Compressed Fast Multipole Representations for Homogeneous 3-D Kernels

R. J. Adams[1], J. C. Young[1], and S. D. Gedney[2]

[1]Electrical & Computer Engineering
University of Kentucky, Lexington, KY, USA
rjadams@uky.edu, john.c.young@uky.edu

[2]Electrical Engineering
University of Colorado Denver, Denver, CO, USA
stephen.gedney@ucdenver.edu

***Abstract* –** For homogeneous kernels, the memory requirements associated with H^2 representations of integral equation matrices can be reduced by incorporating translational invariance. Starting with a non-translationally invariant H^2 representation, this can be accomplished using a left/right iterative algorithm. In this paper, it is shown that a similar algorithm can also be used to compress an existing fast multipole method (FMM). It is observed that the iterative compression converges faster when used to compress an FMM than when it is applied to an H^2 representation. Resulting savings in floating-point operations are indicated, and extensions of the reported method are discussed.

***Index Terms* –** fast multipole method, integral equation.

I. INTRODUCTION

Integral equation (IE) based formulations provide an effective method for formulating 3D electromagnetic interaction problems over a range of frequencies. When modeling fields on large and/or complex domains, it is necessary to use compressed representations of the generally dense system matrix that results from the use of an IE formulation. For static and low-frequency electromagnetic applications, fast multipole methods (FMM) [1, 2] and the H^2 representations [3] provide controllably accurate representations of IE system matrices and have $O(N)$ complexity, where N indicates the number of unknowns in the discretized IE formulation.

Although similar in many ways, the FMM and H^2 representations of integral equation matrices differ in how they represent interactions between source and field groups. In an FMM, all interactions are represented using a common (e.g., multipole) basis. For translationally invariant kernels, this enables significant time and memory savings when building the FMM since only a relatively small number of unique translators are needed at each level of an octree decomposition (at most 316 for a homogeneous kernel).

In contrast, the H^2 representation is often developed from sparse samples of the underlying matrix [4]. (The H^2 representation in [4] is therein referred to as an MLSSM representation; the MLSSM is equivalent to an H^2, as indicated by equation (21) of [4].) Since the underlying geometry is not translationally invariant, the H^2 representation obtained via sparse matrix samples does not retain the translational invariance of the underlying kernel, and each translation matrix is unique. A result is that the time required to build an H^2 representation can be significantly longer than the time required to build a similarly accurate FMM.

Although the time required to build an FMM can be much less than the time required to build a similarly accurate H^2, this saving comes at the expense of requiring larger translation matrices. This increased dimension of the FMM translators can lead to higher costs for matrix-vector product operations when using an FMM versus an H^2 representation. Furthermore, when fast direct solvers such as the $O(N)$ H^2 factorization of [4] are used, the larger translators of the FMM can also yield increased factorization costs relative to an H^2. These additional costs can offset the relative computational savings provided by an FMM when constructing a sparse representation of the system matrix.

It was recently shown that, for translationally invariant kernels, it is possible to reduce memory costs associated with an H^2 representation by converting a non-translationally invariant H^2 matrix into a translationally invariant H^2 representation using an iterative procedure [5, 6]. However, the computational costs of the algorithms used to compress the H^2 have been found to be too large to be practically useful.

In the remainder, a similar algorithm is reported for compressing an existing FMM representation [2]. It is found that the computational costs to compress an FMM

Submitted On: August 1, 2023
Accepted On: March 27, 2024

https://doi.org/10.13052/2024.ACES.J.390201

are significantly less than the costs reported in [5, 6] for compressing an H^2.

II. SHIFT INVARIANT FORM

At a given level of an octree, the interactions between non-touching groups in an FMM can be represented as:

$$\mathbf{Z}_{\text{far}} = \mathbf{U}\,\mathbf{T}_{SI}\,\mathbf{V}^h, \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are block-diagonal matrices containing the FMM aggregation and disaggregation operators that map between the multipole representations of parent and child groups (or between multipole representation and the unknowns, if at the finest level). The matrix $\mathbf{T}_{SI}$ contains all translators at the level, of which only 316 (at most) are unique. Further details are provided in [2].

