

NASA/CR-2008-215353



Direct Solve of Electrically Large Integral Equations for Problem Sizes to 1M Unknowns

John Shaeffer
Marietta, Georgia

September 2008

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Phone the NASA STI Help Desk at (301) 621-0390
- Write to:
NASA STI Help Desk
NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320

NASA/CR-2008-215353



Direct Solve of Electrically Large Integral Equations for Problem Sizes to 1M Unknowns

John Shaeffer
Marietta, Georgia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-02117

September 2008

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information (CASI)
7115 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Direct Solve of Electrically Large Integral Equations for Problem Sizes to 1M Unknowns*

John Shaeffer
3278 Hunterdon Way
Marietta, Georgia 30067
john@shaeffer.com

Abstract - To solve extremely large problem sizes in electromagnetics, on the order of millions of unknowns, a novel matrix compression technique, Adaptive Cross Approximation (ACA), is applied to the Method of Moments (MoM). To implement ACA, several major enhancements to the MoM were required: spatial grouping of unknowns, ACA compression of the impedance matrix, Lower/Upper (LU) compressed factorization factors and ACA compression of the right hand sides.

When unknowns for electrically large bodies are **spatially grouped** into local block regions, the blocks of the Method of Moments (MOM) \mathbf{Z} matrix and its Lower/Upper (\mathbf{LU}) factors are comprised of block sub matrices which, except for the diagonal blocks, can be well approximated by low rank matrices. These low rank approximates can then be computed using the Adaptive Cross Approximation (ACA), a technique which has very significant reduction in memory and operations count. And for monostatic scattering of the above system, where there are many right hand sides, the blocked right hand side \mathbf{V} and current solution \mathbf{J} can also be expressed in terms of low rank approximations. This report describes a novel approach in utilizing the Adaptive Cross Approximation technique to factor the \mathbf{Z} matrix, $\mathbf{Z} = \mathbf{LU}$ and to back-solve the blocked MOM system equation for a frequency domain electric field integral equation for 3D PEC surfaces. The integral equation uses standard Rao Wilton Glisson basis and test functions so that the resulting matrix is symmetric.

Compressible matrices and their low rank approximations fundamentally mean that most of the blocked MOM system matrix equation elements, before compression, contain very little physical information. The ACA can be used to extract and compress the system equation elements, keeping only the necessary physics. The ACA can be used for all steps of the solution: filling the \mathbf{Z} matrix; \mathbf{LU} factoring; and \mathbf{LU} solve. This results, depending on problem size, in *orders of magnitude* reduction of memory and run times and means that one can use inexpensive computer resources for problems that previously took super computers or could not be solved at all.

* NASA funded validation and problem size extensions, earlier portions funded by Mrs. Shaeffer and indirectly by the Office of Naval Research.

It is also important to point out that this ACA LU factorization and solution is completely within the context of standard Method of Moments. There are no requirements for new basis or test functions, no requirements for far or near field distinction, no requirements for auxiliary “equivalent” sources on a grid, and no requirement for analytic Green’s function expansion.

I. Introduction

The solution to Maxwell's frequency domain equations in integral form using the electric field integral equations (EFIE), magnetic field integral equations (MFIE), or combined field integral equations (CFIE) is very well established using the Method Of Moments (MOM) matrix formulation approach. In the MOM, unknowns and test functions are often sub-sectional sampled on a sub wavelength scale resulting in dense matrices of size N , the number of unknown current coefficients. This fact has been the limiting feature in applying MOM to electrically large bodies due to the tyranny of N^2 for matrix fill time and storage and of N^3 time for LU factorization, that is the factorization of \mathbf{Z} into a product of a Lower triangle matrix times an Upper triangle matrix. Computational time for LU factorization thus grows as the sixth or ninth power of body size for surface or volume problems respectively. Solve cost varies as N^2 time for *each* right hand side and becomes a particularly severe issue for electrically large scatterers as one needs ever-increasing backscatter angles to adequately sample the scattering pattern.

Researchers have made various approaches at solving this electrically large body problem using so-called "fast" methods. These have been mostly based on the fact that when unknowns N are grouped in local spatial regions, the resulting off-diagonal blocks of the system impedance matrix \mathbf{Z} can be approximated by low rank matrices. An alternate view is that these matrix blocks from the MOM formulation, where sampling rate requirements are based for near-near interactions, are extremely over-specified in terms of the number of matrix elements per block required for distant interactions. Matrix compression¹ is one way to significantly reduce the number of elements required to represent the physics contained in non-self block matrices. Advantage is then taken of this feature, in a variety of approaches, to speed the solution. A good review of existing "fast" methods for electromagnetic surface integral equations such as FMM, AIM, pFFT, MLFMM, IES3 has been presented by Zhao, Vouvakis and Lee [1].

References [1-5] have used various approaches to compute low rank approximates of the blocked system \mathbf{Z} matrix. Each reduced matrix fill time and storage. Iterative solvers were then used to compute current solutions for each right hand side excitation vector.

Iterative solvers may be quite satisfactory for only a few right hand sides such as antenna or bistatic scattering problems, but for monostatic scattering with many required sampling angles, this part of the problem becomes expensive; and there are often convergence issues with such solvers, particularly with geometries with many mutual interactions such as cavities.

Another "fast" solution is the Simply Sparse approach [7-9], which takes a change of basis approach. The MOM basis and test functions are transformed using unitary orthogonal matrices. This results in only a small fraction of the unknowns radiating to the far field. Resulting blocks of the system matrix are upper corner dense. This property is then used to efficiently LU factor and solve the system equations, [10].

¹ Compressible means that these matrices can be well approximated by low rank matrices.

The problem all of these “fast” methods have overcome, each in their own way, is that of reducing time and memory storage required to fill blocks of the \mathbf{Z} matrix. Bebendorf [2] has recently introduced the Adaptive Cross Approximation (ACA) that attacks this problem in a rather interesting and purely matrix algebra approach. His ACA algorithm computes compressed forms of low rank block matrices using only a few rows and columns of that matrix. This technique has very significant ramifications: a) the matrix block does *not* require *full* computation or fill. Only selected rows and columns are needed; b) the resultant matrix can be stored in compressed format significantly reducing memory storage; c) use of compressed matrices in computations very significantly reduces operations count; and d) the operations count required by the ACA algorithm is small.

A number of authors [1-5, 9] have now reported use of this ACA technique for filling system matrix blocks, including this author. It was during the work reported in [9], that it was observed that when the unknowns are spatially grouped, not only were the off-diagonal blocks of \mathbf{Z} compressible, but also the off-diagonal blocks of the \mathbf{LU} factored matrix were of compressible. This later observation then raised the question: Could the ACA be used to directly \mathbf{LU} factor and solve the system equations without explicit matrix fill? That question is the topic of this report.

This report starts by positing that when unknowns are spatially grouped, the off-diagonal blocks of \mathbf{Z} and its \mathbf{LU} factored form are able to be well approximated by low rank matrices expressed in outer product \mathbf{UV} form. In this outer product \mathbf{UV} form, we say the matrix block is compressed. (The diagonal blocks of the matrix are not compressible.) This means that such matrix blocks can be expressed as the outer product of a column dominant matrix \mathbf{U} times a row dominant² matrix \mathbf{V} , e.g., if \mathbf{A} is an $m \times n$ matrix and can be represented by a low rank matrix, i.e., \mathbf{A} is compressible, \mathbf{A} can be approximated by the outer product of \mathbf{Au} \mathbf{Av} where the column dominant matrix \mathbf{Au} is $m \times k$ and \mathbf{Av} is the row dominant matrix $k \times n$. Low rank means $k \ll m, n$ thus the storage requirement for \mathbf{Au} and \mathbf{Av} is very much less than the storage requirement for \mathbf{A} . And, the operations count involving \mathbf{A} in its compressed low rank form \mathbf{AuAv} can be very much less than using \mathbf{A} in full form.

Thus, spatially grouped off-diagonal blocks of the system impedance matrix \mathbf{Z} can be expressed in the outer product form $\mathbf{Z} = \mathbf{Zu Zv}$ and its \mathbf{LU} factors can be similarly expressed as $\mathbf{Lu Lv}$ and $\mathbf{Uu Uv}$. Further, for monostatic scattering, where there are many RHS incident plane wave-forcing functions, blocks of \mathbf{V} can be compressed as $\mathbf{Vv Vu}$ and the final current solution blocks \mathbf{J} are also can be compressed as $\mathbf{Ju Jv}$.

In the work that follows we will show results for the frequency domain 3D EFIE Surface Integral Equation MOM using RWG triangle basis functions [11, 12] with Galerkin test functions so that a symmetric block system results where the unknowns have been grouped into local spatial regions. The self and near self-terms are computed using the

² A column dominant matrix has fewer columns than rows; a row dominant matrix has fewer rows than columns.

radial-angular approach described in [13]. All computations were done in single precision.

This report is organized as follows:

Section 2 discusses spatial grouping of unknowns, low rank approximations, rank fraction and the singular value decomposition (SVD) character of compressible matrices.

Section 3 presents rank fraction compressibility maps for various parts of the MOM system matrix, and the Z, LU, V, and J matrices for the SLICY and Open Pipe geometries.

Section 4 briefly reviews the ACA method, describes how it can be used for LU factorization, for filling the RHS monostatic voltage matrix, and for the LU forward and back solve. Memory and operations count efficiencies are discussed.

Section 5 presents validation run results against another well-known code for the SLICY geometry.

Section 6 presents a complexity study for the open pipe geometry in terms of matrix fill time, memory, LU factor and RHS solve times.

Section 7 compares run specifics for similar SLICY and open pipe cases.

Appendix presents the formulas for block symmetric LU factorization and solution.

II. Unknown Grouping, Compressible Matrix Blocks and Their Low Rank Approximation

MOM system equations are full rank, but when unknowns are grouped into local spatial regions, the off diagonal block-block or region-region sub matrices are compressible when the regions are spaced some distance apart, see Figure 1. Typical sub-sectional MOM formulations sample the unknown current density at 7 to 20 samples per wavelength (λ) and 50- 400 samples per λ^2 for wire and surface problems respectively. This sampling is required to adequately compute near–near interactions, but is clearly overkill for distant–distant interactions as evidenced by the compressible nature of these block matrices.

Unknowns in this work have been grouped into local regions using a cobblestone distance sorting technique as follows. Create an array of all unsorted spatial locations in 3D space. From this array of spatial points, find the minimum and maximum (x, y, z) points which defines vectors v_{Min} and v_{Max} relative to the origin. Define a reference direction vector $v_{Ref} = v_{Max} - v_{Min}$. Project all unsorted points onto this reference

direction to find the point with the smallest projected distance. Call this spatial location vPt Start. This point is the 1st member of the group. Next compute the distances from vPt Start to all other unsorted points. Order this array by distance from close to far using a sorting algorithm such as quick sort. Fill region with closest unsorted points. Terminate when desired group size is obtained, or if the next candidate group point is further than a specified tolerance. Repeat procedure for next group.

For surface geometries, this procedure produces cobblestone geometric groups, much like a stonemason placing bricks starting from an initial vertex and extending out to arms reach, see Figure 2. Each group is composed of members who are sorted in distance in a smooth regular fashion from near to far. The specified limit distance for a group to accept a new member is that the candidate point must be closer than typically three characteristic dimension lengths. The characteristic dimension length is defined as the average triangle leg length as defined in the geometry mesh.

Once unknowns are grouped, the interaction between a pair of groups becomes a block in the system matrix such that block \mathbf{Z}_{ij} is the interaction between the i^{th} and j^{th} groups. Self-interaction blocks are on the diagonal, near interactions close to the diagonal and distant interactions far from the diagonal of the system matrix.

Matrix Compression and Low Rank Outer Product Approximations: Central questions at this point are three: 1) How can we tell if a matrix block \mathbf{A} is compressible? 2) What is a low rank approximation to \mathbf{A} if \mathbf{A} is compressible? And 3) what is the error in the low rank outer product approximation to $\mathbf{A} \sim \mathbf{A}\mathbf{u}\mathbf{A}\mathbf{v}$?

The starting point for this discussion is to review Singular Value Decomposition (SVD) theory from linear algebra. The results of a SVD show if a matrix is compressible and how a low rank outer product can approximate a compressible matrix. We note that SVD is a backdrop discussion, because in actual practice the Adaptive Cross Approximation will be used to actually compute low rank approximates, for reasons to be evident shortly.

Any matrix can be SVD factored into the product of three matrices [14, 15]. The SVD of complex matrix (not necessarily a square) $\mathbf{A} \in \mathbb{C}^{m \times n}$ is

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^h = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_j & \mathbf{u}_m \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_j & 0 \\ 0 & 0 & \sigma_p \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \cdots & \cdots \\ \mathbf{v}_j & \cdots & \cdots \\ \mathbf{v}_n & \cdots & \cdots \end{bmatrix} \quad (1)$$

where $\mathbf{U} \in \mathbb{C}^{m \times m}$ and $\mathbf{V}^h \in \mathbb{C}^{n \times n}$, and $\mathbf{\Sigma} \in \mathbb{R}^{m \times p}$ is the diagonal matrix of real singular values σ sorted from high to low. The \mathbf{U} and \mathbf{V}^h (the superscript h indicates that this is a Hermitian conjugate matrix) are unitary matrices and the singular value matrix $\mathbf{\Sigma}$ has only real diagonal entries ordered from maximum to minimum

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0 \quad \text{where } p = \min(m, n) \quad (2)$$

The shape (number of rows and columns) of Σ is the same as \mathbf{A} . The unitary matrices \mathbf{U} and \mathbf{V}^h are square of size $m \times m$ and $n \times n$ respectively. The column vectors \mathbf{u}_j are the columns of \mathbf{U} and the row vectors \mathbf{v}_j are the rows of \mathbf{V}^h . The \mathbf{u}_j 's are not only orthogonal, but also orthonormal to each other, similarly with the \mathbf{v}_j 's. The columns of \mathbf{U} and rows of \mathbf{V} form a complex basis set of vectors in multi-dimensional space in the multi-dimensional spaces $C^{m \times 1}$ and $C^{1 \times n}$, respectively.

The individual diagonal elements of the singular value matrix Σ , σ_j , are the singular values of matrix \mathbf{A} . The rank of \mathbf{A} is r , the number of 'non-zero' singular values.

The SVD of \mathbf{A} can also be written as the sum of r rank-one outer product matrices due to the properties of Σ [14, 15]

$$\mathbf{A} = \sum_{j=1}^r \sigma_j \mathbf{u}_j \mathbf{v}_j = \sum_{j=1}^r \sigma_j \begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \end{bmatrix} \quad (3)$$

where \mathbf{u}_j and \mathbf{v}_j are the individual columns and rows of \mathbf{U} and \mathbf{V}^h respectively and the sum is over the r non zero singular values.

