolerance used in the RRQR. In addition, by inspecting the results given in Fig. 7 and Table 1, as well as the results of the wide range of performed simulations (different tree levels and RRQR relative tolerance) using the same higher order model, scattering results for the PEC sphere that can readily be considered as accurate are obtained in simulations with the maximal rank $r \geq 490$. Maximal rank in the HSS algorithm corresponds to the maximal number of independent DoFs needed for an accurate numerical simulation [49]. The number of DoFs needed to accurately model a scatterer should depend solely on the scatterer properties and not on the used discretization [37, 43, 49].

Further analysis that compares low- and high-order modeling results shows the advantage of higher order modeling when capturing the real rank of the scattering problem. In particular, the adopted low-order model of the same PEC scatterer consists of 7,776 geometrically first-order ($K_u = K_v = 1$) quadrilaterals with the maximal size of $0.12\lambda_0$. The adopted current approximation orders are $N_u = N_v = 1$, resulting in a total of $N = 15{,}552$ unknowns. After the initial discretization, when compared to the higher order model, the number of unknowns in the low-order model is more than 5 times larger. However, due to the physical properties of the scatterer, the maximal numerical rank of the compressed HSS matrix should be approximately the same for both models.

On the other hand, Fig. 8, shows the relative error metric, $\sigma_i^{\mathrm{num}}/\sigma_i^{\mathrm{MIE}}$, in the RCS for three choices of low-order simulations and one higher-order simulation, confirming that in order to achieve similar accuracy, a low-order simulation needs about one-and-a-half times the numerical rank in the compressed system. This is confirmed by the information in Table II that contains the average error for the four simulations given in Fig. 8.

Further, note that the true low-order modeling based on RWG functions [1] defined over flat triangular patches uses from 500 to 600 unknowns per square wavelength [17, 18, 48], which leads to a truly low order model of a PEC sphere with around 35,000 to 40,000 unknowns, making the number of unknowns more than 10 times larger than in the adopted higher order model.

TABLE I
SIMULATION PARAMETERS FOR THE RESULTS GIVEN IN FIG. 7

| RRQR rel. tolerance | Maximal rank | Time [s] | Storage [GB] | $\zeta$ [dB] ($\phi = 0$ cut) | $\zeta$ [dB] ($\theta = 90°$ cut) |
|---|---|---|---|---|---|
| $\tau = 1\cdot10^{-2}$ | 311 | 6.55 | 0.04 | -11.823 | -10.412 |
| $\tau = 5\cdot10^{-3}$ | 351 | 6.90 | 0.04 | -16.700 | -17.189 |
| $\tau = 2\cdot10^{-3}$ | 399 | 6.94 | 0.05 | -21.279 | -20.970 |
| $\tau = 1\cdot10^{-3}$ | 436 | 7.08 | 0.06 | -22.118 | -21.936 |
| $\tau = 5\cdot10^{-4}$ | 468 | 7.24 | 0.07 | -22.002 | -22.064 |
| $\tau = 2\cdot10^{-4}$ | 510 | 7.83 | 0.08 | -22.097 | -22.079 |