To compress the FMM, we first compute a compressed representation $\mathbf{T}$ from $\mathbf{T}_{SI}$. The matrix $\mathbf{T}$ is obtained by computing the $O(\tau)$ truncated SVD of column- and row-blocks of the FMM translation matrix $\mathbf{T}_{SI}$. The resulting singular vectors are then used to project the column-/row-blocks of $\mathbf{T}_{SI}$ yielding $\mathbf{T}$. Herein, the SVD truncation tolerance τ is selected to be equal to the accuracy of the FMM representation divided by 10. (In the numerical examples below, τ=1e-7.)

Once $\mathbf{T}$ is obtained, block diagonal matrices $\mathbf{L}$ and $\mathbf{R}$ are computed such that:

$$\mathbf{T} = \mathbf{L}\mathbf{T}_{\text{SI}}\mathbf{R}, \tag{2}$$

which has the same structure as equation (4) of [6].

III. SOLVING FOR L AND R

In the following, an iterative algorithm is outlined for determining the blocks of the block-diagonal matrices $\mathbf{L}$ and $\mathbf{R}$. It is noted that, if the singular vectors obtained in the column-/row-block analysis outlined above are used to form $\mathbf{L}$ and $\mathbf{R}$, then (2) holds; this is the initialization used in the following algorithm. It is noted that this initialization for $\mathbf{L}$ and $\mathbf{R}$ does not provide an effective compression of the FMM for the following reasons.

First, if the matrices $\mathbf{L}\mathbf{T}_{SI}\,\mathbf{R}$ are multiplied together in (2), then the redundancy of $\mathbf{T}_{SI}$ is lost (the individual translators that constitute the global translation matrix $\mathbf{T}$ are each unique, whereas $\mathbf{T}_{SI}$ is comprised of at most 316 unique submatrices). Second, if the representation on the right side of (2) is used without multiplying the matrices together, then the cost to apply $\mathbf{T}_{SI}$ to a vector during an iterative solve is not reduced relative to a standard FMM. The purpose of the algorithm outlined below is to find alternative diagonal blocks for $\mathbf{L}$ and $\mathbf{R}$ that utilize only a fraction of the FMM DOF space, thus reducing the cost to apply $\mathbf{T}_{SI}$ without increasing the cost to store $\mathbf{T}_{SI}$. (In this paper, the terms FMM DOF and DOF are used to indicate the dimension of the FMM translator blocks and the corresponding dimensions of the diagonal blocks in $\mathbf{L}$ and $\mathbf{R}$. This terminology differs from that used in [7].)

The algorithm used to compute the diagonal blocks of $\mathbf{L}$ and $\mathbf{R}$ is summarized in Fig. 1, which is a reverse-bootstrapping procedure This is a modified version of the bootstrapping algorithm reported in [5, 6]. The algorithm begins with the initialization summarized above.

In Fig. 1, matrices $\mathbf{L}_g$ and $\mathbf{R}_g$ are the diagonal blocks of $\mathbf{L}$ and $\mathbf{R}$, and subscript g implies a loop over the groups at this level of the octree. The integer m is the size of each block of $\mathbf{T}_{SI}$ and is equal to the number of columns/rows in each diagonal block of $\mathbf{L}$ and $\mathbf{R}$. The decrement d is the size of the reduction in the number of FMM DOF to be tested in the current iteration of the WHILE loop.

It has been observed that the compression converges to nearly identical values of m for arbitrarily large values of d for cases tested. This property makes the algorithm of Fig. 1 significantly more efficient than the bootstrapping algorithms previously reported for compressing H^2 representations [5, 6], which required d=1. The difference between those methods and the current application to an FMM is likely due to the fact that, in developing a translationally invariant representation from an existing H^2, one starts with an inaccurate representation that is iteratively improved by adding additional DOF. In contrast, when compressing an existing FMM, one starts with an accurate representation, that is compressed by removing DOF.

- n=# DOF in FMM basis; err=0; d = round(n/4)
- while $err < \tau$
  1. $m = n - d$
  2. for k = 1:n_steps (n_steps=2 here)
     a. fix $\mathbf{L}$ and compute a least squares (LS) solution for $\mathbf{R}_g$ using only the first m DOF
        - when k=1, this reduces the number of rows in the $\mathbf{R}_g$ to m
        - when k>1, the $\mathbf{R}_g$ do not change in size
     b. fix $\mathbf{R}$ and compute a LS solution for $\mathbf{L}_g$
        - when k=1, this reduces the number of columns in the $\mathbf{L}_g$ to m
        - when k>1, the $\mathbf{L}_g$ do not change in size
  3. $err\_new = \| \mathbf{T} - \mathbf{L}\,\mathbf{T}_{\text{SI}}\mathbf{R} \| / \|\mathbf{T}\|$
  4. If ($err\_new < \tau$), $n = m$; $err = err\_new$; else,
     - if d>1, m=n; d = round(d/2)
     - else, m=n; end while

Fig. 1. Algorithm for compressing an existing FMM.