If the above summation is truncated to index v with $v < r$, then this v^{th} partial sum captures as much of the energy of \mathbf{A} as is possible for any approximation of rank v with "energy" defined by either the 2-norm or the Frobenius norm [15].

Now we have to look at this from a practical computational perspective where we have finite computer arithmetic and finite and/or desired precision in computations of the various aspects of our problem.

The rank of \mathbf{A} is r , the number of 'non-zero' singular values [15]. From a practical perspective, Trefethen and Bau [15] also define the 'numerical rank' of \mathbf{A} as the number of singular values greater than some judiciously chosen value or tolerance. The chosen tolerance is, of course, up to the user and the problem at hand. Consequently 'numerical rank' and tolerance are related.

Matrix \mathbf{A} is compressible if its singular values decrease rapidly. In this case where the singular values drops to levels close to 32 bit machine precision, the sum of rank one outer products of \mathbf{u} and \mathbf{v} with singular value coefficients σ_j can be truncated at some index value $k < r$ (where r was the number of non zero singular values of \mathbf{A} and is the rank of \mathbf{A}). In this case, we can approximate \mathbf{A} with its truncated SVD over k singular values rather than all r singular values:

$$\begin{aligned}
\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^h &\approx \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_j & \mathbf{u}_k & \mathbf{u}_m \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_j & & \\ & & \sigma_k & \\ & & & \sim 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \dots & \dots & \dots \\ \mathbf{v}_j & \dots & \dots & \dots \\ \mathbf{v}_k & \dots & \dots & \dots \\ \mathbf{v}_n & \dots & \dots & \dots \end{bmatrix} \\
&\approx \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_j & \mathbf{u}_k & 0 \\ \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & 0 \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_j & & \\ & & \sigma_k & \\ & & & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \dots & \dots & \dots \\ \mathbf{v}_j & \dots & \dots & \dots \\ \mathbf{v}_k & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{4}$$

If the singular values drop rapidly, then k is much smaller than r , $k \ll r$, and the outer product \mathbf{UV} approximation to \mathbf{A} has low rank.

In the SVD summation form (3) we have the truncated SVD approximation as the first k outer product sums

$$\mathbf{A} \approx \sum_{j=1}^k \sigma_j \mathbf{u}_j \mathbf{v}_j = \sum_{j=1}^k \sigma_j \begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_j & \mathbf{u}_k \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 \mathbf{v}_1 & \dots & \dots & \dots \\ \sigma_j \mathbf{v}_j & \dots & \dots & \dots \\ \sigma_k \mathbf{v}_k & \dots & \dots & \dots \end{bmatrix} \tag{5}$$

In this case, matrix \mathbf{A} with original rank r can be approximated by an outer product of a column dominant matrix times a row dominant matrix. The rank of the approximation is k . This approximation is reasonable provided matrix \mathbf{A} is compressible, i.e., its singular values rapidly decrease in value, $\sigma_r \ll \sigma_k$.

For a typical MOM problem, what is the behavior of the singular values of spatially blocked (grouped) source and test functions? A practical MOM block matrix example is shown in Figure 3 where the ordered singular values are plotted on a *logarithmic* scale. This matrix block is the interaction expressing the electrical influence of a source group of $n = 214$ unknowns on a distant test group of $m = 220$ test functions. Here we see the singular values decrease *exponentially* to near 32-bit machine precision by the 30th singular value out of a possible 214 singular values. This matrix block is thus compressible, $k \ll m$, and can be approximated as the outer product of a $m \times k$ column dominant matrix times a $k \times n$ row dominant matrix where k is at most 30 and perhaps less based on the desired precision for the approximate outer product. Thus we immediately see one rationale for using low rank approximates in our computations. The

memory storage for column like \mathbf{U} and row like \mathbf{V} is potentially very much less than for the original block.

What was the ‘precise’ rank of the original 220 x 214 matrix in the above example, i.e., the number of non-zero singular values of the SVD expansion? Certainly finite computer arithmetic calculations will never produce this answer. But we see that from a practical point of view that this matrix is rank deficient [1], that is out of a possible 220 singular values only about 30 are greater than levels close to 32-bit machine precision.

Thus a matrix is compressible if its singular values drop rapidly in which case the truncated SVD $\mathbf{A} \sim \mathbf{U} \mathbf{\Sigma} \mathbf{V}$ approximation to \mathbf{A} is of low rank.

For a compressible matrix, the SVD computes the *best* low rank approximation to \mathbf{A} [15]. Best meaning for a given error in either the 2-norm or Frobenius norm, the SVD produces the smallest value of k for the number of columns in \mathbf{U} and rows in \mathbf{V} for any choices of k orthonormal columns in \mathbf{U} and k orthonormal rows in \mathbf{V} .

The SVD is, however, not the only tool available to compute a low rank outer product approximation to \mathbf{A} . QR factorization is another approach and the Adaptive Cross Approximation (ACA) is yet another.

The SVD is not a practical approach to compute this approximation for two important reasons: 1) All elements of \mathbf{A} must be known prior to computing its SVD; and 2) The SVD operations count for a square matrix is of order m^3 , i.e., the SVD is an expensive computation. Similar issues exist with QR approaches.

The ACA [2-5] on the other hand is a very practical approach for computing an outer product low rank approximation because only a few rows and columns of \mathbf{A} are needed for the computation and the operations count is much smaller than required by SVD (discussed below).

The next issue concerns the error in the truncated outer product (low rank) approximation for \mathbf{A} denoted by $\mathbf{A}_c(k)$ where

$$\mathbf{A}_c(k) = \mathbf{A} \mathbf{u} \mathbf{A} \mathbf{v} = \begin{bmatrix} \mathbf{A} \mathbf{u} \\ \mathbf{A} \mathbf{v} \end{bmatrix} \begin{bmatrix} \mathbf{A} \mathbf{v} \\ \mathbf{A} \mathbf{u} \end{bmatrix} \cong \mathbf{A} \quad (6)$$

where column like $\mathbf{A} \mathbf{u}$ is $m \times k$ and row like $\mathbf{A} \mathbf{v}$ is $k \times n$ and their product matrix $\mathbf{A} \mathbf{c}$ is the low rank outer product approximation to \mathbf{A} .

The fractional error tolerance ϵ of this approximation is measured in terms of Frobenius matrix norm, namely,

$$\frac{\|\mathbf{A} - \mathbf{A}_c(k)\|}{\|\mathbf{A}\|} \leq \varepsilon \quad (7)$$

where the numerator is the norm of the approximation error.

If matrix $\mathbf{A}_c = \mathbf{A}\mathbf{u}\mathbf{v}$ represents \mathbf{A} to this tolerance, we say that \mathbf{A} can be approximated by a matrix of rank k for tolerance ε . In this sense, rank of the approximate matrix \mathbf{A}_c and tolerance are directly related. We also note that for a given tolerance ε , the rank of the outer product approximate using the SVD will be less than with using the ACA. That is, the SVD computation is the ‘best’ Frobenious low-rank approximate to \mathbf{A} for a given ε [15].

If the ordered singular values decrease rapidly, then \mathbf{A} can be well approximated by an outer product approximation of low rank; and, therefore, the matrix \mathbf{A} is compressible. And lastly, the approximation error is measured using matrix norms.

Rank Fraction³: When evaluating memory storage of compressible block matrices using the ACA UV form, one needs a relative evaluation of the reduced memory storage for the $\mathbf{Z}\mathbf{u}\mathbf{Z}\mathbf{v}$ approximation relative to the memory required to store \mathbf{Z} . The ratio of the storage for $\mathbf{Z}\mathbf{u}$ and $\mathbf{Z}\mathbf{v}$ compared to the storage for \mathbf{Z} is a normalized measure of the compressibility of matrix \mathbf{Z} . In this report, this ratio is called rank fraction and is defined as the memory storage for $\mathbf{Z}\mathbf{u}$ and $\mathbf{Z}\mathbf{v}$, $k(m+n)$, compared to the $m*n$ storage for \mathbf{Z} . This ratio is the rank fraction RF,

$$RF = \frac{k(m+n)}{mn} \quad (8)$$

If $k \ll m, n$, that is, \mathbf{Z} is of low rank, then RF is much less than unity. If $m = n$, and \mathbf{Z} is full rank, that is $k = m$, then $RF = 2$, and the UV representation takes twice the storage of \mathbf{Z} .

In this work we will express rank fraction on a $10\log_{10}$ scale in decibels (dBrf) (since single block rank fractions have approached 0.01), that is

$$dBrf = 10\log_{10}\left(\frac{k(m+n)}{mn}\right). \quad (9)$$

Thus a rank fraction of 0.01, 99% compressed, has a rank fraction on this scale of -20 dBrf. We will have occasion to see such values for individual block matrices.

³ The term rank fraction and matrix block compression will be used interchangeably in this report.

Before we discuss the ACA efficient computation of compressed low rank matrices, let us first examine the rank fraction for various components of the MOM system matrix that show they are indeed compressible.

III. System Block Equation Compressibility Maps

In this report we show results for the frequency domain electric field integral equation (EFIE) for PEC surfaces. Rao-Wilton Glisson (RWG) triangle basis functions [11, 12] with Galerkin test functions are used so that a symmetric block system results where the unknowns have been grouped into local spatial regions. The self and near self-terms are computed using the radial-angular approach described in [13].

Matrix element integration uses three points each in source and test triangles. For self and near self, each triangle is split into three parts and a fourth-order Gaussian quadrature is used for the radial integration [13]. Self-term symmetry is forced by swapping source and test triangles and using the average.

In the symmetric matrix results to follow, both lower and upper triangles will be shown for ease of presentation and understanding. Of course, they are just the transpose of each other.

Off-diagonal blocks of the impedance \mathbf{Z} matrix have been known to be compressible with low rank approximates [1-6, 9, and 18]. How about the other system matrices, such as those comprising the LU factorization of \mathbf{Z} , the RHS voltage vector / matrix when \mathbf{V} is due to many incident plane waves, and \mathbf{J} , the current solution?

$$\mathbf{Z} \mathbf{J} = \mathbf{L} \mathbf{U} \mathbf{J} = \mathbf{V}. \quad (10)$$

Let us consider two electrically large geometries, SLICY and Open Pipe, Figures 4 and 5, where the cobblestone region grouping has been applied. We will examine the rank fraction of the system LU blocks.

Rank and tolerance are related. In the rank fraction maps shown in this report, the tolerance is relative to the norm of the largest block in a block row, i.e., the diagonal block.

SLICY: This target is composed of two vertical circular interacting cylinders, one of which is a cavity, and of a corner reflector interacting with the cylinders. These multiple bounce interactions are principally specular in nature. The number of unknowns is 90,711, maximum group size is 800, and average triangle edge length is 0.117λ . The ACA approximation tolerance for the LU factorization matrix blocks was 10^{-4} while the ACA tolerance for the solve \mathbf{V} and \mathbf{J} blocks was 10^{-3} .

The block system \mathbf{Z} rank fraction is shown in Figure 6 on a dBrf scale. Clearly the rank fraction of the interaction matrices decrease as the interaction distance increases. The sparseness⁴ of \mathbf{Z} is 95%.

The block system LU factored matrix rank fraction is shown in Figure 7, also on a dBrf scale. The LU off-diagonal blocks are low rank, but not as low as the original \mathbf{Z} blocks due to factorization fill in. The average sparseness of the entire matrix is now 90%.

The important observation is that for this highly interacting target, the LU factored form is still 90% sparse (as comprised of individual low rank blocks).

What about the RHS \mathbf{V} matrix and the current matrix solution \mathbf{J} ? Are these compressible? If the problem is antenna or bistatic scattering with only a few RHS's, then the answer is no. How about monostatic scattering where there are many RHS incident plane waves? Shown in Figure 8 is the rank fraction map for 250 angles, two polarizations each, so that the number of RHS = 500 and $\mathbf{V} \in \mathbb{C}^{90,711 \times 500}$.⁵ The rank fraction of the blocks comprising \mathbf{V} , shown in Figure 8 for a tolerance of $\epsilon = 10^{-4}$, is approximately -8 dBrf which represents a sparseness of 86%. Clearly \mathbf{V} is of compressible.

What about the current solution matrix $\mathbf{J} \in \mathbb{C}^{90,711 \times 500}$ for this monostatic case? The block rank fraction for \mathbf{J} is shown in Figure 8. It too is compressible with an average block rank fraction of approximately -3 dBrf that represents a sparseness of 55% compared to 86% for \mathbf{V} . Because of fill in, current solution blocks are typically not as sparse as RHS forcing matrices \mathbf{V} .

OPEN PIPE: This geometry has many scattering features: internal wave guide propagation and/or cavity multi-bounce propagation, external traveling waves, leading and trailing edge rim diffraction, creeping waves, and specular scattering mechanisms depending on plane wave excitation angle. The open pipe represents a robust target for computing sensitive scattering mechanisms. This pipe has a length of 36" and the inner and outer diameters are 3.87" and 4.0". In the rank fraction maps to be shown, the excitation frequency is 6 GHz, the total number of unknowns is 92,220, the maximum group size is 800 unknowns per group and the LU ACA tolerance is $\epsilon = 10^{-4}$.

The open pipe block system \mathbf{Z} rank fraction is shown in Figure 9 on a dBrf scale. Clearly the rank fraction of the interaction matrices decrease as the interaction distance increases. Overall sparseness is 95%.

⁴ The sparseness of the compressed \mathbf{Z} matrix is defined as the total memory storage, including the non-compressed diagonal blocks, relative to the storage required for uncompressed \mathbf{Z} matrix.

⁵ The number of RHS per solve is sometimes less than the total number of RHS for a given problem. The number of RHS per solve depends on available computer memory. Thus for many RHS and limited memory, a number of solves are required. Sparseness for \mathbf{V} and \mathbf{J} here is stated for the number of RHS solved per group.

The block system LU factored matrix rank fraction is shown in Figure 10, also on a dBrf scale. The LU blocks are low rank, but not as low as the original Z blocks due to fill in. Overall sparseness drops to 92%.

The monostatic RHS $\mathbf{V} \in \mathbb{C}^{92,220 \times 500}$ is compressible, see Figure 11. Overall V sparseness is 96%. The corresponding current solution $\mathbf{J} \in \mathbb{C}^{92,220 \times 500}$, is also compressible 85% sparse, but not as low as V due to fill in.

These results for two distinctly different interacting target types clearly show that the system equation block matrices for \mathbf{Z} , \mathbf{L} , \mathbf{U} , \mathbf{V} and \mathbf{J} are compressible. The next question is, can we take advantage of this observation for reducing memory and speeding up the LU factor and solve?