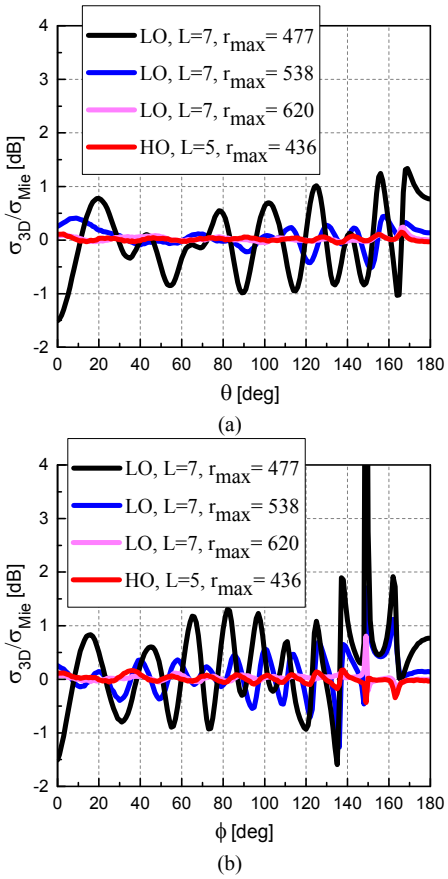



Fig. 8. Error of the normalized bistatic RCS of the PEC scatterer computed by the HSS-MoM-SIE method with respect to the exact Mie solution: (a) $\phi = 0$ cut and (b) $\theta = 90°$ cut. We compare low-order (LO) modeling with higher order (HO) models. $L$ denotes the number of levels in the HSS tree.

Next, we test the scalability performance of the HSS-MoM-SIE method – in the same example. Because the higher order model with only 96 quadrilaterals is not well fitted for simulations on hundreds of processes, the scalability testing of the method is done on a low-order model of the sphere. All simulations used in the test are set up for the tolerance $\tau = 5\cdot10^{-4}$ and level 5 full postordered HSS tree. Note that the number of leaves in the tree is 16, which, for the particular example, corresponds to the minimal number of processes that can be used in a parallel run of the HSS-MoM-SIE code. The runtime of the parallel code on 16 processes is thus adopted to be the baseline computational time used in the speed-up calculations. To measure scalability, we run the same model on 64, 256, and 1,024 processes, and observe an excellent scalability performance in Fig. 9.

### *B. Example 2: NASA Almond*

In the second example, the HSS-SIE-MoM code is applied to analyze scattering from a NASA almond [50], an established benchmarking structure for monostatic RCS computations. In specific, we consider a PEC almond of the maximal size $42\lambda_0$ at a frequency of 50 GHz, with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2017.2673660, IEEE Transactions on Antennas and Propagation



overall surface of the scatterer being equal to $1{,}111\,\lambda_0^2$. The constructed higher order model of the almond uses a total of 16,384 curvilinear quadrilateral elements with $K_u = K_v = 2$ and the current approximation in different directions on different patches ranging from $N_{u/v} = 1$ to $N_{u/v} = 3$ depending on the electrical dimensions of the quadrilateral element. The final number of unknowns (that would be even smaller if larger patches and even higher $N_{u/v}$ were used) is 149,756. A similar scatterer is analyzed in [40] using the model with around 450,000 unknowns. In addition, applying, for comparison, the true lower order quadrilateral modeling to the almond scatterer requires 524,288 unknowns defined over 262,144 patches with $K_u = K_v = 1$ and $N_{u/v} = 1$ on all the patches.

Fig. 10 shows the monostatic scattering computations of the DHO model of the PEC almond obtained by the HSS-MoM-SIE method and validated by the full-storage direct solver using ScaLAPACK LU decomposition [44] simulation of the same model, as well as against the low-order modeling results [40]. The normalized monostatic RCS is calculated for 361 different directions, in the $z = 0$ plane, with the polarization of the incident electric field along the $z$-axis.

HSS construction for this example is done using a relative tolerance of $\tau = 3{\cdot}10^{-4}$ on a 5 level HSS tree. The relative rank tolerance $\tau$ was selected heuristically in such a way as to satisfy the needs for the electrical size of the problem. In general, scatterers which are electrically large will require a smaller rank tolerance than electrically small scatterers to maintain the same level of solution accuracy, so the second example will require a smaller $\tau$. The maximal rank of the compressed matrix in this example comes out to be 3,926. The simulation is run in parallel on 256 processes, with matrix calculation and HSS construction times being 846 s and 2,596 s, respectively. Further, the HSS factorization time is 41.95 s, while the total backsubstitution time for all 361 excitation vectors is 5.36 s. ScaLAPACK LU decomposition time on the same number of processes is 8,565 s, which, in terms of the performance can be compared to total time of the HSS construction and factorization: 2,638 s. The compressed matrix storage in the DHO HSS-MoM-SIE simulation amounts to 20.78 GB, while the full matrix storage for the same model would require 180 GB of memory. In addition, the LO quadrilateral model described above would require 2.2 TB. Hence, we observe great advantages of the DHO modeling coupled with the direct solver and compression of the MoM-SIE matrix in the analysis of electrically large objects with multiple excitations (right-hand side values).

