Compressing H^2 Matrices for Translationally Invariant Kernels

R. J. Adams and J. C. Young
Electrical & Computer Engineering
University of Kentucky
Lexington, KY, USA
rjadams@uky.edu, john.c.young@uky.edu

S. D. Gedney
Electrical Engineering
University of Colorado Denver
Denver, CO, USA
stephen.gedney@ucdenver.edu

***Abstract*—H^2 matrices provide compressed representations of the matrices obtained when discretizing surface and volume integral equations. The memory costs associated with storing H^2 matrices for static and low-frequency applications are $O(N)$. However, when the H^2 representation is constructed using sparse samples of the underlying matrix, the translation matrices in the H^2 representation do not preserve any translational invariance present in the underlying kernel. In some cases, this can result in an H^2 representation with relatively large memory requirements. This paper outlines a method to compress an existing H^2 matrix by constructing a translationally invariant H^2 matrix from it. Numerical examples demonstrate that the resulting representation can provide significant memory savings.**

***Keywords*—integral equations, sparse matrices.**

I. INTRODUCTION

Integral equations provide a useful method for formulating both linear and nonlinear electromagnetic problems. In most cases, discretized integral equations yield matrix equations having the following form,

$$\mathbf{Z}\mathbf{x} = \mathbf{f}\,, \tag{1}$$

where $\mathbf{Z}$ is the N-by-N system matrix, $\mathbf{f}$ is the N-by-1 source vector, and $\mathbf{x}$ is the vector of unknowns. The elements of $\mathbf{Z}$ are obtained from samples of an underlying kernel. Although the kernel, denoted G below, is often translationally invariant, the geometry usually is not. Consequently, the matrix $\mathbf{Z}$ seldom exhibits translational invariance.

The matrix in (1) is dense, and it is necessary to use compressed representations for large simulations. H^2 matrices provide one such representation [1, 2]. In this paper, the specific H^2 representation described in an appendix of [3] is used (the representation described in [3] was therein referred to as an MLSSM representation; it has since been determined that the MLSSM is equivalent to an H^2 matrix). In an H^2 representation, $\mathbf{Z}$ is decomposed using an octree as $\mathbf{Z} = \mathbf{Z}_{\text{near}} + \mathbf{Z}_{\text{far}}$, where $\mathbf{Z}_{\text{near}}$ contains only those source/field interactions that occur between touching groups at the finest level of the tree. All other interactions are included in $\mathbf{Z}_{\text{far}}$.

The matrix $\mathbf{Z}_{\text{far}}$ is decomposed by levels in the L-level octree (here, l=1 is the root level),

$$\mathbf{Z}_{\text{far}} = \mathbf{Z}_{\text{far}}^{(L)} + \mathbf{Z}_{\text{far}}^{(L-1)} + \cdots + \mathbf{Z}_{\text{far}}^{(3)}\,. \tag{2}$$

This work was supported in part by Office of Naval Research Grant N00014-16-1-3066 and NASA Grant NNX16AO88G.

Submitted On: September 8, 2020
Accepted On: September 9, 2020

$\mathbf{Z}_{\text{far}}^{(l)}$ is the portion of $\mathbf{Z}_{\text{far}}$ corresponding to interactions between non-touching groups at level-l of the tree that are not represented at a parent level.

Each matrix on the right side of (2) has the following form,

$$\mathbf{Z}_{\text{far}}^{(l)} = \mathbf{U}\mathbf{T}\mathbf{V}^h\,, \tag{3}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal block-diagonal matrices that map from the original sources/observers in each level-l group to the minimum source/field interaction bases used by each level-l group. As in the fast multipole method (FMM), $\mathbf{U}$ and $\mathbf{V}$ are themselves a multilevel product of change-of-basis operators from the finest level of the octree to the basis used at level-l. This detail is omitted since only $\mathbf{T}$ is considered in the following.