IV. Adaptive Cross Approximation

It should be clear by now that for electrically large problems, the various block components of the system matrix $\mathbf{ZJ} = \mathbf{LUJ} = \mathbf{V}$ are compressible when unknowns are spatially grouped. The next question is how do we take computational advantage of this? Clearly performing a singular value decomposition (SVD) on each block would be a possibility, but that has a complexity of m^3 per block, not very efficient even though that was the approach in [19] for compressing the Z matrix before the advent of ACA. One could also compute low rank blocks using the Modified Graham Schmidt (MGS) approach as was done in [18]. But that also is of significant complexity and is no longer suggested by the authors [1].

What is needed is an approach to compute low rank block approximates that *do not* require first filling either the \mathbf{Z} or \mathbf{LU} blocks and has a low operations count. See Appendix for detailed explanation of the symmetric LU block factorization.

The Adaptive Cross Approximation (ACA) algorithm as recently developed by Bebendorf [2] and further discussed by [1, 3-6, 9] is just such an approach.

The reader can consult the references just cited for the ACA details, but briefly, the ACA method computes the compressed approximate form of matrix $\mathbf{A} \sim \mathbf{Au Av}$ using only k rows and k columns of \mathbf{A} to obtain \mathbf{Au} and \mathbf{Av} . The ACA operations count [2] is of order $O(k^2(m+n))$ where $\mathbf{A} \in \mathbb{C}^{m \times n}$, $\mathbf{Au} \in \mathbb{C}^{m \times k}$, $\mathbf{Av} \in \mathbb{C}^{k \times n}$. If \mathbf{A} is low rank, $k \ll m, n$, then the ACA algorithm has a very small operations count, and more importantly, requires only a small fraction of \mathbf{A} for input.

The ACA starts its 0th approximation by choosing a row and column of \mathbf{A} to form the first outer product $\mathbf{u}_0 \mathbf{v}_0$ approximation. These become the first row and column members of \mathbf{Au} and \mathbf{Av} . Adding more columns to \mathbf{Au} and rows to \mathbf{Av} makes successive improvements to reduce the approximation error. These new rows and columns are based on a maximum element pivoting strategy for choosing the next row and column [1,2]. This has the very practical effect of speeding the convergence. Successive improvements

converge when the norm of the k^{th} $\mathbf{u}_k \mathbf{v}_k$ outer product's next computed term is less than the tolerance ϵ times the norm of \mathbf{A} (or in the MOM equation case, less than the norm of the largest row block norm). The norm of \mathbf{A} , when required, is computed using a recursive approach during the ACA loop [2]. Note that \mathbf{A} does not have to exist in full form for the ACA operation provided one has a subroutine to compute the few needed rows and columns of \mathbf{A} .

An ACA example of compression for the $C^{220 \times 214}$ low rank MOM matrix of Figure 3 is shown in Figure 12 where the ACA was used to compute the compressed representation. Just 20 terms are needed for a relative tolerance of 10^{-5} . Also shown are the stepwise errors relative to the ACA recursive norm [3] and the actual norm. The singular values are also plotted.

Operations Count Efficiencies

Rank fraction is a direct indicator of memory storage for low rank matrices, and is one of the key benefits of working with compressed representations. Operations (ops) count reduction is the other key benefit.

When \mathbf{A} is represented as a $\mathbf{U}\mathbf{V}$ outer product with $k \ll m, n$, we say matrix \mathbf{A} is compressed. In the notation that follows, we will write compressed matrices as $\mathbf{A} \approx \mathbf{A}\mathbf{u} \mathbf{A}\mathbf{v}$.

Matrix multiplication of compressed matrices is very efficient *if done in the proper order*. In the following example, we will multiply compressed matrix $\mathbf{A} \approx \mathbf{A}\mathbf{u} \mathbf{A}\mathbf{v}$ times a matrix \mathbf{B} to form matrix $\mathbf{C} = \mathbf{A}\mathbf{B}$. We will assume that \mathbf{C} is used in its compressed form $\mathbf{C}\mathbf{u} \mathbf{C}\mathbf{v}$.

Case One: Matrix \mathbf{B} is not compressed. We compare the operations count for the product $\mathbf{C} = \mathbf{A}\mathbf{B}$ versus the approximation $\mathbf{C}\mathbf{u} \mathbf{C}\mathbf{v} = \mathbf{A}\mathbf{u} \mathbf{A}\mathbf{v} \mathbf{B}$. For $\mathbf{A} \in C^{m \times n}$ and $\mathbf{B} \in C^{n \times p}$, the product operations count to compute \mathbf{C} is (mnp) .

The compressed multiply has the form

$$\begin{aligned}
 \mathbf{C} &= \mathbf{A} \mathbf{B} \\
 \mathbf{C} &\approx \mathbf{C}\mathbf{u} \mathbf{C}\mathbf{v} \\
 \mathbf{A} \mathbf{B} &\approx \mathbf{A}\mathbf{u} \mathbf{A}\mathbf{v} \mathbf{B}
 \end{aligned}$$

$$\mathbf{A}\mathbf{u} \mathbf{A}\mathbf{v} \mathbf{B} = \begin{bmatrix} \mathbf{A}\mathbf{u} \\ \mathbf{A}\mathbf{v} \end{bmatrix} \left(\begin{bmatrix} \mathbf{B} \end{bmatrix} \right) = \begin{bmatrix} \mathbf{C}\mathbf{u} \\ \mathbf{C}\mathbf{v} \end{bmatrix} \quad (11)$$

In this case, $\mathbf{C}\mathbf{u} = \mathbf{A}\mathbf{u}$ and the ops count for $\mathbf{C}\mathbf{v} = \mathbf{A}\mathbf{v} \mathbf{B}$ is (knp) .

The ops efficiency for computing compressed \mathbf{C} versus full $\mathbf{C} = \mathbf{A}\mathbf{B}$ is 1st order small in (k/m) ,

$$\frac{kn p}{mnp} = \frac{k}{m} \ll 1 \text{ if } k \ll m. \quad (12)$$

Case Two: Matrix \mathbf{B} is compressed and available, $\mathbf{B} = \mathbf{B}_u \mathbf{B}_v$ where $\mathbf{B}_u \in C^{n \times r}$ and $\mathbf{B}_v \in C^{r \times p}$ with $r \ll n, p$. The operations count for compressed \mathbf{C} follows from

$$\mathbf{C} = \mathbf{A} \mathbf{B}$$

$$\mathbf{C} \approx \mathbf{C}_u \mathbf{C}_v$$

$$\mathbf{A} \mathbf{B} \approx \mathbf{A}_u \mathbf{A}_v \mathbf{B}_u \mathbf{B}_v$$

$$\mathbf{A}_u \mathbf{A}_v \mathbf{B}_u \mathbf{B}_v = \begin{bmatrix} \mathbf{A}_u \\ \end{bmatrix} \left(\begin{bmatrix} \mathbf{A}_v & \end{bmatrix} \begin{bmatrix} \mathbf{B}_u \\ \end{bmatrix} \right) \begin{bmatrix} \mathbf{B}_v & \end{bmatrix} = \begin{bmatrix} \mathbf{A}_u \\ \end{bmatrix} \begin{bmatrix} \mathbf{T} & \end{bmatrix} \begin{bmatrix} \mathbf{B}_v & \end{bmatrix} \quad (13)$$

$$= \begin{bmatrix} \mathbf{C}_u \\ \end{bmatrix} \begin{bmatrix} \mathbf{C}_v & \end{bmatrix}$$

The intermediate matrix $\mathbf{T} = \mathbf{A}_v \mathbf{B}_u$ is of an inner product form with a reduced ops count of (knr) .

The ops count for multiplying the small matrix \mathbf{T} with \mathbf{A}_u is (mkr) or with \mathbf{B}_v is (rkn) .

The ops cost to compute $\mathbf{C}_u \mathbf{C}_v$ is $\{ knr + mkr \}$ or $\{ knr + rkn \}$ either of which is much less than computing full \mathbf{C} with ops of (mnp) .

The ops efficiency for computing compressed versus full \mathbf{C} is 2nd order small

$$\frac{knr + mkr}{mnp} = \frac{kr(m+n)}{mnp} \ll 1 \text{ if } k, r \ll m, n, p. \quad (14)$$

If \mathbf{A} and \mathbf{B} are square with m, n , and p equal, the ops count efficiency is 2nd order small ($2kr$ compressed operations $\ll m^2$ uncompressed operations),

$$\frac{knr + mkr}{mnp} = \frac{kr2m}{m^3} = \frac{2kr}{m^2} \ll 1 \text{ if } k, r \ll m. \quad (15)$$

The key observation is that compressed matrix multiplication is very operations count efficient as well as memory efficient.

When working with compressed matrices in a multiplication sequence, one always tries to minimize the operations count by choosing the multiply order that is of the inner product versus outer product form since the inner product form has significantly fewer operations.

Operations Count Metric

Just as low rank matrices save significant memory storage, they also save significant number of operations count in the compressed block LU factorization. A block wise measure of reduced ops count is the ratio of compressed to non-compressed LU factorization operations, measured on a dB scale ($10 \log_{10}$),

$$dB_ops_reduction = 10 \log_{10} \left(\frac{\text{ACA LU Factor ops}}{\text{Normal LU ops}} \right) \quad (16)$$

where the numerator is the LU block operations count using compressed matrices to compute the compressed block $[\mathbf{Uu} \ \mathbf{Uv}]_{i,j}$,

$$[\mathbf{Uu} \ \mathbf{Uv}]_{i,j} = [\mathbf{Zu} \ \mathbf{Zv}]_{i,j} - \sum_{p=1}^{i-1} [\mathbf{Uv}^T \ \mathbf{Uu}^T]_{ip} \mathbf{D}_{pp}^{-1} [\mathbf{Uu} \ \mathbf{Uv}]_{pj} \quad (17)$$

and the denominator is the block operations count required to compute the un-compressed LU factor \mathbf{U}_{ij} using non-compressed blocks,

$$\mathbf{U}_{ij} = \mathbf{Z}_{ij} - \sum_{p=1}^{i-1} \mathbf{U}_{ip}^T \mathbf{D}_{pp}^{-1} \mathbf{U}_{pj} \quad (18)$$

An example of the ops count reduction is shown in Figure 13 for the 90,711 unknowns SLICY problem where reductions of up to -25 dB are seen.

V. Validation Results for SLICY

The computational times reported here were obtained using a PC Workstation with two Xeon core 2 duo processors (4 cores total), 16 GB of memory, and four SCSI 15k rpm disk drives set up in RAID 0 (disk stripe) mode. The operating system was Windows XP 64.

SLICY [20] is a target that has multiple interactions between the ground plane, two vertical cylinders (one of which is a cavity), a dihedral, and two trihedrals, Figure 4. The Mercury MOM model description and run parameters are shown in Table 1.

Table 1 SLICY Validation Run Parameters

Geometry	SLICY
Number of Unknowns	90,711
Frequency	514.3 MHz
Average Edge Length (λ)	0.117
Couples per λ^2	259
Maximum region size	800
Number of RHS	722
ACA factor tolerance	10^{-4}
ACA solve tolerance	10^{-4}
Z sparseness	95%
LU factor sparseness	90%
Voltage sparseness	86%
Current solution sparseness	55%
Total wall run time (PC Workstation)	0.78 hr

The full polarization scattering matrix for backscatter was computed for an azimuth cut with 361 angles (722 RHS's) at an elevation angle of 15° look-down. The reference full matrix computation⁶ was computed with the method of moments code named *CARLOS* 4.4 using the iterative out-of-core solver option with a block LU (BLU) pre-conditioner, one polarization per run. Run time was approximately 48 hours on a Silicon Graphic Incorporated (SGI) R12000 processor. SLICY measured data were compared to *CARLOS* predictions in [21].

The polarization scattering matrix results for HH, VV, and cross polarization VH, HV of the compressed ACA MOM solution virtually lay on top of the full matrix solution of *CARLOS*, Figures 14-16.

This agreement clearly shows that a very significant fraction of the MOM system equation matrix was low rank and most of the matrix elements do not contain useful problem information and can be expressed in compressed form. Spatial grouping, low rank, and ACA LU factor / solve can effectively be used to eliminate a large fraction of the problem which does not contain useful physics. This results in much less memory, much faster run times, and the ability to use inexpensive computer platforms.

⁶ Thanks to Steve Carter from the National Ground Intelligence Center for providing the *CARLOS* electromagnetic predictions of the SLICY model.

VI. Complexity Results for Open Pipe

Complexity metrics using *only* the number of problem unknowns N have been the traditional approach for measuring the efficacy of “fast” methods and that is the approach that will be taken in this work.

However, it is worth noting that complexity is a poor measure since the following parameters significantly influence the efficacy of a given “fast” method:

Surface area measured in λ^2 : Complexity involves the number of unknowns N in a given problem as a measure of problem size. A much more robust approach would be to use the surface area measured in λ^2 (for surface integral equations). This metric would allow taking into account benefits due to higher order basis/test functions, better integration techniques and most importantly, reduced sampling in terms of unknowns / λ^2 .

Geometry mutual coupling: The degree of mutual coupling in a given geometry can influence complexity. This would influence the time need for iterative solvers to converge, while in direct methods, it would influence the amount of compression for the LU factor and solve matrices.

Computer hardware cost: Can the algorithm be used on an inexpensive PC, a workstation, a cluster of networked computers, or a super computer? Computer costs involve the amount of memory and the number of cpu’s and hard drives.

Wall clock time: This is not cpu time, but rather the time between when a user submits a job and the time results are obtained. Reduced wall clock time is associated with increased computer resource expense. Reduced wall clock time allows more design space iterations and is critical for applications using computational methods. Computation time can also be influenced by matrix algorithm efficiencies. Algorithms, which make use of highly optimized matrix libraries such as Level 3 BLAS and Lapack, can have significant impact on reducing time even though operations count remains constant.

Accuracy: Accuracy affects run times. “Fast” methods always involve tolerances. In our case, we have an LU factor ACA tolerance and a RHS ACA tolerance for filling the V matrix, and for the forward and back solves. Tight tolerances will lead to greater run times.

“Fast” methods need to be compared against a metric that takes into account these issues; however, the following complexity metrics use only problem size as measured by N , the number of unknowns.

Open Pipe Results: The open pipe geometry [22], outlined previously, was run for fourteen frequencies, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 18 and 20 GHz computing backscatter RCS over 0 to 180 degrees. Unknown sample edge lengths averaged 0.1λ (380-400 unknowns/ λ^2). The number of right hand sides ranged from 310 to 6162. The

ACA convergence tolerance⁷ was 10^{-5} of the norm of the diagonal block (in the block row). The ACA tolerance for solve, for computing V, for LU forward and back solve to obtain J, was 10^{-4} .