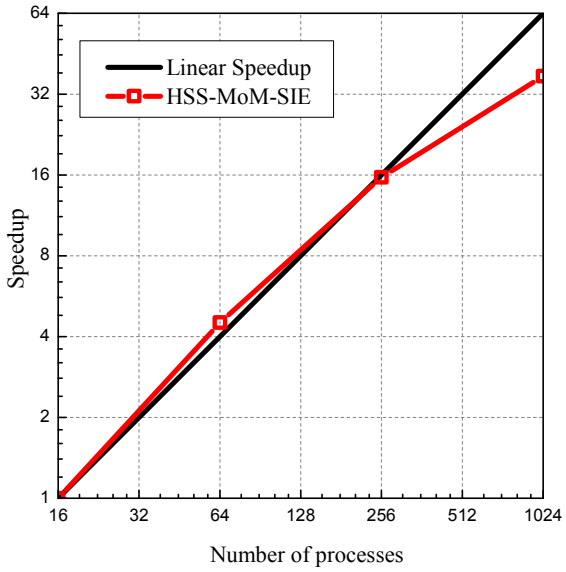



Fig. 9. Performance and scalability of the HSS-MoM-SIE method applied to the simulation of a low-order PEC sphere model with the number of levels in the HSS tree equal to 5.

TABLE II
SIMULATION PARAMETERS FOR THE RESULTS GIVEN IN FIG. 8

| Model type | RRQR rel. tolerance | Maximal rank | $\zeta$ [dB] ($\phi = 0$ cut) | $\zeta$ [dB] ($\theta = 90°$ cut) |
|---|---|---|---|---|
| High Order | $\tau = 1{\cdot}10^{-3}$ | 436 | -22.118 | -21.936 |
| Low Order | $\tau = 1{\cdot}10^{-2}$ | 477 | -6.901 | -7.767 |
| | $\tau = 5{\cdot}10^{-3}$ | 538 | -14.152 | -14.485 |
| | $\tau = 2{\cdot}10^{-3}$ | 620 | -23.619 | -27.039 |

## VI. CONCLUSIONS

This paper has proposed a novel fast scalable parallel algorithm and solver for large scattering problems based on double (geometrical and current-approximation) higher order MoM in the SIE formulation and the frequency domain in conjunction with a direct solver for dense linear systems with hierarchically semiseparable structures, namely, with a HSS matrix representation for compression, factorization, and solution of the system matrix. In addition, an RRQR decomposition for memory compression has been used, with a stopping criterion in terms of the relative tolerance value, allowing for the method to store only the low-rank approximation of the original matrix that satisfies predefined accuracy. In order to enhance the HSS construction and parallelization, a method for geometrical preprocessing of the scatterers based on the cobblestone distance sorting technique

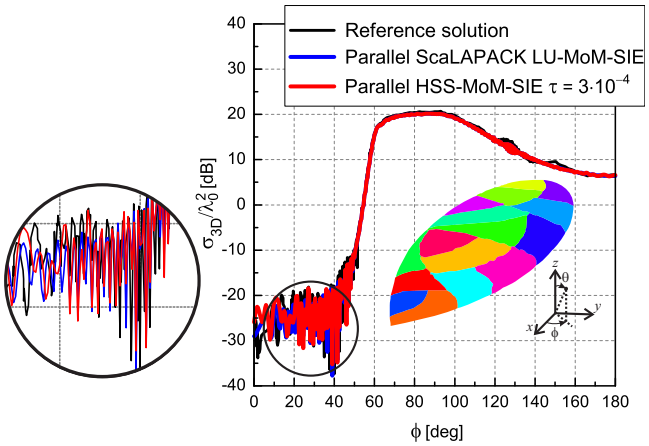



Fig. 10. Normalized monostatic RCS of a PEC NASA almond computed at 50 GHz by the full-storage direct ScaLAPACK LU-MoM-SIE and HSS-MoM-SIE methods. Note that the full LU factorization solution and HSS-MoM solutions (both from same geometrical model) show good agreement.

has been employed, such that the MoM unknowns belonging to the same mesh group and thus having spatial locality also exhibit the data locality in the matrix system of equations.