IV. EXAMPLES

The following examples compress the black-box FMM (bbFMM) [2] representation of the static Green's function, $G(\vec{r},\vec{r}') = 1/|\vec{r}-\vec{r}'|$. Due to the scale-invariance of the homogeneous kernel, the total number of unique translation matrices needed in the bbFMM representation of G across all levels of the octree is 316.

The bbFMM representation of G is constructed with an accuracy of 1e-6. The Chebyshev order and SVD truncation tolerance [2] of the underlying bbFMM required to obtain 1e-6 accuracy were determined as follows. A dense (i.e., no empty groups), five-level octree was formed, and source and observer points were densely distributed over the surfaces of all octree boxes at the finest level of the tree. The relative RMS error in the bbFMM representation (versus the exact kernel evaluations) was computed for all interactions between non-touching groups at the finest level. Table 1 shows the resulting error as a function of the SVD truncation tolerance when a 729-point bbFMM was used (9 points in the x-, y-, and z-directions for each group). An RMS error less than 1e-6 is obtained when the SVD truncation tolerance is 1e-7, resulting in approximately 157 FMM DOF (the size of each translation matrix is 157-by-157). This is the base FMM representation used in the following examples.

Table 1: Number of FMM DOF remaining after the global SVD compression step outlined in [2] as a function of the SVD tolerance. A 729-point grid of Chebyshev nodes is used, so that the total number of FMM DOF prior to the SVD compression step is 729. Lower order Chebyshev grids fail to provide sufficient accuracy, and higher order grids did not reduce the number of FMM DOF required to achieve 1e-6 accuracy below 157.

SVD Tolerance	# DOF Before SVD	# DOF After SVD	Base-10 log of Error
1e-4	729	44	5.0e-4
1e-5	729	73	9.6e-5
1e-6	729	109	9.6e-6
1e-7	729	157	8.0e-7

A. Example: Points on a line

To illustrate the performance of the compression algorithm, we first consider the static kernel $G(\vec{r},\vec{r}')$ when N=50000 points are distributed along the line defined by x=0, y=0, -1<z<1. A six-level octree is used to decompose the problem, with level-1 being the root box. There are 8, 16, 32, and 64 non-empty groups at levels 3, 4, 5, and 6 of the octree for this geometry.

After the bbFMM representation described above is built, at each level the matrix **T** is constructed as discussed above, and the diagonal blocks of **L** and **R** in (2) are found using the algorithm summarized in Fig. 1. When the iteration completes, the number of remaining FMM DOF at a given level is m. The values of m at each level of the octree for this example are reported in the third column of Table 2. The last column of the table indicates the savings in the floating point costs required to apply the compressed translation matrix $\mathbf{T}_{SI}$ to a vector. The resulting block-diagonal matrices **L** and **R** of (2) are multiplied into the block-diagonal **U** and $\mathbf{V}^h$ matrices shown in (1), effectively compressing the basis matrices (in addition to the translator matrix) at each level.

Table 2: Results of compressing the 1e-6 bbFMM for points on a line. The initial number of FMM DOF at all levels is 157. The last column indicates the reduction in floating-point operations to apply that level's translation matrix to a vector

Octree Level	# FMM DOF Before Compression (n)	# FMM DOF After Compression (m)	FP Savings Factor $(n/m)^2$
3	157	7	503
4	157	17	85
5	157	21	56
6	157	23	47

Finally, it is noted that the relative RMS error in the compressed FMM representation at each level (and globally) is less than 1e-6 (see Step 3 of the algorithm reported in Fig. 1). This ensures that the accuracy of the original FMM representation is retained by the compressed FMM representation.