Table 2 Open Pipe Complexity Run Cases

Geometry	Open Pipe
Number of Unknowns	2,592 – 1,025,109
Frequencies	1 to 20 GHz
Average Edge Length (λ)	0.098 - 0.1
Couples per λ^2	410-383
Maximum region size	2000 - 3000
Number of RHS	310 - 6142
ACA factor tolerance	10^{-5}
ACA solve tolerance	10^{-4}
Z sparseness	63% - 99%
LU factor sparseness	62% - 97%
Total Wall Time	17 sec – 121 hr

Several of the very largest problems had to store the LU factorization to disk due to the limitation of installed memory. This has the effect of longer LU factor times due to the slower disk IO access.

The following comparisons are relative to the standard non-sparse complexity methodology for a LU MOM solution. The system matrix memory required for ACA LU factorization scales as $N^{1.5}$, Figure 17, compared to N^2 for the standard factorization. Fill time for the Z matrix, as required for ACA LU factorization input, Figure 18, scales as $N^{1.3}$, also compared to N^2 for the standard fill time.

ACA LU factorization time scales as $N^{2.1}$, Figure 19, compared to N^3 . Solve time per RHS vector, Figure 20, scales as $N^{1.67}$ compared to N^2 .

The overall problem wall clock time scales as $N^{1.9}$, Figure 21. This is the time for the complete problem.

The 157,059 unknowns' problem, at a frequency of 8 GHz, with 2466 RHS was completed on the PC Workstation, in core, with a wall time of 1.28 hours.

VII. SLICY and Open Pipe Run Comparisons

A comparison of these two targets, each with approximately the same number of unknowns is instructive, Table 3. Each has distinctly different scattering mechanisms.

⁷ The Open Pipe has more non-specular low-level scattering mechanisms than SLICY, which has mostly high-level specular mechanisms. Thus the Open Pipe requires tighter ACA factor and solve tolerances.

Each has different average edge lengths and ACA tolerances for LU factor. Each has different surface areas measured in square wavelength.

Yet, when matrix storage, matrix fill and LU factor and total run time are compared normalized to wavelength, each of these parameters are roughly equivalent.

Table 3 Comparison of similar Open Pipe and SLICY runs

Geometry	Open Pipe	SLICY
Surface Area	231 λ^2	350 λ^2
Number of Unknowns	92,220	90,711
Average Edge Length (λ)	0.1	0.117
Couples per λ^2	400	259
Maximum region size	800	800
Number of RHS	1850	722
ACA factor tolerance	10^{-5}	10^{-4}
ACA solve tolerance	10^{-4}	10^{-4}
Z sparseness	95%	95%
LU factor sparseness	92%	90%
Voltage sparseness	96%	86%
Current solution sparseness	85%	55%
LU memory (GB)	2.8	3.3
<i>LU memory per λ^2</i>	<i>0.012</i>	<i>0.0093</i>
Matrix fill & factor (hr)	0.446	0.70
<i>Matrix fill & factor per λ^2</i>	<i>$1.9 \cdot 10^{-3}$</i>	<i>$2.0 \cdot 10^{-3}$</i>
Total run time (hr)	0.512	0.78
<i>Total run time per λ^2</i>	<i>$2.2 \cdot 10^{-3}$</i>	<i>$2.2 \cdot 10^{-3}$</i>

Summary

It is clear that electrically large problems with unknowns grouped in local spatial regions have a system matrix \mathbf{Z} and its lower/upper (**LU**) factored forms that are comprised of compressible off-diagonal blocks. For monostatic scattering, where the right hand side (RHS) voltage matrix is composed of many incident plane waves, blocks of \mathbf{V} are also compressible as well as the current solution \mathbf{J} .

Compressible matrices and their low rank approximations fundamentally means that most of the Method of Moments (MOM) system matrix equation elements, before compression, contain very little physical information. The Adaptive Cross Approximation (ACA) can be used to extract and compress the system equation elements, keeping only the necessary physics. The ACA can be used for all steps of the solution: filling the \mathbf{Z} matrix; **LU** Factoring; and solving. This results, depending on problem size, in *orders of magnitude* less memory and run times. One can use inexpensive computer resources for problems that previously took super computers. Or solve problems that could not be solved at all.

It is also important to point out that this ACA LU factorization and solution is completely within the context of standard Method of Moments. There are no requirements for new basis or test functions, no requirements for far or near field distinction, no requirements for auxiliary “equivalent” sources on a grid, and no requirement for analytic Green’s function expansion.

This direct LU factor and solve technique is validated against another well-known code for a 90,711 unknown SLICY geometry. Complexity of this approach is studied by numerical experiment for specific error tolerances for an open pipe scattering target with number of unknowns ranging from 2,592 to 1,025,109 and the number of RHS varying from 310 to 6,162. Matrix fill time scales as $N^{1.3}$, LU matrix memory storage scales as $N^{1.5}$, LU factor time scales as $N^{2.1}$, and the time per RHS solve scales as $N^{1.67}$. Complete solution clock time scales as $N^{1.9}$.

Appendix

Complex Symmetric LU Block Factorization and Solve

The formulas used for block symmetric **LU** factor and for forward / back solve are obtained by comparing the symmetric factor form to the standard non-symmetric scalar factor forms.

Block forms of scalar LU expressions are straightforward. One simply uses block matrices in place of scalar entities. The only difference is for division, which in block form becomes a matrix inverse, i.e., $1/d \Rightarrow \mathbf{D}^{-1}$. And the order of operations must follow matrix rules for multiply.

Block factorization [14-17] of a complex symmetric matrix is

$$\mathbf{Z} = \mathbf{LDL}^T = \mathbf{U}^T \mathbf{DU} \quad (\text{A-1})$$

where T represents transpose. **L** is lower triangular, **U** is upper triangular and **D** are the diagonal entities (scalar or blocks). Either **L** or **U** is required when **Z** is symmetric.

Our goal is to obtain the formulas required to compute **U** and to obtain the solution formulas. We will do this by comparing the above expression with that for non-symmetric scalar matrices.

Scalar Non-Symmetric LU Formulas

The standard LU non-symmetric factorization for a scalar matrix is

$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \quad (\text{A-2})$$

The column wise factorization formulas for u and l are [15],

$$u_{ij} = z_{ij} - \sum_{p=1}^{i-1} l_{ip} u_{pj} \quad (\text{A-3})$$

$$l_{ij} = \frac{1}{u_{jj}} \left(z_{ij} - \sum_{p=1}^{j-1} l_{ip} u_{pj} \right) \quad (\text{A-4})$$

where u_{jj} are the diagonal entries.

The solve process has forward and back steps. Given $\mathbf{ZJ} = \mathbf{LUJ} = \mathbf{V}$, the solution for \mathbf{J} starts with $\mathbf{LX} = \mathbf{V}$ where $\mathbf{X} = \mathbf{UJ}$. This is the forward solve process that uses the lower triangle matrix \mathbf{L} starting with the 1st row, $i=1$,

$$x_i = v_i - \sum_{p=1}^{i-1} l_{ip} x_p \quad (\text{A-5})$$

The back solve process for \mathbf{J} uses the upper triangle matrix \mathbf{U} , starting with the last row, $i=n$:

$$j_i = \frac{1}{u_{ii}} \left(x_i - \sum_{p=i+1}^n u_{ip} j_p \right). \quad (\text{A-6})$$

The next step is to compare these standard expressions with those for symmetric block matrices.

LU Factorization of a Complex Symmetric Block Matrix

This work uses the upper \mathbf{U} form for symmetric factorization, and for purposes of comparison, the symmetric block will be identified as \mathbf{U}' as the block contents will not be the same entities as the u 's of non-symmetric scalar expressions,

$$\mathbf{Z} = \mathbf{U}'^T \mathbf{D} \mathbf{U}' \quad (\text{A-7})$$

Expanding to full form

$$\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \mathbf{Z}_{13} \\ \mathbf{Z}_{12}^T & \mathbf{Z}_{22} & \mathbf{Z}_{23} \\ \mathbf{Z}_{13}^T & \mathbf{Z}_{23}^T & \mathbf{Z}_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{U}'_{12}^T & \mathbf{1} & \mathbf{0} \\ \mathbf{U}'_{13}^T & \mathbf{U}'_{23}^T & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{U}'_{12} & \mathbf{U}'_{13} \\ \mathbf{0} & \mathbf{1} & \mathbf{U}'_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (\text{A-8})$$

The diagonal entries of \mathbf{U}' are unity. Putting $\mathbf{U}'^T \mathbf{D} \mathbf{U}'$ into LU form, we first expand $\mathbf{D} \mathbf{U}'$ to obtain the form

$$\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \mathbf{Z}_{13} \\ \mathbf{Z}_{12}^T & \mathbf{Z}_{22} & \mathbf{Z}_{23} \\ \mathbf{Z}_{13}^T & \mathbf{Z}_{23}^T & \mathbf{Z}_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{U}'_{12}^T & \mathbf{1} & \mathbf{0} \\ \mathbf{U}'_{13}^T & \mathbf{U}'_{23}^T & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{11} \mathbf{U}'_{12} & \mathbf{D}_{11} \mathbf{U}'_{13} \\ \mathbf{0} & \mathbf{D}_{22} & \mathbf{D}_{22} \mathbf{U}'_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_{33} \end{bmatrix} \quad (\text{A-9})$$

Making the substitutions

$$\begin{aligned} \mathbf{U}_{ij} &= \mathbf{D}_{ii} \mathbf{U}'_{ij} \\ \mathbf{U}_{ii} &= \mathbf{D}_{ii} \\ \mathbf{L}_{ij} &= \mathbf{U}'_{ji}^T = [\mathbf{D}_{ii}^{-1} \mathbf{U}_{ij}]^T = \mathbf{U}_{ji}^T \mathbf{D}_{ii}^{-1} \end{aligned} \quad (\text{A-10})$$

we arrive at the standard LU form

$$\begin{bmatrix} \mathbf{Z}_{11} & \mathbf{Z}_{12} & \mathbf{Z}_{13} \\ \mathbf{Z}_{12}^T & \mathbf{Z}_{22} & \mathbf{Z}_{23} \\ \mathbf{Z}_{13}^T & \mathbf{Z}_{23}^T & \mathbf{Z}_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{1} & \mathbf{0} \\ \mathbf{L}_{31} & \mathbf{L}_{31} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} & \mathbf{U}_{13} \\ \mathbf{0} & \mathbf{U}_{22} & \mathbf{U}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_{33} \end{bmatrix} \quad (\text{A-11})$$

with factorization

$$\mathbf{U}_{ij} = \mathbf{Z}_{ij} - \sum_{p=1}^{i-1} \mathbf{U}_{ip}^T \mathbf{D}_{pp}^{-1} \mathbf{U}_{pj} \quad (\text{A-12})$$

and solve

$$\mathbf{X}_i = \mathbf{V}_i - \sum_{p=1}^{i-1} \mathbf{U}_{ip}^T \mathbf{D}_{pp}^{-1} \mathbf{X}_p ; \text{Forward} \quad (\text{A-13})$$

$$\mathbf{J}_i = \mathbf{D}_{ii}^{-1} \left(\mathbf{X}_i - \sum_{p=i-1}^1 \mathbf{U}_{ip} \mathbf{J}_p \right) ; \text{Back}$$

which completes the block symmetric matrix LU factor and solve expressions.

ACA LU Block Factorization

ACA factorization computes blocks of \mathbf{U} in low rank compressed form

$$\mathbf{U} = \mathbf{Uu Uv} \quad (\text{A-14})$$

which are transpose stored for efficient organization,

$$\mathbf{U}^T = ([\mathbf{Uu}] [\mathbf{Uv}])^T = [\mathbf{Uv}]^T [\mathbf{Uu}]^T \quad (\text{A-15})$$

This is a straightforward application of the ACA algorithm to the LU block factorization computational expression,

$$\mathbf{U}_{ij} = [\mathbf{Uu Uv}]_{i,j} = \mathbf{Z}_{ij} - \sum_{p=1}^{i-1} [\mathbf{Uv}^T \mathbf{Uu}^T]_{ip} \mathbf{D}_{pp}^{-1} [\mathbf{Uu Uv}]_{pj} \quad (\text{A-16})$$

The ACA algorithm requires only selected rows and columns of \mathbf{U}_{ij} to obtain the compressed form $\mathbf{Uu Uv}$. Only a few selected rows and columns of \mathbf{Z}_{ij} must be computed.

It is important to note that the blocks of the original \mathbf{Z} matrix are never filled. Only the \mathbf{Z}_{ij} rows and columns, needed by the ACA algorithm to compute the \mathbf{U}_{ij} block factorization, are filled. This further means that this procedure for ACA factorization does not have a direct analogy to the direct matrix fill concept.

Forward and back solve forms are:

$$\mathbf{X}_i = \mathbf{X}_i - \sum_{p=1}^{i-1} (\mathbf{U}\mathbf{u}\mathbf{U}\mathbf{v})_{ip}^T \mathbf{D}_{pp}^{-1} (\mathbf{X}\mathbf{u}\mathbf{X}\mathbf{v})_p ; \text{Forward}$$

(A-17)

$$\mathbf{J}_i = \mathbf{D}_{ii}^{-1} \left(\mathbf{X}_i - \sum_{p=i-1}^1 (\mathbf{U}\mathbf{u}\mathbf{U}\mathbf{v})_{ip} (\mathbf{J}\mathbf{u}\mathbf{J}\mathbf{v})_p \right) ; \text{Back}$$

References

1. Kezhong Zhao, Marinou Vouvakis and Jin-Fa Lee, "The Adaptive Cross Approximation Algorithm for Accelerated Method of Moment Computations of EMC Problems," IEEE Transactions on Electromagnetic Compatibility, Vol. 47, No. 4, pp. 763-773, November 2005.
2. Mario Bebendorf, "Approximation of boundary element matrices," Numer. Math. (2000) 86: 565:589.
3. Stefan Kurz, Oliver Rain, and Sergej Rjasanow, "Application of the Adaptive Cross Approximation Technique for the Coupled BE-FE Solution of Electromagnetic Problems," IABEM 2002, International Association for Boundary Element Methods, UT, Austin TX, May 28-30, 2002.
4. Stefan Kurz, "The Adaptive Cross-Approximation Technique for the 3-D Boundary-Element Method," IEEE Transactions on Magnetics, Vol. 38, No. 2, pp. 421-424, March 2002.
5. Sergej Rjasanow, "Adaptive Cross Approximation of Dense Matrices," IABEM 2002, International Association for Boundary Element Methods, UT Austin, TX, USA, May 28-30, 2002.
6. J. Ostrowski, Z. Andjelic, M. Bebendorf, B. Cranganu-Cretu and J. Smajic, "Fast BEM-Solution of Laplace Problems with H-Matrices and ACA", IEEE Transaction on Magnetics, 42 (4) : 627-630, April 2006
7. Francis X. Canning, "Simply Sparse, A New and General Fast Method", Electromagnetic Code Consortium Meeting, Kirtland Air Force Base, NM, May 8, 2002.
8. John Shaeffer and Francis Canning, "Preliminary Scaling of SPARSE MOM On a PC for Problem Sizes from 2,592 to 157,059 Unknowns", Electromagnetic Code Consortium (EMCC) Meeting, May 2004, Seattle, WN.
9. John Shaeffer and Francis Canning, "Adaptive Cross Approximation for MOM Matrix Fill for PC Problem Sizes to 157000 Unknowns", Applied Computational Electromagnetics Society (ACES) meeting, March 2005, Honolulu, HI.
10. Francis X. Canning, "Compression of Interaction Data using Directional Sources or Testers," U.S. Patent Application, filed January 2003.
11. S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic Scattering by Surface of Arbitrary Shape," IEEE Trans. Antennas Propag., vol. AP-30, pp. 409-418, May 1982.