Numerical examples have shown how the accuracy of the DHO HSS-MoM-SIE method is easily controllable by using the relative tolerance for the matrix compressions. Moreover, the examples have demonstrated low memory consumption, as well as much faster simulation times, when compared to direct LU decompositions. Finally, great scalability of the algorithm has been demonstrated on more than thousand processes.

Overall, the DHO HSS-MoM-SIE method and its future extensions and advancements are asymptotically faster direct algorithms for IE solutions that are memory and communication-efficient and amenable to extreme-scale parallel computing. They also are purely algebraic and kernel-independent and enable dramatically faster monostatic scattering and other multiple-excitation computations than iterative solvers. The combination of the DHO and HSS approaches, along with the distance sorting technique and parallel method, fully utilizes problem's properties to reduce the solution complexity (both computation time and storage) by minimizing the problem's DoFs based on both its physical and algebraic properties. The presented coupling and implementation of the two approaches is a natural way to advance the MoM-SIE modeling, since compressing a low-order matrix may be considered less efficient. For example, based on the presented numerical results, it can be concluded that the prime problem size reduction is obtained by combining the DHO and HSS approaches together.

This paper has focused on metallic scatterers. Nonetheless, due to the truly algebraic nature of the method, its extension to any electromagnetic system solved by the MoM-SIE analysis, or any other IE method, is straightforward.

## VII. Acknowledgement

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575.

## References

[1] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.

[2] W. Hackbusch, "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices," *Computing*, Vol. 62, pp. 89–108, 1999.

[3] W. Hackbusch and B. N. Khoromskij, "A sparse $\mathcal{H}$-matrix arithmetic. Part II: Application to multi-dimensional problems," *Computing*, Vol. 64, pp. 21–47, 2000.

[4] W. Hackbusch, L. Grasedyck, and S. Börm, "An introduction to hierarchical matrices," *Math. Bohem.*, Vol. 127, pp. 229-241, 2002.

[5] S. Börm, "$\mathcal{H}2$-matrices–Multilevel methods for the approximation of integral operators," *Computing and Visualization in Science*, Vol. 7, no. 3–4, pp. 173–181, Aug. 2004.

[6] W. Chai and D. Jiao, "An $\mathcal{H}2$-Matrix-Based Integral-Equation Solver of Reduced Complexity and Controlled Accuracy for Solving Electrodynamic Problems," *IEEE Trans. Antennas Propag.*, Vol. 57, no. 10, pp. 3147–3159, Oct. 2009.

[7] W. Chai and D. Jiao, "Dense matrix inversion of linear complexity for integral-equation based large-scale 3-D capacitance extraction," *IEEE Trans. Microw. Theory Tech*, Vol. 59, no. 10, pp. 2404–2421, Oct. 2011.

[8] T. Wan, Z. N. Jiang, and Y. J. Sheng. "Hierarchical matrix techniques based on matrix decomposition algorithm for the fast analysis of planar layered structures," *IEEE Trans. Antennas Propag.*, Vol. 59, no. 11, pp. 4132–4141, Nov. 2011.

[9] W. Chai and D. Jiao, "Linear-complexity direct and iterative integral equation solvers accelerated by a new rank-minimized representation for large-scale 3-D interconnect extraction," *IEEE Trans. Microw. Theory Tech*, Vol. 61, no. 8, Aug. 2013.

[10] W. Chai and D. Jiao, "Fast-Matrix-Based Direct Integral Equation Solver With Reduced Computational Cost for Large-Scale Interconnect Extraction," *IEEE Trans. Compon. Packag. Manuf. Technol.*, Vol. 3, no. 2, Feb. 2013.