As discussed in [3], the H^2 representation of (1) is obtained from sparse samples of $\mathbf{Z}$. Thus, $\mathbf{T}$ in (3) does not exhibit any translational invariance possessed by the kernel. This is in contrast to an FMM, which preserves a kernel's translational invariance. For this reason, the storage requirements associated with an H^2 representation can be larger than those required by an equivalent FMM. The remainder of this paper outlines a method for compressing $\mathbf{T}$ by developing a translationally invariant H^2 representation for translationally invariant kernels.

II. TRANSLATIONALY INVARIANT H^2 MATRICES

At each level, given $\mathbf{T}$, we seek an alternative representation with the form,

$$\mathbf{T} = \mathbf{L}\mathbf{T}_{\text{SI}}\mathbf{R}^h\,. \tag{4}$$

The following structure is imposed on the matrices on the right-side of (4). $\mathbf{L}$ and $\mathbf{R}$ are non-square, block-diagonal matrices, with one diagonal block per level-l group. The number of rows in each diagonal block of $\mathbf{L}$ and $\mathbf{R}$ must match the corresponding column dimensions of the blocks of $\mathbf{U}$ and $\mathbf{V}$; this value is denoted as n_g in the following. The number of columns in each block of $\mathbf{L}$ and $\mathbf{R}$ is discussed below. The translation matrix $\mathbf{T}_{\text{SI}}$ in (4) has the same structure as $\mathbf{T}$; $\mathbf{T}_{\text{SI}}$ has a nonzero interaction block (specified below) for each nonzero interaction block in $\mathbf{T}$, and the other blocks are zero. Unlike $\mathbf{T}$, $\mathbf{T}_{\text{SI}}$ is constructed to preserve any translational invariance present in the underlying kernel.

The procedure used to construct the representation on the right side of (4) consists in first specifying $\mathbf{T}_{\text{SI}}$, and then solving for the diagonal blocks of $\mathbf{L}$ and $\mathbf{R}$.

https://doi.org/10.47037/2020.ACES.J.351165

A. Constructing $\mathbf{T}_{SI}$

At level-l of the octree, $\mathbf{T}_{\text{SI}}$ is obtained as follows. A set of m random points, $\{\vec{\mathbf{r}}_i\}_{i=1}^{m}$, is uniformly distributed within a cube that is centered on the origin and has a size equal to a level-l octree box. Shifted copies of these points serve as both source and observer points within each non-empty group at this level; the shift is equal to the translation vector from the origin to the center $\vec{\mathbf{r}}_c$ of each non-empty octree box, $\{\vec{\mathbf{r}}_i - \vec{\mathbf{r}}_c\}_{i=1}^{m}$. If there are M groups, this procedure yields a square matrix, $\mathbf{T}_{\text{SI}}$, that preserves a kernel's translational invariance. The size of $\mathbf{T}_{\text{SI}}$ is (mM)-by-(mM). The value of m is determined dynamically, as discussed in the next section.

B. Solving for $\mathbf{L}$ and $\mathbf{R}$

The diagonal blocks of $\mathbf{L}$ and $\mathbf{R}$, as well as the value of m, are obtained using the following "bootstrapping" procedure, which is presented using Matlab notation. In this procedure, $\mathbf{L}_{\text{g}}$ and $\mathbf{R}_{\text{g}}$ denote the diagonal blocks of $\mathbf{L}$ and $\mathbf{R}$, the subscript "g" implies a loop over the level-l groups, and "LS" denotes least-squares.

- m=0; err=1; $\mathbf{R}_{\text{g}}$ = zeros(n_{g}, 0)
- while $err > tol$
  - $m = m + m_{\text{inc}}$
  - retaining any previously computed portions of each $\mathbf{T}_{\text{SI}}$ block, fill in the new portion (which is due to incrementing m)
  - $\mathbf{L}_g$ = zeros(n_{g}, m); if (m==m_{inc}), $\mathbf{R}_g$ = [$\mathbf{R}_g$, rand(n_{g}, m_{inc})]; else, $\mathbf{R}_g$ = [$\mathbf{R}_g$, zeros(n_{g}, m_{inc})];
  - for k = 1 : n_steps (n_steps=20 here)
    - fix $\mathbf{R}$, and compute a LS solution for $\mathbf{L}_g$
    - fix $\mathbf{L}$, and compute a LS solution for $\mathbf{R}_g$
  - err = $\| \mathbf{T} - \mathbf{L}\,\mathbf{T}_{\text{SI}}\,\mathbf{R}^h \| / \|\mathbf{T}\|$

At the termination of this procedure, the right side of (4) matches $\mathbf{T}$ with relative accuracy tol. For the following examples, tol = 1e-5, which is the same tolerance that will be used to approximate $\mathbf{Z}_{\text{far}}$ in (2) in the following examples.