B. Example: Points on a plane

Table 3 reports the result of applying the FMM compression of Fig. 1 to the static kernel for the case of source/observer points distributed on the square surface shown in Fig. 2. For this geometry, each group has a larger interaction list than for the line example, and more FMM DOF are needed to retain an accuracy of 1e-6. This yields a savings of slightly more than 4.5 in the floating-

Table 3: Results of compressing the 1e-6 bbFMM for the square surface geometry illustrated in Fig. 2. The initial number of FMM DOF at all levels is 157. The last column indicates the reduction in floating-point operations to apply that level's translation matrix to a vector

Octree Level	# FMM DOF Before Compression (n)	# FMM DOF After Compression (m)	FP Savings Factor $(n/m)^2$
3	157	47	11
4	157	71	4.9
5	157	74	4.6
6	157	74	4.6

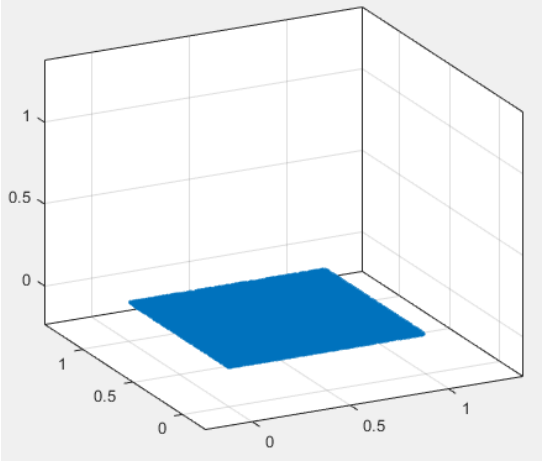


Fig. 2. Surface point distribution (N=1e5 points).

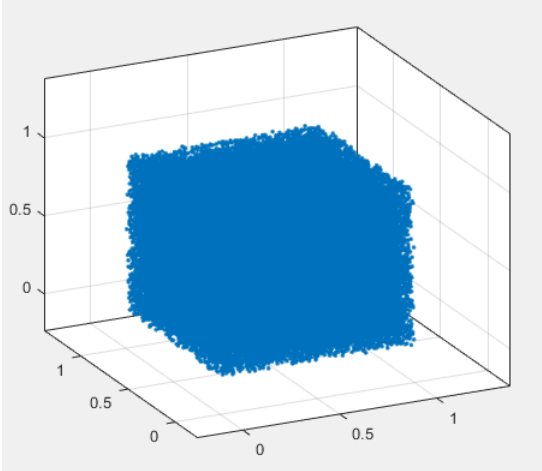


Fig. 3. Volumetric point distribution (N=1e5 points).

point cost to apply the translation matrix to a vector (relative to the original, 1e-6 bbFMM representation).

C. Example: Points in a volume

Table 4 reports the result of applying the FMM compression of Fig. 1 to the static kernel for the case of source/observer points distributed within the cubic volume shown in Fig. 3. For this geometry, each group has a larger interaction list than in the previous two examples, and more FMM DOF are needed to retain an accuracy of 1e-6. For this reason, the floating-point savings provided by the FMM compression is limited to a factor of approximately 2, as indicated by the last column of the table.

D. Example: Coil lattice

Finally, consider the 4-by-4 coil lattice geometry shown in Fig. 4. There are a total of N=80000 point sources/observers in this geometry, and the static kernel, $G(\vec{r},\vec{r}')$, is compressed using a six-level octree with results shown in Table 5.

Table 4: Results of compressing the 1e-6 bbFMM for the volumetric point distribution illustrated in Fig. 2. The initial number of FMM DOF at all levels is 157. The last column indicates the reduction in floating-point operations to apply that level's translation matrix to a vector

Octree Level	**# FMM DOF Before Compression (n)**	**# FMM DOF After Compression (m)**	**FP Savings Factor $(n/m)^2$**
3	157	91	3.0
4	157	108	2.1
5	157	109	2.1

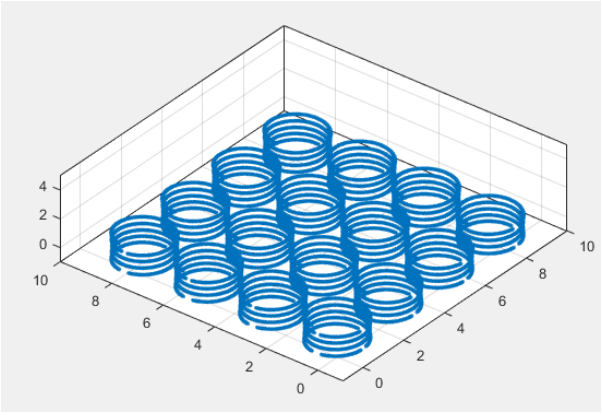


Fig. 4. Coil lattice geometry (N=8e4 points).