[11] A. Heldring, J. M. Rius, J. M. Tamayo, J. Parrón, and E. Ubeda, "Fast direct solution of method of moments linear system," *IEEE Trans. Antennas Propag.*, Vol. 55, no. 11, pp. 3220–3228, Nov. 2007.

[12] G. H. Golub and C. V. Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[13] M. Gu, and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing QR factorization," *SIAM J. Sci. Comput.*, Vol. 17, no. 4, pp. 848–869, Jul. 1996.

[14] K. Zhao, M. N. Vouvakis, and J. F. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems," *IEEE Trans. EMC*, Vol. 47, no. 4, pp. 763–773, Nov. 2005.

[15] J. Shaeffer, "Direct solve of electrically large integral equations for problem sizes to 1 M unknowns," *IEEE Trans. Antenna Propag.*, Vol. 56, no. 8, pp. 2306–2313, Aug. 2008.

[16] J. Laviada, F. Las-Heras, M. R. Pino, and R. Mittra, "Solution of electrically large problems with multilevel characteristic basis functions," *IEEE Trans. Antennas Propag.*, Vol. 57, no. 10, pp. 3189–3198, Oct. 2009.

[17] J. M. Tamayo, A. Heldring, and J. M. Rius, "Multilevel adaptive cross approximation," *IEEE Trans. Antennas Propag.*, Vol. 59, pp. 4600–4608, Dec. 2011.

[18] A. Heldring, J. M. Tamayo, C. Simon, E. Ubeda, and J. M. Rius, "Sparsified adaptive cross approximation algorithm for accelerated method of moments computations," *IEEE Trans. Antennas Propag.*, Vol. 61, no. 1, pp. 240–246, Jan. 2013.

[19] F. Gesztesy, and K. A. Makarov, "(Modified) Fredholm determinants for operators with matrix-valued semi-separable integral kernels revisited," *Integral Equations and Operator Theory* , Vol. 47, no. 4, pp. 457–497, 2003.

[20] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, AJ. van der Veen, and D. White, "Some fast algorithms for sequentially semiseparable representations," *SIAM J. Sci. Comput.*, Vol. 27, no. 2, pp. 341–364, 2015.

[21] E. Corona, PG. Martinsson, and D. Zorin, "An O (N) direct solver for integral equations on the plane," *Appl. Comput. Harmon. Anal.*, Vol. 38, no. 2, pp. 284–317, Mar. 2015.

[22] M. Djordjevic and B. M. Notaros, "Double Higher Order Method of Moments for Surface Integral Equation Modeling of Metallic and Dielectric Antennas and Scatterers," *IEEE Trans. Antennas Propag*, Vol. 52, no. 8, pp. 2118–2129, Aug. 2004.

[23] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, Vol. 30, no. 3, pp. 409–418, May 1982.

[24] B. M. Kolundzija and A. R. Djordjevic, *Electromagnetic Modeling of Composite Metallic and Dielectric Structures*. Boston, MA: Artech House, 2002.

[25] K. C. Donepudi, J.-M. Jin, and W. C. Chew, "A higher order multilevel fast multipole algorithm for scattering from mixed conducting/dielectric bodies," *IEEE Trans. Antennas Propag.*, Vol. 51, no. 10, pp. 2814–2821, Oct. 2003.

[26] O. Borries, P. Meincke, E. Jørgensen, and P. C. Hansen, "Multi-level fast multipole method for higher-order discretizations," *IEEE Trans. Antennas Propag.*, Vol. 62, no. 9, pp. 4695–4705, 2014.

[27] L. P. Zha, Y. Q. Hu, and T. Su, "Efficient surface integral equation using hierarchical vector bases for complex em scattering problems," *IEEE Trans. Antennas and Propag.*, Vol. 60, no. 2, pt. 2, pp. 952–957, Feb. 2012

[28] X. Mu, H.-X. Zhou, K. Chen and W. Hong, "Higher order method of moments with a parallel out-of-core LU solver on GPU/CPU platform", *IEEE Trans. Antennas Propag.*, Vol. 62, no. 11, pp.5634–5646, Nov. 2014.