12. Andy F. Peterson, S. L. Ray, and R. Mittra, **Computational Methods for Electromagnetics**, IEEE Press, New York, NY, 1998.
13. P. W. Fink, D. R. Wilton and M. A. Khayat, "Issues and Methods Concerning the Evaluation of Hypersingular and Near-Hypersingular Integrals in BEM Formulations," ICEAA Conference, September 2005, Torino, Italy.
14. Gene Golub and Charles Van Loan, **Matrix Computations**, Third Edition, The John Hopkins University Press, Baltimore, MD, 1996.
15. Lloyd Trefethen and David Bau, **Numerical Linear Algebra**, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.
16. William Press, Saul Teukolsky, William Vetterling and Brian Flannery, **Numerical Recipes**, Second Edition, Cambridge University Press, 1992.
17. E. Anderson et. al., **LAPACK Users' Guide**, Second Edition, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.
18. Robert Burkholder and Jin-Fa Lee, "Fast Dual-MGS Block-Factorization Algorithm for Dense MoM Matrices," IEEE Trans. Antennas and Propagation, Vol. 52, No. 7, July 2004.
19. S. Kapur and D. E. Long, "IES³: A fast integral equation solver for efficient 3-dimensional extraction," in *Proc. ICCAD*, 1997, pp. 448-455.
20. J Gatesman, C. Beaudoin, R. H. Giles, J. Waldman, and W. E. Nixon, "VHF/UHF Imagery and RCS Measurements of Ground Targets in Forested Terrain", Aerosence Conference, Orlando, FL, April 2002.
21. J. Putnam and J. Kotulski, "Parallel CARLOS-3D Code Development," Applied Computational Electromagnetics Society (ACES) meeting, March 1996, Monterey, CA.
22. Kam Hom and Deirdre Ryan, "EMCC Code Assessment Results of the Pipe and Cavity Targets", Electromagnetic Code Consortium (EMCC) Meeting, May 2004, Seattle, WA.

List of Figures

Figure 1 Spatially grouped unknowns lead to low rank block matrices Z, L, U, V, and J for electrically large bodies. (from [9])

Figure 2 Cobblestone regions on a flat plate geometry: 24933 unknowns, maximum region size 600 unknowns, 51 groups

Figure 3 Singular values for a 220 by 214 Z interaction block matrix. Singular values drop to machine precision by $k = 30$.

Figure 4 SLICY test geometry showing unknowns grouped into local spatial regions

Figure 5 Open Pipe Geometry showing unknowns grouped into local spatial regions

Figure 6 SLICY Z matrix rank fraction in dBrf, 90,711 unknowns, maximum group size 800, overall sparseness is 95 %

Figure 7 SLICY LU U block rank fraction in dBrf, 90,711 unknowns, , maximum group size 800 overall sparseness is 90 %

Figure 8 SLICY RHS voltage matrix V(black) and current solution J(blue) rank fraction, 20 dB scale ($0.01 < RF < 1.0$), by region. 90,711 unknowns, 722 RHS monostatic incident plane waves. V is 86% sparse while J is 55% sparse.

Figure 9 Open Pipe: Z matrix rank fraction, 20 dB scale ($0.01 < RF < 1.0$), 92,220 unknowns, maximum group size 800, overall sparseness = 95 %

Figure 10 Open Pipe: factored matrix rank fraction, 20 dB scale ($0.01 < RF < 1.0$), 92,220 unknowns, maximum group size 800, overall sparseness = 92 %

Figure 11 Open Pipe: V and J rank fraction (dBrf), 6 GHz, 92,220 Unknowns, V is 96% sparse while J is 85% sparse.

Figure 12 ACA norm approximation errors for 220 x 214 Z block of Figure 3 as a function of k UV rows and columns. Convergence to 10^{-4} by $k = 20$.

Figure 13: SLICY LU operations count, 30 dB scale ($0.001 < RF < 1.0$), 90,711 unknowns, maximum group size 800, $\text{tol} = 10^{-4}$

Figure 14 SLICY VV polarization, MERCURY MOM and CARLOS predictions overlay

Figure 15 SLICY HH polarization, MERCURY MOM and CARLOS prediction overlay

Figure 16 SLICY HV and VH cross polarization, MERCURY MOM and CARLOS predictions overlay

Figure 17 Compressed matrix storage complexity

Figure 18 ACA Matrix fill time complexity

Figure 19 ACA LU factor time complexity

Figure 20 ACA Solve time per RHS complexity

Figure 21 Total run wall time complexity

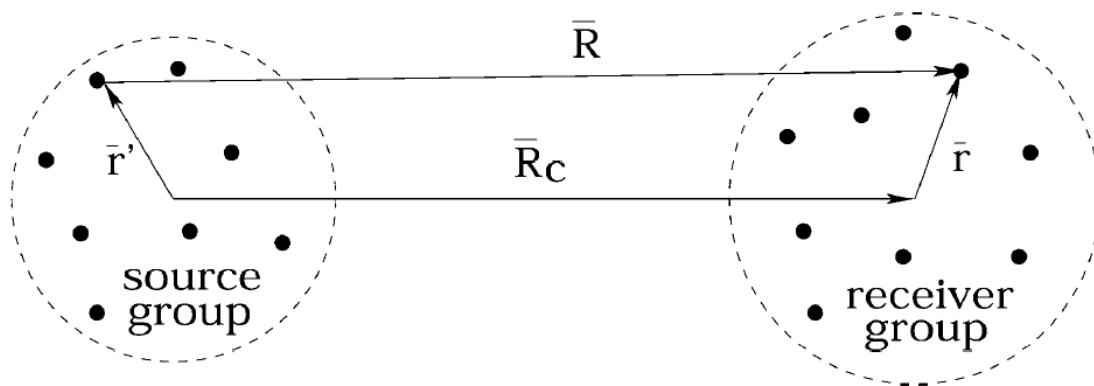


Figure 1. Spatially grouped unknowns lead to low rank block matrices Z , L , U , V , and J for electrically large bodies (from [1]).

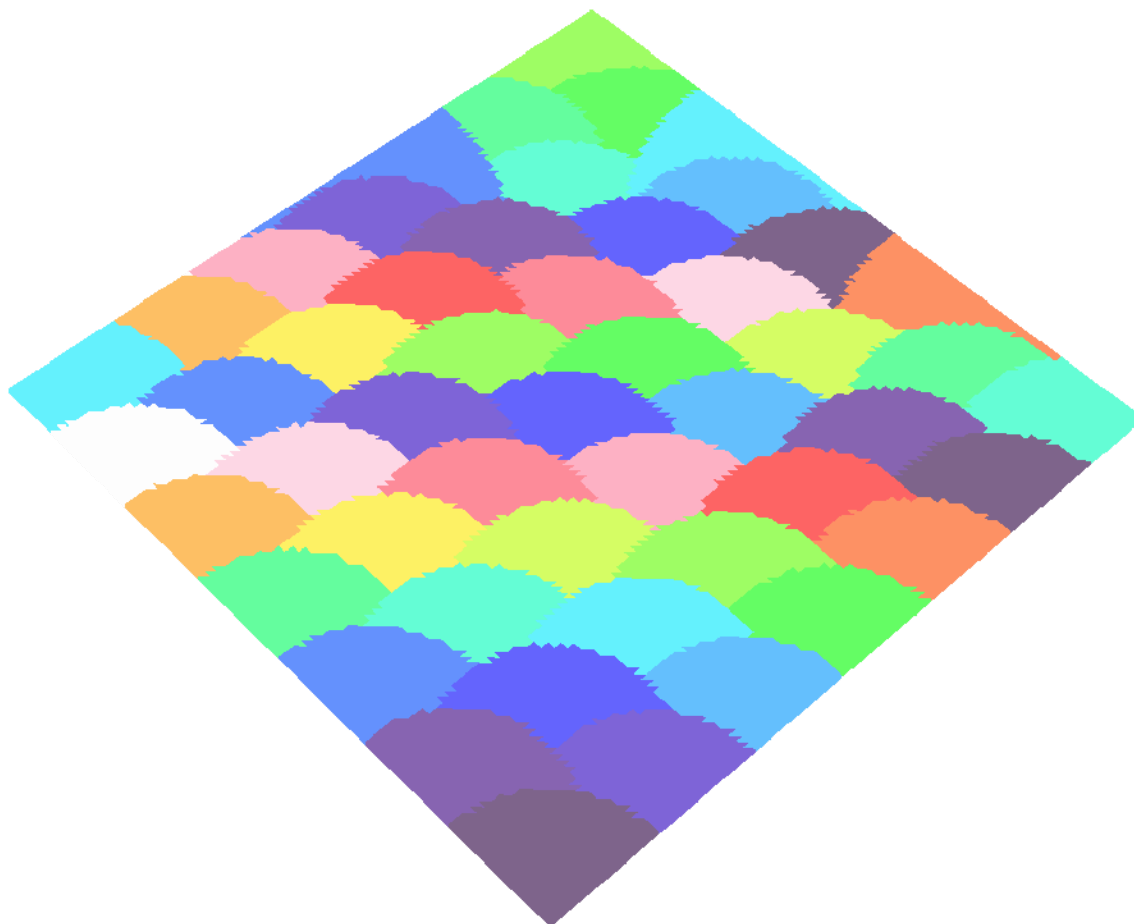


Figure 2. Cobblestone regions on a flat plate geometry: 24933 unknowns, maximum region size 600 unknowns, 51 groups.

Singular Values for a Numerically Low Rank Interaction Matrix

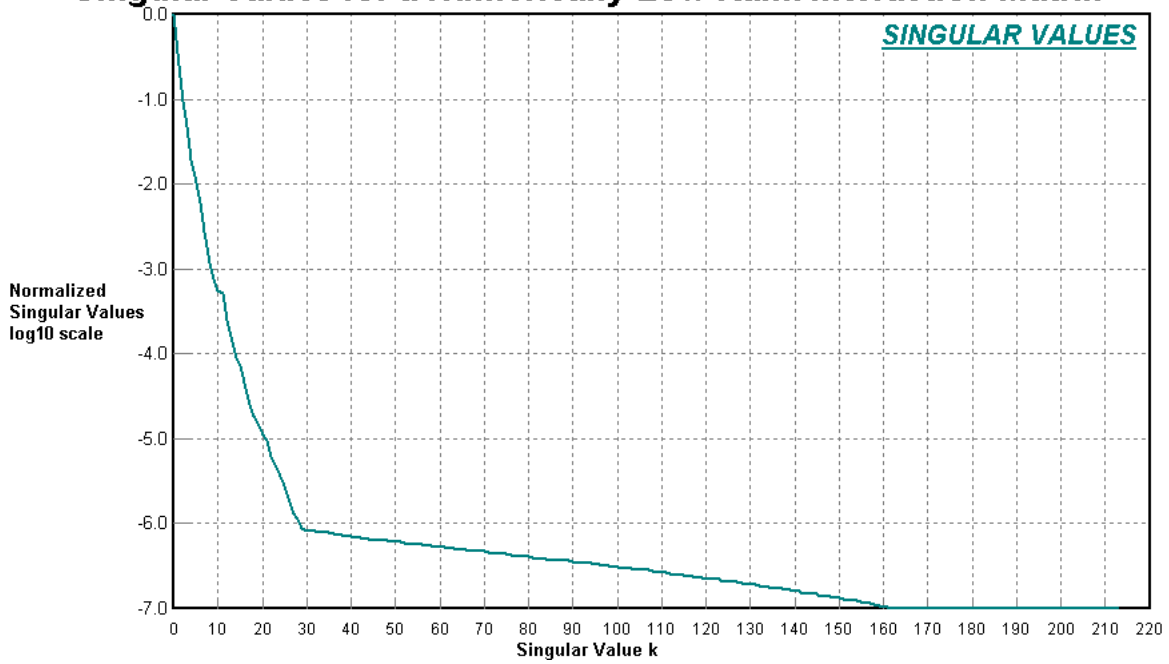


Figure 3. Singular values for a 220 by 214 Z interaction block matrix. Singular values drop to machine precision by $k = 30$.

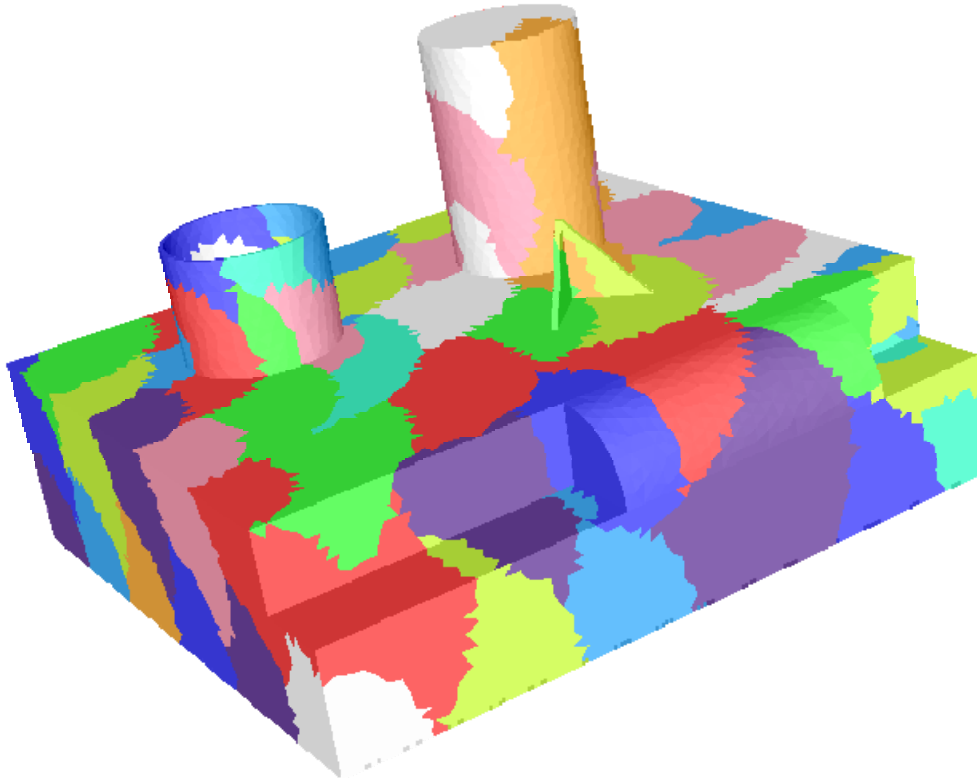


Figure 4. SLICY test geometry showing unknowns grouped into local spatial regions.