Table 5: Results of compressing the 1e-6 bbFMM for the coil lattice geometry illustrated in Fig. 4. The initial number of FMM DOF at all levels is 157. The last column indicates the reduction in floating-point operations to apply that level's translation matrix to a vector

Octree Level	**# DOF Before Compression (n)**	**# DOF After Compression (m)**	**FP Savings Factor $(n/m)^2$**
3	157	52	9.1
4	157	76	4.3
5	157	64	6.0
6	157	58	7.3

V. CONCLUSION

A method for compressing an existing FMM representation of point-to-point interactions for homogeneous kernels in three dimensions has been reported.

Previously reported methods for compressing an existing H^2 representation [5, 6] relied on a bootstrapping method, which caused them to be computationally inefficient. The method reported here is a modified form of [5, 6] and utilizes a reverse-bootstrapping algorithm. The compression has been observed to be insensitive to the reverse-bootstrapping step size that is used to reduce the FMM DOF space. This insensitivity to step-size renders the proposed method computationally efficient for practical 3-D applications.

The FMM compression algorithm has been applied to the bbFMM representation of $G(\vec{r},\vec{r}') = 1/|\vec{r}-\vec{r}'|$ for four different 3-D point distribution examples consisting of points on a line, a surface, a volume and a coil lattice. Significant compression is observed in several cases, with the volumetric point distribution resulting in the least amount of compression relative to the original bbFMM representation. This result is expected, since the bbFMM is already optimized for volumetric point distributions.

Finally, we briefly consider the application of the compression algorithm of Fig. 1 to more complex point distributions, such as that shown in Fig. 5. Unlike the other examples considered above, the point distribution in Fig. 5 has different types of point distributions in different spatial regions. In one corner of the domain, points are distributed along three intersecting lines; in another corner of the domain, points are densely distributed throughout a cubic sub-volume. Assuming that the point distributions are dense, an application of the algorithm in Fig. 1 to this problem at fine levels of the octree can be expected to yield compression results more similar to those shown in Table 4 than those shown in Table 2. This is because the number of FMM DOF, m, retained by the compression algorithm of Fig. 1 is the same for all groups at a given level, and m must be sufficiently large to represent the interactions between groups having dense interaction lists (i.e., octree groups located in the region containing the volumetric point distribution).

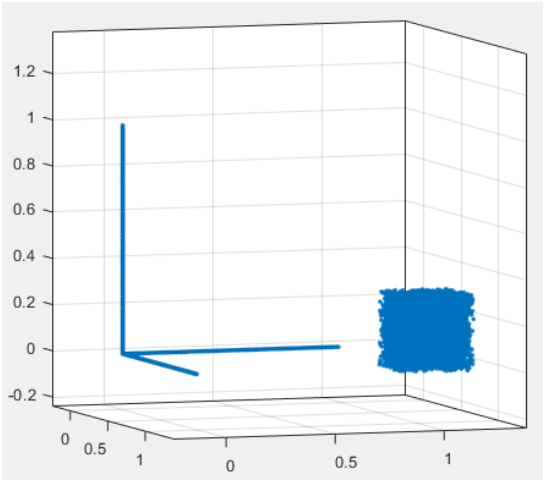


Fig. 5. Example of a mixed DOF distribution containing a region with a dense set of volumetric points and a region with three intersecting lines of points.

However, it has been observed that significant additional compression can be obtained for problems involving non-uniform point distributions (such as Fig. 5) through a straightforward extension the algorithm reported in Fig. 1. The extension is achieved by allowing the number of FMM DOF used for each source/field group at a given level of the octree to vary independently of one another, which leads to non-square translation matrices while retaining translational redundancy. This extended version of the algorithm will be reported separately elsewhere.

ACKNOWLEDGMENT

This work was supported in part by Office of Naval Research Grants N00014-16-1-3066 and N00014-21-1-2599.

REFERENCES

[1] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, no. 2, pp. 325-348, 1987.

[2] W. Fong and E. Darve, "The black-box fast multipole method," *J. Comput. Phys.*, vol. 228, no. 23, pp. 8712-8725, Dec. 2009.

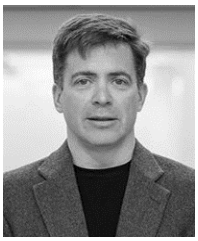
[3] W. Hackbusch, *Hierarchical Matrices: Algorithms and Analysis* (Springer Series in Computational Mathematics). Berlin: Springer-Verlag, p. 511, 2015.