[29] Y. Zhang, Z. Lin, X. Zhao, and T. K. Sarkar, "Performance of a Massively Parallel Higher-Order Method of Moments Code Using Thousands of CPUs and Its Applications," *IEEE Trans. Antennas Propag.*, Vol. 62, no. 12, pp. 6317–6324, Dec. 2014.

[30] H. Guo, J. Hu, and Z. Nie, "An MPI-OpenMP hybrid parallel $\mathcal{H}$-LU direct solver for electromagnetic integral equations," *International Journal of Antennas and Propagation*, Vol. 2015, Article ID 615743, 10 pages, 2015.

[31] A. Schroder, H.D. Bruns, and C. Schuster, "Fast evaluation of electromagnetic fields using a parallelized adaptive cross approximation," *IEEE Trans. Antennas Propag.*, Vol. 62, No. 5, 2818–2822, May. 2014.

[32] M. Djordjevic and B. M. Notaros, "Higher-Order Hierarchical Basis Functions with Improved Orthogonality Properties for Moment-Method Modeling of Metallic and Dielectric Microwave Structures," *Microw. Opt. Techn. Let.*, Vol. 37, No. 2, pp. 83–88, Apr. 2003.

[33] B. M. Notaros, B. D. Popovic, J. P. Weem, R. A. Brown, and Z. Popovic, "Efficient large-domain MoM solutions to electrically large practical EM problems," *IEEE Trans. Microw. Theory Tech*, Vol. 49, pp.151–159, Jan. 2001.

[34] S. Wang, X. S. Li, J. Xia, Y. Situ, and M. V. De Hoop, "Efficient Scalable Algorithms for Solving Dense Linear Systems with Hierarchically Semiseparable Structures," *SIAM J. Sci. Comput.*, Vol. 35, no. 6, pp. C519–C544, 2013.

[35] A. B. Manic, F.-H. Rouet, X. S. Li, and B. M. Notaros, "Efficient EM Scattering Analysis Based on MoM, HSS Direct Solver, and RRQR Decomposition," *Proceedings of 2015 IEEE International Symposium on Antennas and Propagation*, July 19-24, 2015, Vancouver, BC, Canada, pp. 1660–1661.

[36] B. M. Notaros, A. B. Manic, X. S. Li, and F.-H. Rouet, "Controlling the Accuracy of Double Higher Order Surface Integral Equation Modeling by Relative Tolerance for Matrix Compression," Invited Paper, *Proceedings of the 2016 Applied Computational Electromagnetics Society Conference – ACES 2016*, March 13-17, 2016, Honolulu, HI, pp. 165–166.

[37] E. Michielssen and A. Boag, "A multilevel matrix decomposition algorithm for analyzing scattering from large structures," *IEEE Trans. Antennas Propag.*, Vol. 44, no. 8, pp. 1086–1093, Aug. 1996.

[38] G. Han, H. Jun, and E. Michielssen, "On MLMDA/Butterfly compressibility of inverse integral operators," *IEEE Antennas Wireless Propag. Lett.*, Vol. 12, pp. 31–34, 2013.

[39] Y. Brick, V. Lomakin, A. Boag, "Fast Direct Solver for Essentially Convex Scatterers Using Multilevel Non-Uniform Grids," *IEEE Trans. Antennas Propag* , Vol. 62, no. 8, pp.4314–4324, Aug. 2014.

[40] A. Heldring, J. M. Rius, J. M. Tamayo, J. Parrón, and E. Úbeda, "Multiscale compressed block decomposition for fast direct solution of method of moments linear system," *IEEE Trans. Antennas Propag.*, vol. 59, no. 2, pp. 526–536, Feb. 2011.