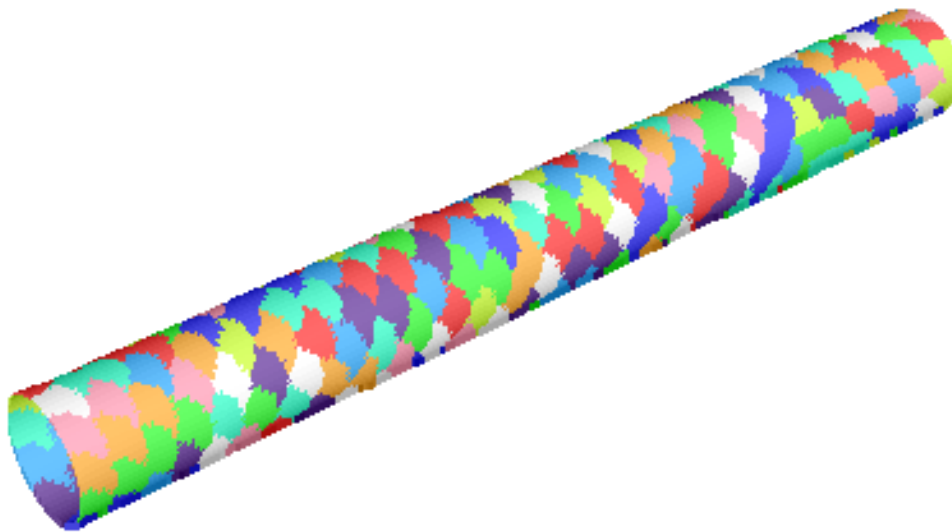


Figure 5. Open pipe geometry showing unknowns grouped into local spatial regions.

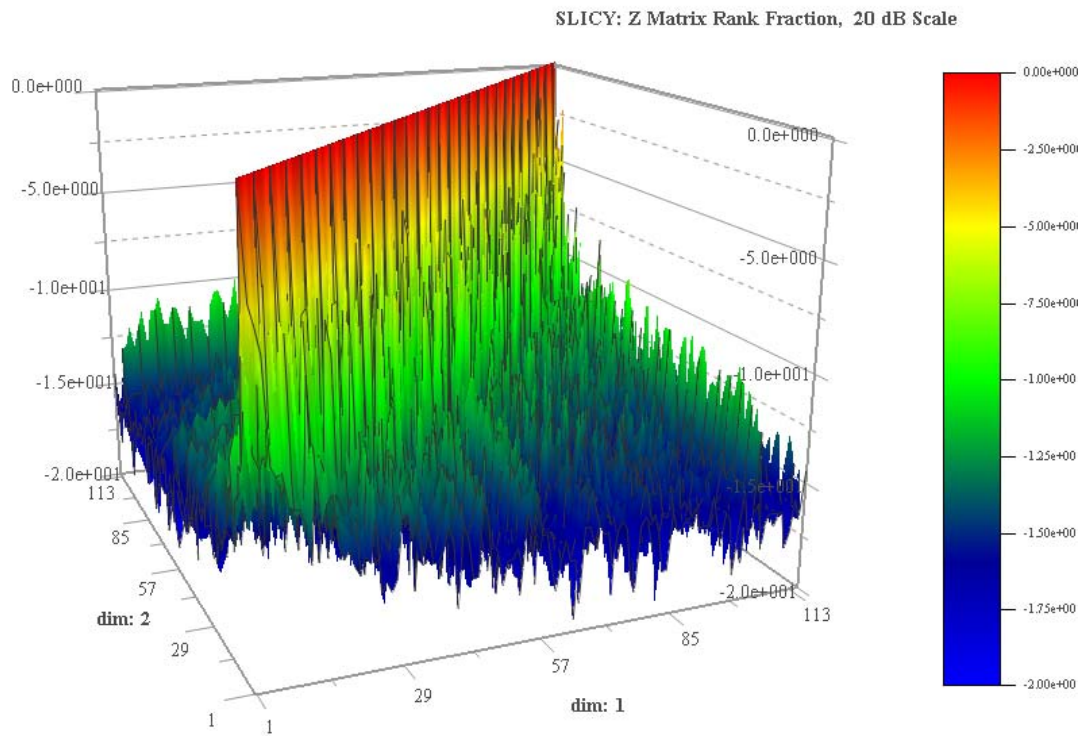


Figure 6. SLICY Z matrix rank fraction, 20 dB scale ($0.01 < RF < 1.0$), 90,711 unknowns, maximum group size 800, overall sparseness = 95 %.

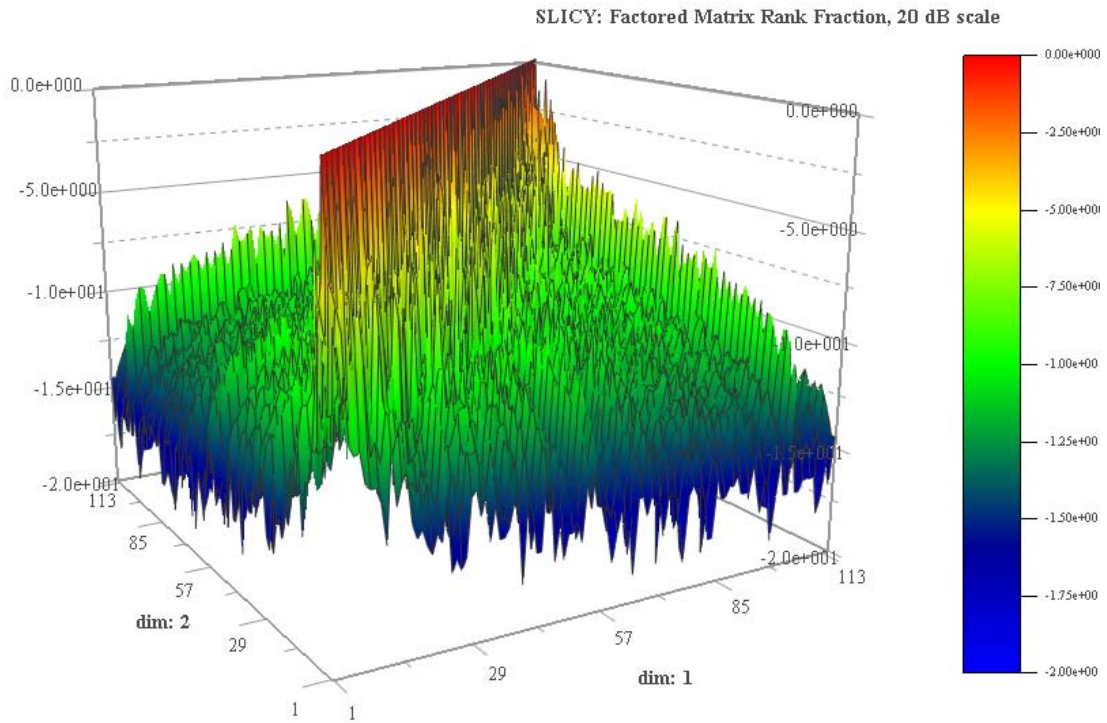


Figure 7. SLICY factored matrix rank fraction, 20 dB scale ($0.01 < RF < 1.0$), 90,711 unknowns, maximum group size 800, overall sparseness = 90 %.

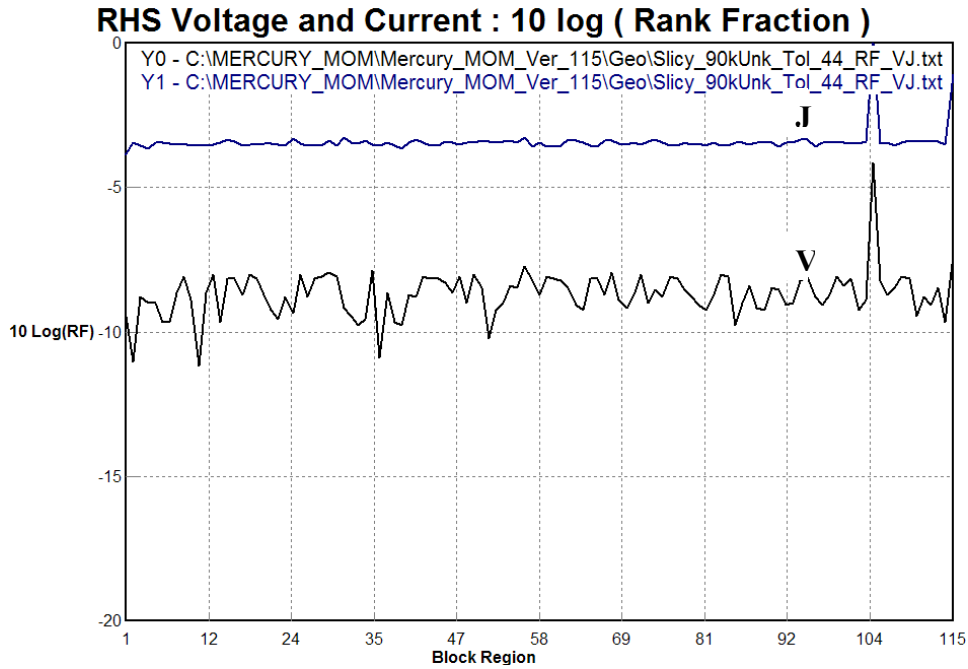


Figure 8. SLICY RHS voltage matrix V(black) and current solution J(blue) rank fraction, 20 dB scale ($0.01 < RF < 1.0$), by region. 90,711 unknowns, 722 RHS monostatic incident plane waves. Overall V is 86% sparse and J is 55% sparse.

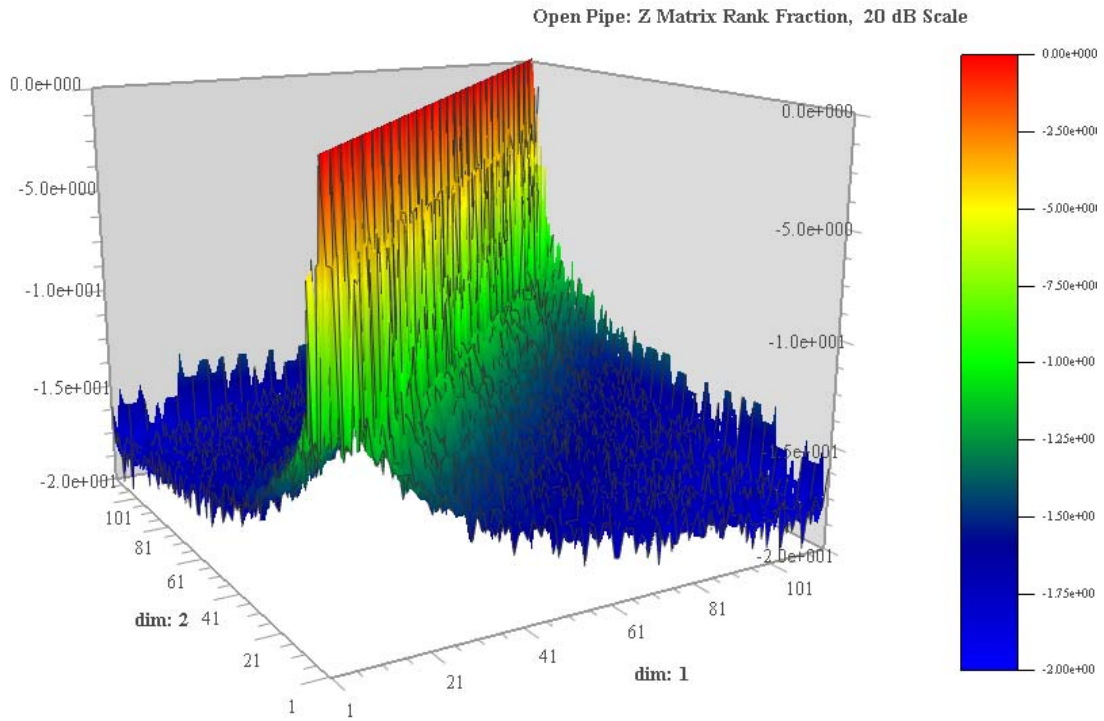


Figure 9. Open Pipe: Z matrix rank fraction, 20 dB scale ($0.01 < \text{RF} < 1.0$), 92,220 unknowns, maximum group size 800, overall sparseness = 95 %.

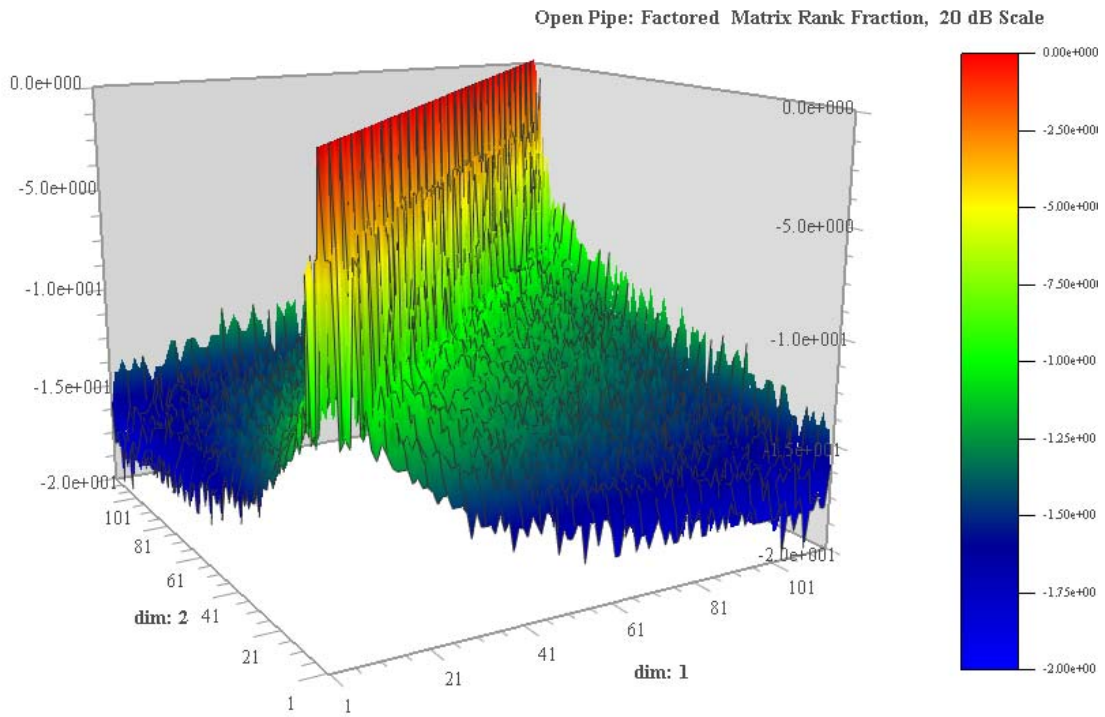


Figure 10. Open Pipe: factored matrix rank fraction, 20 dB scale ($0.01 < RF < 1.0$), 92,220 unknowns, maximum group size 800, overall sparseness = 92 %.