[4] X. Xu and R. J. Adams, "Sparse matrix factorization using overlapped localizing LOGOS modes on a shifted grid," *IEEE T. Antenn. Propag.*, vol. 60, no. 3, pp. 1414-1424, 2012.

[5] R. J. Adams, J. C. Young, and S. D. Gedney, "Compressing H2 matrices for translationally invariant kernels," *Applied Computational Electromagnetics Society (ACES) Journal*, vol. 35, no. 11, pp. 1392-1393, Nov. 2020.

[6] R. J. Adams, J. C. Young, and S. D. Gedney, "Efficiency improvements in compressing H2 matrices for translationally invariant kernels," presented at the *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, Denver, CO, July 10-15, 2022.

[7] R. J. Adams, J. C. Young, and S. D. Gedney, "Compressing a fast multipole method representation of an integral equation matrix," presented at the 2023 *International Applied Computational Electromagnetics Society Symposium*, Monterey/Seaside, CA, 2023.



Robert J. Adams received the B.S. degree from Michigan Technological University, Houghton, MI, USA, in 1993, and the M.S. and Ph.D. degrees in electrical engineering from Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, USA, in 1995 and 1998, respectively.

From 1999 to 2000, he was a Research Assistant Professor with Virginia Tech. Dr. Adams joined the University of Kentucky in 2001, where he is currently a Professor with the Department of Electrical and Computer Engineering.

Dr. Adams has made novel contributions to mesh and frequency stable integral equation formulations of electromagnetic problems, constraint-based methods for high-order MOM discretizations, spectral splitting methods for implementing shadowing effects in integral equations at high frequencies, and sparse direct solution methods for low-to-moderate frequency electromagnetics applications. Dr. Adams' group developed the first $O(N)$ sparse direct solver for 3-D H^2 and FMM representations of static and low-frequency EM problems. Dr. Adams is a senior member of the IEEE.

John C. Young received the B.E.E. degree in electrical engineering from Auburn University in 1997, the M.S. degree in electrical engineering from Clemson University in 2000, and the Ph.D. degree in electrical engineering also from Clemson University in 2002.

From January 2003 to April 2003, he served as a post-doctoral researcher at Clemson University, and from 2003 to 2005, he served as a post-doctoral researcher at Tokyo Institute of Technology, Tokyo, Japan. From 2005 to 2008, he worked at Japan Radio Co. Since 2008, he has been with the Department of Electrical and Computer Engineering at the University of Kentucky, Lexington, KY, where he is currently an associate professor.

Dr. Young's research interests include integral equation methods, finite element methods, electromagnetic theory, waveguides, array antennas, and magnetic signature modeling of hysteretic materials. He is a member of IEEE, the Applied Electromagnetics Society (ACES), and URSI Commission B. He currently serves as an Associate Editor for the IEEE Transactions on Antennas and Propagation and on the Education Committee of the Antennas and Propagation Society. He also served (2020-2023) on the Board of Directors of ACES where he is currently Secretary.

Stephen D. Gedney received the B.Eng.-Honors degree from McGill University, Montreal, P.Q., in 1985, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Illinois, Urbana-Champaign, IL, in 1987 and 1991, respectively.

He is currently the Don and Karen White Professor of the Department of Electrical Engineering at the University of Colorado Denver (CUD). Previously he was a Professor of Electrical Engineering at the University of Kentucky from 1991-2014. He worked for the U.S. Army Corps of Engineers, Champaign, IL (1985-1987). He was a visiting Professor at the Jet Propulsion Laboratory, (1992-1993), HRL laboratories (1996-1997) and Alpha Omega Electromagnetics (2004-2005). He received the Tau Beta Pi Outstanding Teacher Award in 1995 and 2013. From 2002-2014, he was the Reese Terry Professor of Electrical and Computer Engineering at the University of Kentucky. He was titled as a Distinguished Professor of the University of Colorado in 2022. He is a past Associate Editor of the IEEE Transactions on Antennas and Propagation (1997-2004), a member of the IEEE Antennas and Propagation Society ADCOM (2000-2003), and served as the chair of the IEEE Antennas and Propagation Society Membership Committee (1995-2002). He is a Fellow of the IEEE and member of Tau Beta Pi.