[41] M. Kostic and B. Kolundzija, "Maximally orthogonalized higher order bases over generalized wires, quadrilaterals, and hexahedra," *IEEE Trans. Antennas Propag.*, vol. 61, no. 6, pp. 3135–3148, Jun. 2013.

[42] I. Bogaert, and F. P. Andriulli, "Maximally Orthogonal High-Order Basis Functions have a Well-Conditioned Gram Matrix," *IEEE Trans. Antennas Propag.*, Vol. 62, no. 8, pp. 4096–4104, Aug. 2014.

[43] A. Heldring, J. M. Tamayo, and J. M. Rius, "On the degrees of freedom in the interaction between sets of elementary scatterers," presented at the 3rd Eur. Conf. Antennas Propag. (Eucap), Berlin, Germany, Mar. 23–27, 2009.

[44] A. B. Manic, E. Chobanyan, M. M. Ilic, and B. M. Notaros, "Parallelization of Double Higher Order FEM and MoM Techniques" *Proceedings of 2014 IEEE Antennas and Propagation Society International Symposium*, July 6–12, 2014, Memphis, Tennessee.

[45] ScaLAPACK—Scalable Linear Algebra PACKage, http://www.netlib.org/scalapack.

[46] *BLACS*, http://www.netlib.org/blacs.

[47] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. Ray Scott, N. Wilkins-Diehr, "XSEDE: Accelerating Scientific Discovery", *Computing in Science & Engineering*, Vol.16, no. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80

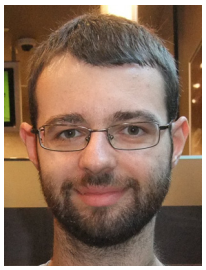
[48] https://portal.tacc.utexas.edu/user-guides/ stampede#overview

[49] O. M. Bucci and G. Franceschetti, "On the degrees of freedom of scattered fields," *IEEE Trans. Antennas Propag.*, Vol. 37, no. 7, pp. 918–926, July 1989.

[50] A. C. Woo, H. T. G. Wang, M. J. Schuh, and M. L. Sanders, "Benchmark radar targets for the validation of computational electromagnetic programs," *IEEE Antennas Propagat. Mag.*, Vol. 35, pp. 84–89, Feb. 1993.

**Ana B. Manić** was born in Belgrade, Serbia, in 1986. She graduated with B.S. from the Department of Electronics in the School of Electrical Engineering at Belgrade University in 2009. She received her Ph.D. degree at Colorado State University, Department of Electrical and Computer Engineering, in 2015. During her Ph.D. study, she worked on different applications of the method of moments in computational electromagnetics such as: method of moments coupling with the finite element method, diakoptics domain decomposition method, fast direct solvers, accurate and efficient calculation of singular and hypersingular integrals arising in the Galerkin-type method of moments system of equations, etc.

**Aaron P. Smull** (S'15) was born in Santa Rosa, California in 1993. He received a B.S. (2015) and is pursuing an M.S. in Electrical Engineering from Colorado State University, and will soon be pursuing his Ph.D. His current research interests include the development of numerical algorithms for high frequency electromagnetic scattering analysis and the design of novel electromagnetic and optical devices based on exotic material parameters.



**François-Henry Rouet** was born in La Roche-sur-Yon, France, in 1985. He received M.Sc. and Ph.D. degrees in applied mathematics and computer science from the Institut National Polytechnique de Toulouse, France, in 2009 and 2012, respectively.

He was a postdoctoral research fellow in the Scalable Solvers Group at the Lawrence Berkeley National Laboratory from 2012 to 2015. He is currently a research scientist at Livermore Software Technology Corporation, California, USA, where he works on the multi-physics finite element code LS-DYNA. His fields of interest are parallel computing, sparse matrix computations, and graph algorithms. He also collaborates on MUMPS and STRUMPACK linear algebra projects.

**Xiaoye Sherry Li** is a Senior Scientist in the Computational Research Division, Lawrence Berkeley National Laboratory. She earned Ph.D. in Computer Science from UC Berkeley in 1996.