RHS Voltage and Current : 10 log (Rank Fraction)

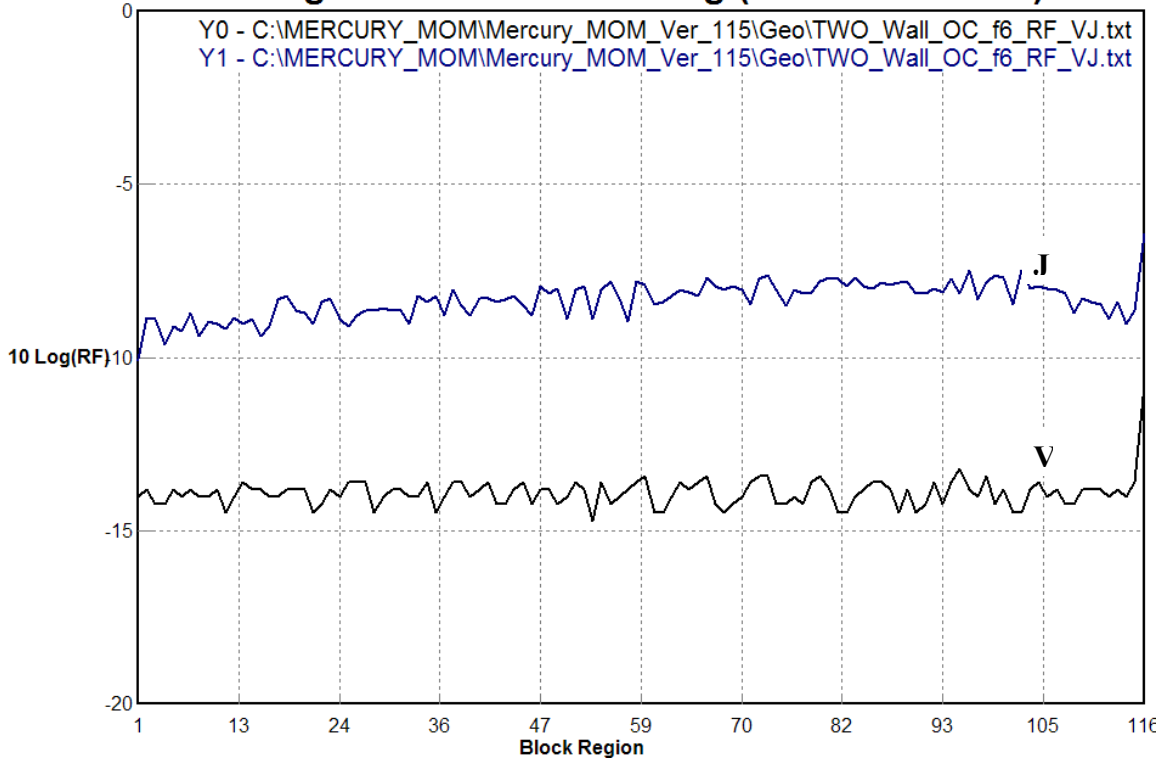


Figure 11. Open Pipe: RHS voltage matrix V(black) and current solution J(blue) rank fraction, 20 dB scale ($0.01 < RF < 1.0$), 6 GHz, 92,220 Unknowns, V is 96% sparse, J is 85% sparse.

ACA Recursive Norm for a Numerically Low Rank Interaction Matrix

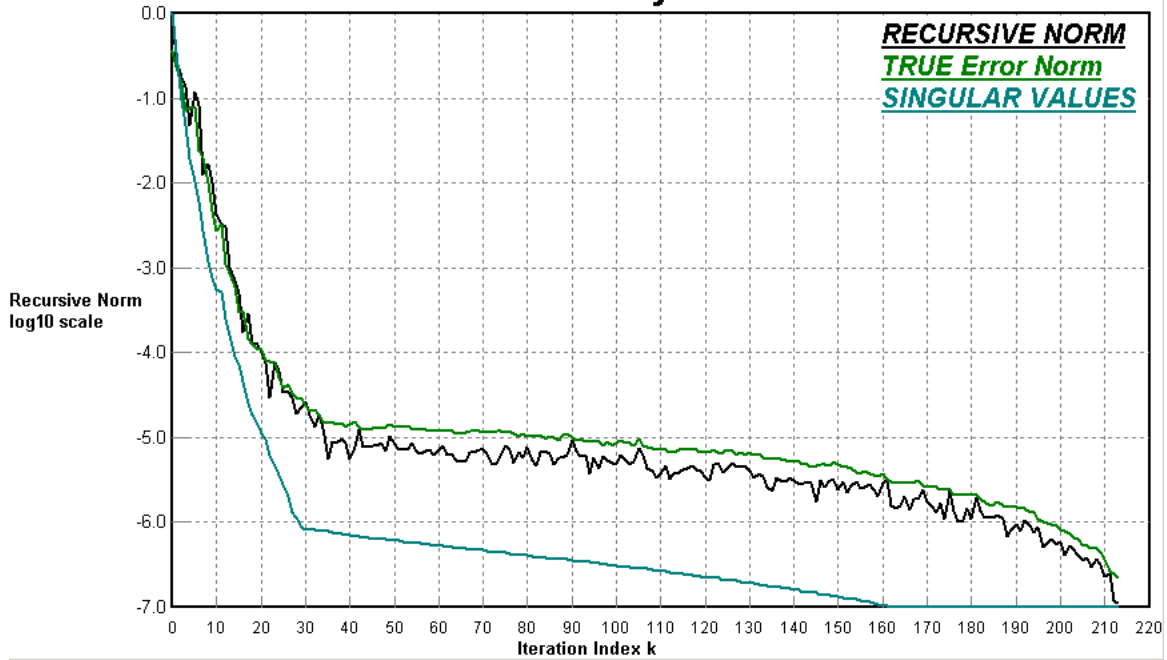


Figure 12. ACA norm approximation errors for 220 x 214 Z block of Figure 3 as a function of k UV rows and columns. Convergence to 10^{-4} by $k = 20$.

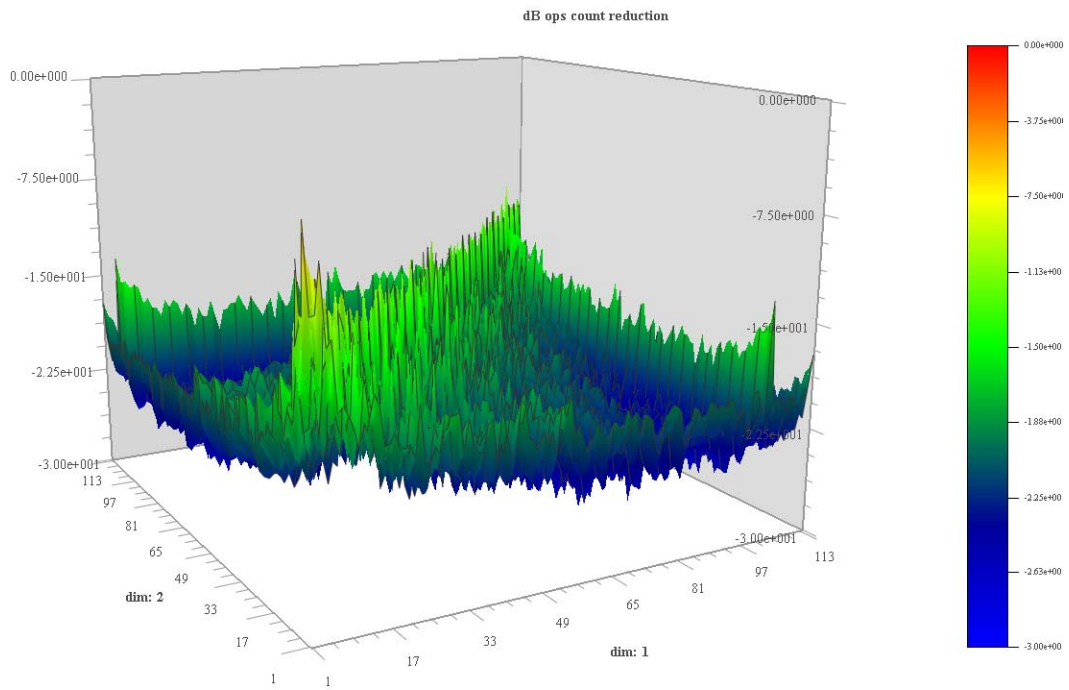


Figure 13. SLICY LU operations count, 30 dB scale ($0.001 < RF < 1.0$), 90,711 unknowns, maximum group size 800, $tol = 10^{-4}$.

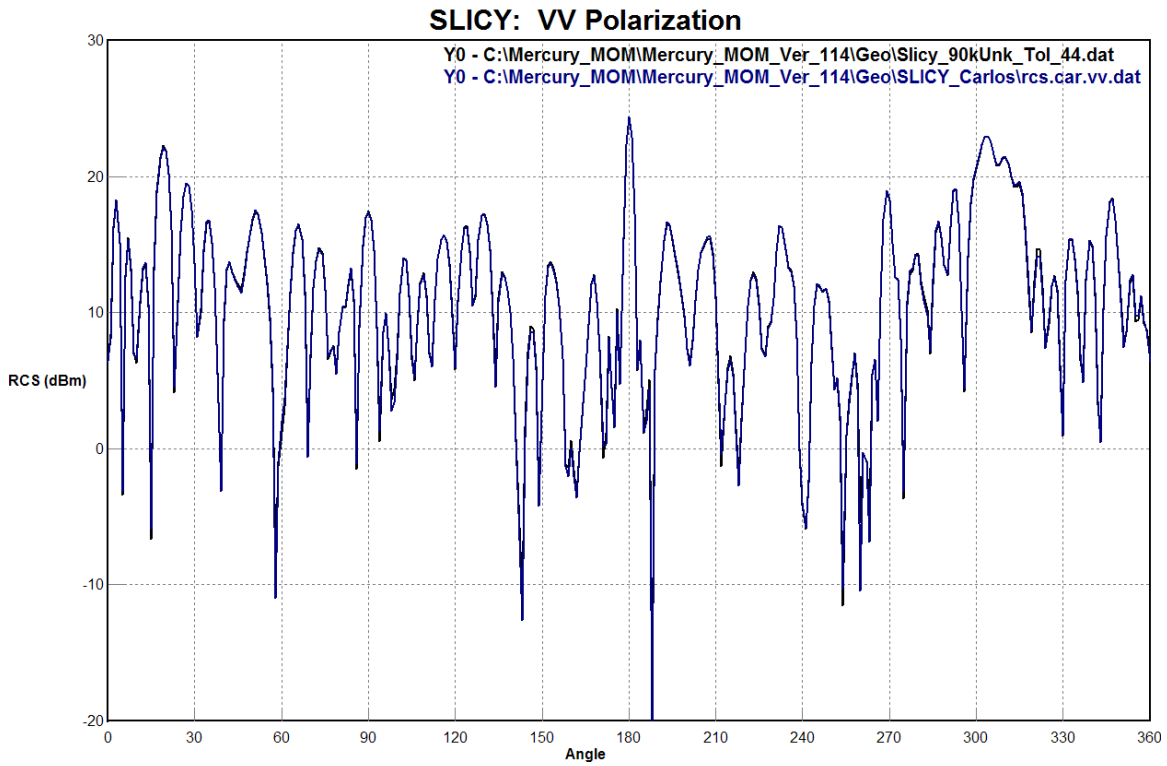


Figure 14. SLICY VV polarization, MERCURY MOM and CARLOS predictions overlay.

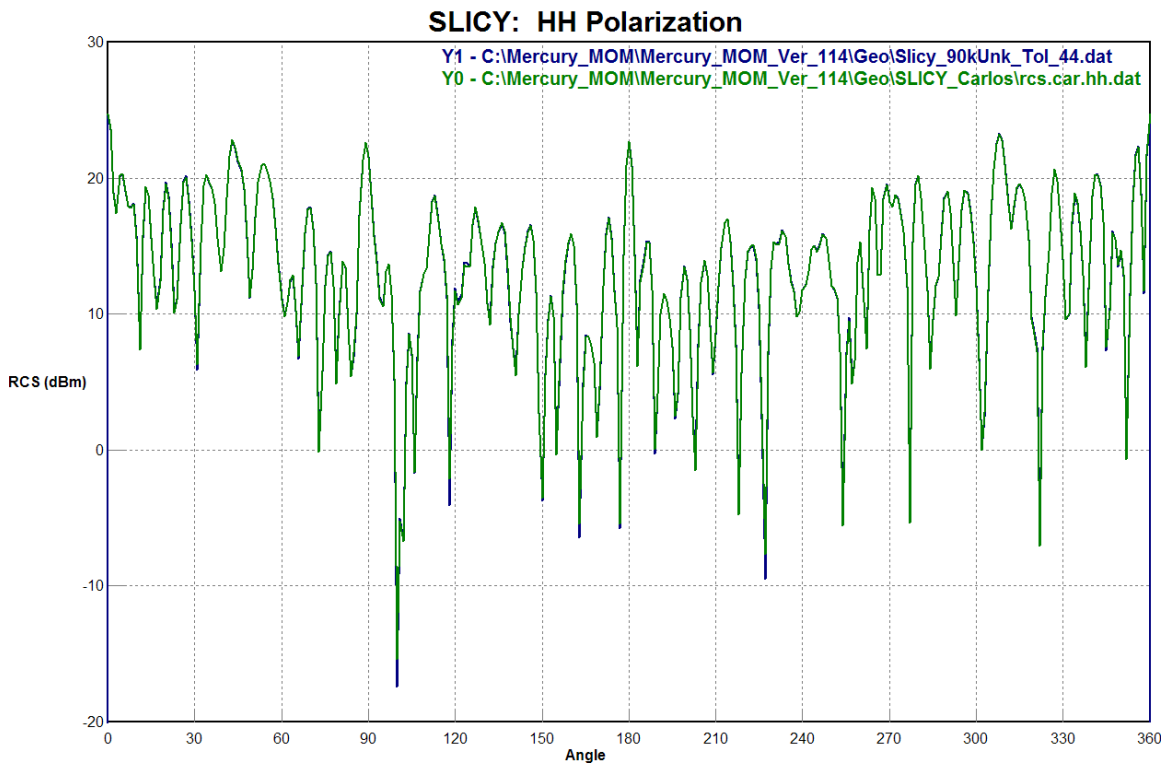


Figure 15. SLICY HH polarization, MERCURY MOM and CARLOS prediction overlay.