III. Numerical Examples

The compression scheme (4) is illustrated for the kernel, $G(\mathbf{r}\text{-}\mathbf{r}') = 1/|\mathbf{r}\text{-}\mathbf{r}'|$, and three $\mathbf{Z}$ matrices using this kernel are considered below. In Example-1, $\mathbf{Z}_{\text{far}}$ consists of point-to-point samples of G, corresponding to interactions between a cluster of 2e4 points randomly distributed in a 1-by-1-by-1 meter cube; the matrix $\mathbf{T}$ is constructed at level-3 of the octree using the algorithm outlined in the appendix of [3]. Example-2 is similar to Example-1, except the 2e4 points are randomly located along a 1 meter line; the matrix $\mathbf{T}$ is constructed at level-6 of the tree. Finally, for Example-3, $\mathbf{Z}_{\text{far}}$ is obtained from a locally corrected Nyström discretization of the magnetostatic integral equation for a thin spherical steel shell [4] with N=18738; $\mathbf{T}$ is obtained at level-3 of the octree and is shown in Fig. 1.

Table I reports the compression ratio, ρ, obtained using the right side of (4) instead of $\mathbf{T}$. In all three examples, the matrix $\mathbf{T}_{\text{SI}}$ is constructed using only point samples of the scalar kernel, G. The table also reports the number of nonempty groups, M, as well as m, the number of random points used in each nonempty level-l group (l=3 for Example-1 and Example-3, l=6 for Example-2). The final value of m is different for each geometry.

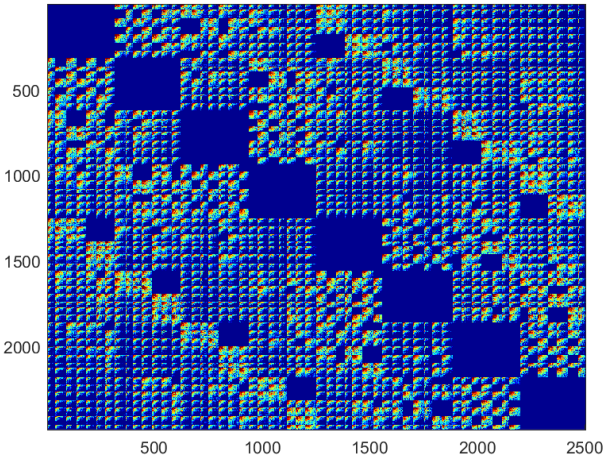

Fig. 1. Absolute value of the elements of $\mathbf{T}$ for Example-3 (log scale, dark blue colors indicate zeros). There are 56 nonempty level-3 groups; 56 column blocks and 56 row blocks are seen in the image.

TABLE I. Compression Ratios for the Three Examples

	M	m	ρ
Example-1	64	56	5.95
Example-2	32	12	1.01
Example-3	56	75	4.21

IV. Summary

A method to compress the $\mathbf{T}$ matrix of an H^2 representation has been reported. The method works directly with $\mathbf{T}$ and samples of the underlying scalar kernel, G. It has been shown that the method provides significant memory savings in some cases. Although not shown here due to space constraints, it has also been observed that the proposed method provides super-algebraic convergence with increasing m.

References

[1] S. Börm, "Construction of data-sparse H2-matrices by hierarchical compression," SIAM Journal on Scientific Computing, vol. 31, no. 3, pp. 1820-1839, 2009.

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

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

[4] J. C. Young and S. D. Gedney, "A Locally Corrected Nystrom Formulation for the Magnetostatic Volume Integral Equation," IEEE Transactions on Magnetics, vol. 47, no. 9, pp. 2163-2170, Sep. 2011.