She has worked on diverse problems in high performance

scientific computations, including parallel computing, sparse matrix computations, high precision arithmetic, and combinatorial scientific computing. She has (co)authored over 95 publications, and contributed to several book chapters. She is the lead developer of SuperLU, a widely-used sparse direct solver, and has contributed to the development of several other mathematical libraries, including ARPREC, LAPACK, PDSLin, STRUMPACK, and XBLAS. She has collaborated with many domain scientists to deploy the advanced mathematical software in their application codes, including those from accelerator engineering, chemical science, earth science, plasma fusion energy science, and materials science.

Dr. Li has served on the editorial boards of the SIAM J. Scientific Comput. and ACM Trans. Math. Software, as well as many program committees of the scientific conferences. She is a SIAM Fellow and an ACM Senior Member.

**Branislav M. Notaroš** (M'00-SM'03-F'16) was born in Zrenjanin, Yugoslavia, in 1965. He received the Dipl.Ing. (B.S.), M.S., and Ph.D. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1988, 1992, and 1995, respectively.

From 1996 to 1999, he was an Assistant Professor in the School of Electrical Engineering at the University of Belgrade. He spent the 1998-1999 academic year as a Visiting Scholar at the University of Colorado at Boulder. He was an Assistant Professor, from 1999 to 2004, and Associate Professor, from 2004 to 2006, in the Department of Electrical and Computer Engineering at the University of Massachusetts Dartmouth. From 2006 to 2012, he was an Associate Professor in the Department of Electrical and Computer Engineering at Colorado State University, where he is currently a Professor and University Distinguished Teaching Scholar, as well as Director of Electromagnetics Laboratory. His research interests and activities are in computational electromagnetics, higher order numerical methods, antennas, scattering, microwaves, metamaterials, characterization of snow and rain, surface and radar precipitation measurements, RF design for MRI at ultra-high magnetic fields, and electromagnetics education. His publications include more than 200 journal and conference papers, and three workbooks in electromagnetics and in fundamentals of electrical engineering (basic circuits and fields). He is the author of textbooks *Electromagnetics* (Prentice Hall, 2010), *MATLAB-Based Electromagnetics* (Prentice Hall, 2013), and *Conceptual Electromagnetics* (CRC Press, 2017).

Dr. Notaroš is Fellow of the IEEE and of the Applied Computational Electromagnetics Society (ACES). He served as General Chair for the 11th International Workshop on Finite Elements for Microwave Engineering – FEM2012, June 4-6, 2012, Estes Park, Colorado, USA, and as Guest Editor of the Special Issue on Finite Elements for Microwave Engineering, Electromagnetics, Vol. 34, Issue 3-4, 2014. He serves on the Board of Directors of ACES (2016-2019) and as Chair of the Technical Committee for the Commission B of the US National Committee, International Union of Radio Science (URSI) (since 2014). He was the recipient of the 2005 IEEE MTT-S Microwave Prize (best-paper award for IEEE Transactions on MTT), 1999 IEE Marconi Premium (best-paper award for IEE Proceedings on Microwaves, Antennas and Propagation), 1999 URSI Young Scientist Award, 2005 UMass Dartmouth Scholar of the Year Award, 2004 UMass Dartmouth College of Engineering Dean's Recognition Award, 1992 Belgrade Chamber of Industry and Commerce Best M.S. Thesis Award, 2009, 2010, 2011, and 2014 Colorado State University Electrical and Computer Engineering Excellence in Teaching Awards, 2010 Colorado State University College of Engineering George T. Abell Outstanding Teaching and Service Faculty Award, 2012 Colorado State University System Board of Governors Excellence in Undergraduate Teaching Award, 2014 Colorado State University Provost's N. Preston Davis Award for Instructional Innovation, 2012 IEEE Region 5 Outstanding Engineering Educator Award, 2014 Carnegie Foundation for the Advancement of Teaching Colorado Professor of the Year Award, 2015 American Society for Engineering Education ECE Distinguished Educator Award, and 2015 IEEE Undergraduate Teaching Award.