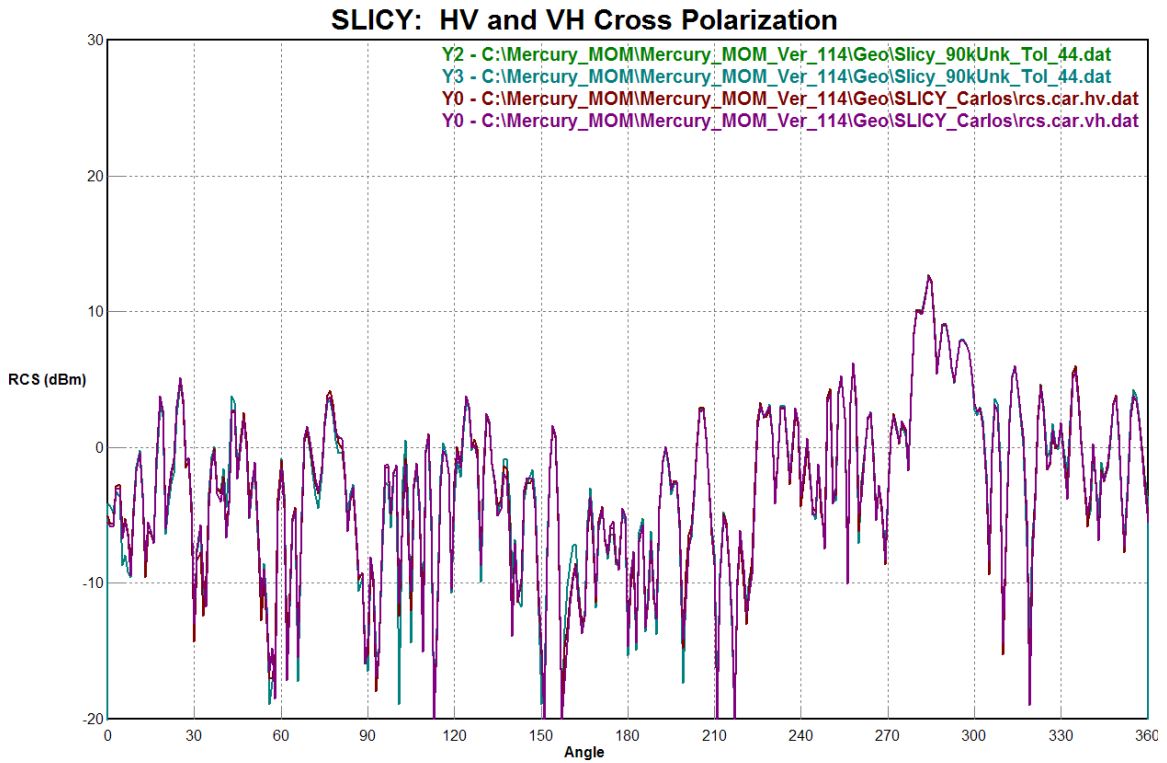


Figure 16. SLICY HV and VH cross polarization, MERCURY MOM and CARLOS predictions overlay.

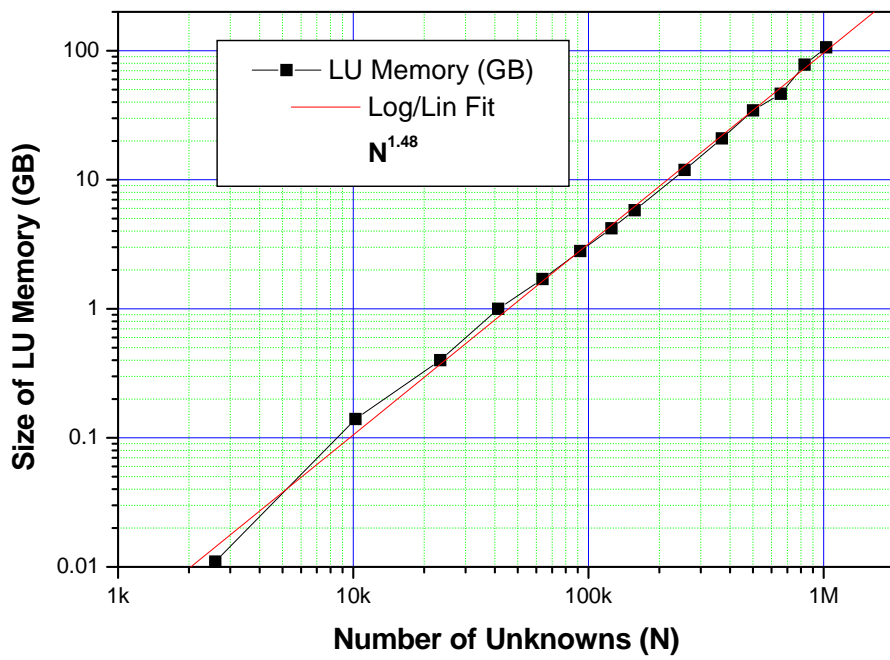


Figure 17. Compressed matrix storage complexity.

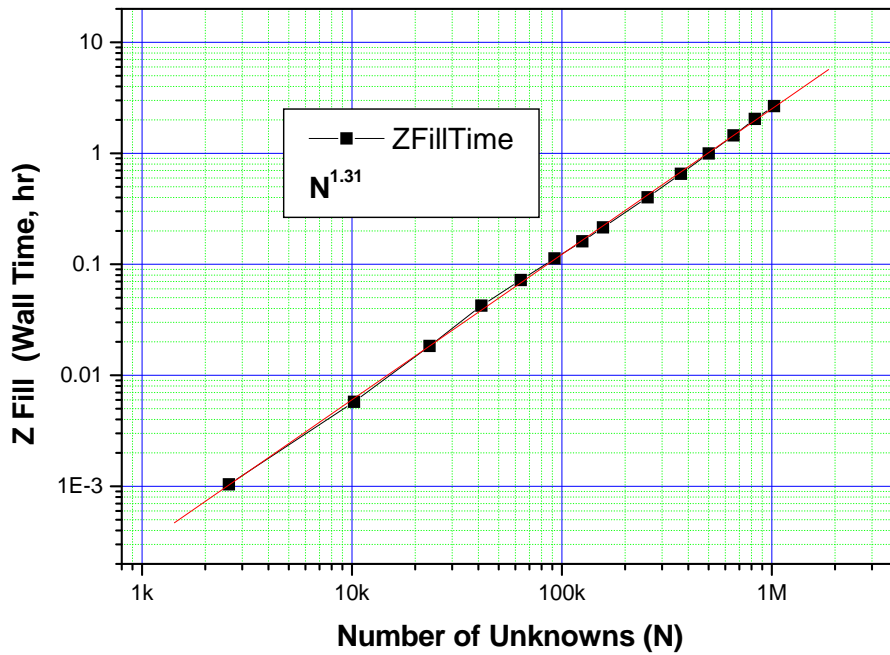


Figure 18. ACA Matrix fill time complexity.

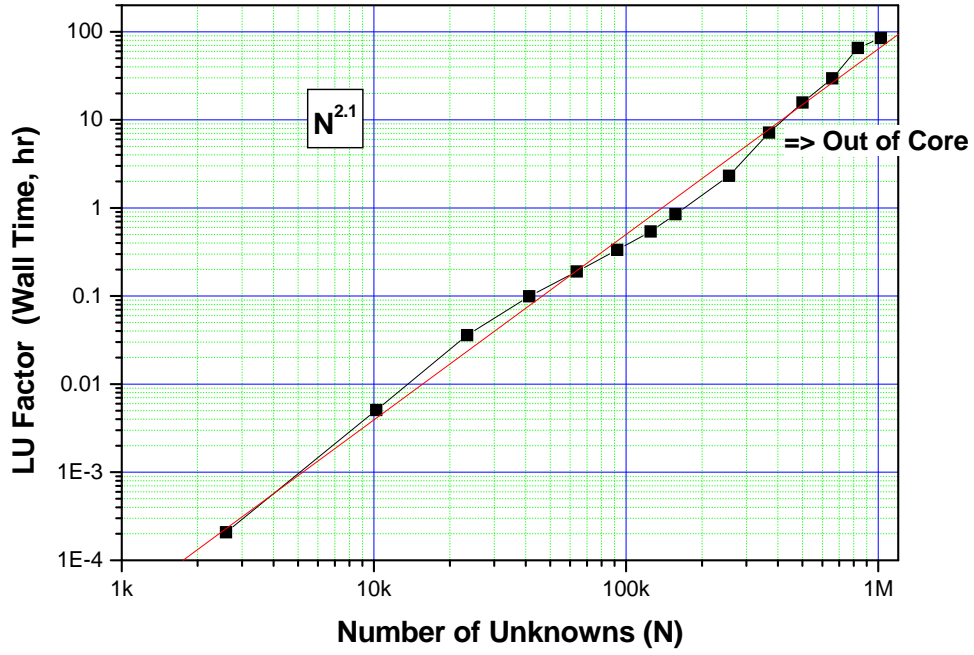


Figure 19. ACA LU factor time complexity.

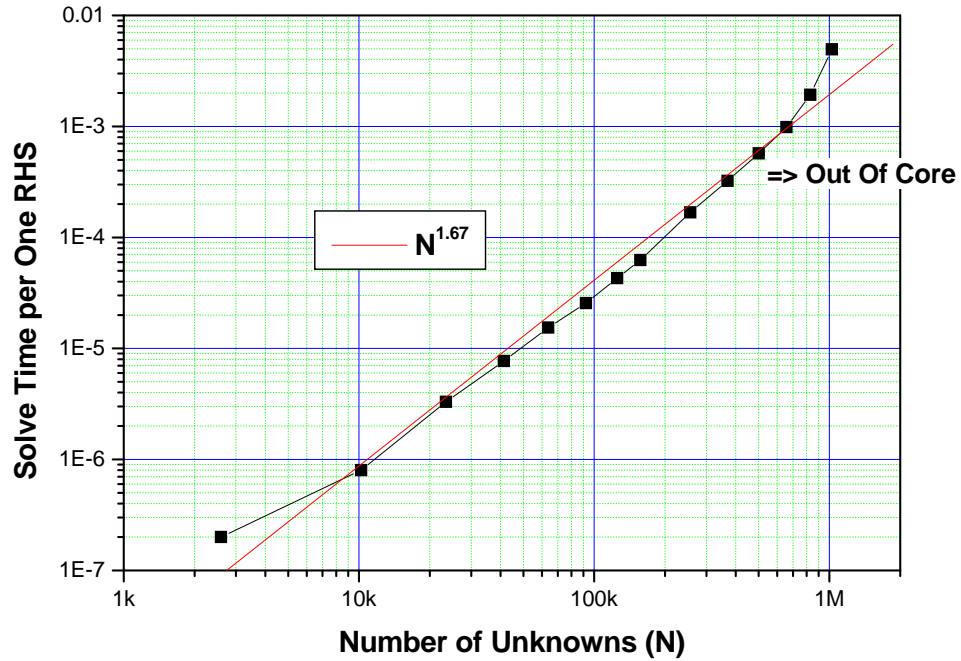


Figure 20. ACA solve time per RHS complexity.

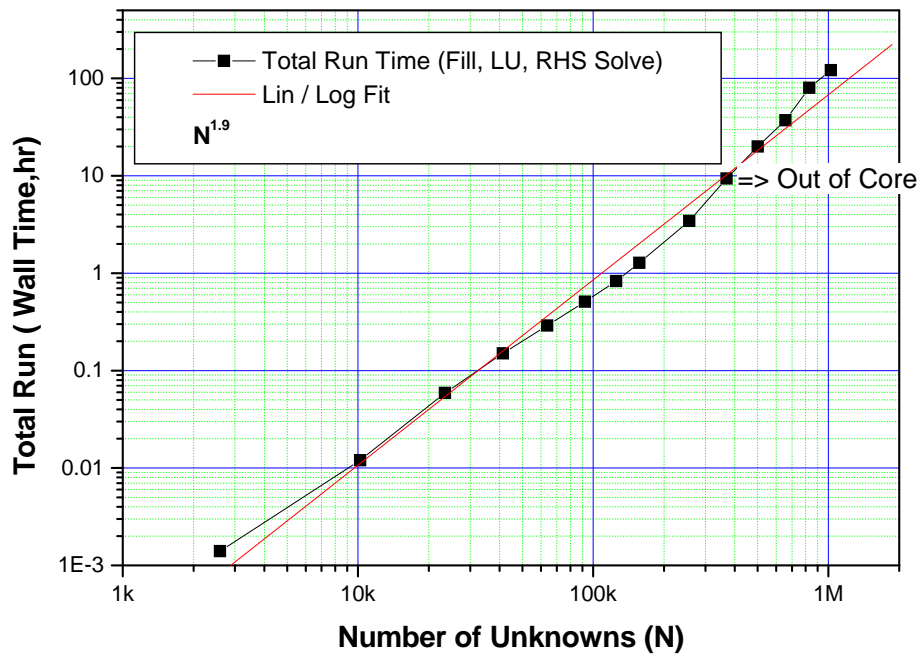


Figure 21. Total run wall time complexity.

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 01-09-2008			2. REPORT TYPE Contractor Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Direct Solve of Electrically Large Internal Equations for Problem Sizes to 1M Unknowns					5a. CONTRACT NUMBER NAS1-02117	
					5b. GRANT NUMBER	
					5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Shaeffer, John					5d. PROJECT NUMBER	
					5e. TASK NUMBER	
					5f. WORK UNIT NUMBER 561581.02.10.07.04	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199					8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001					10. SPONSOR/MONITOR'S ACRONYM(S) NASA	
					11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/CR-2008-215353	
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 33 Availability: NASA CASI (301) 621-0390						
13. SUPPLEMENTARY NOTES Langley Technical Monitor: Kam W. Hom This work was performed by the author as a Consultant working under subcontract to the National Institute of Aerospace, Hampton, Virginia.						
14. ABSTRACT Matrix methods for solving integral equations via direct solve LU factorization are presently limited to weeks to months of very expensive supercomputer time for problems sizes of several hundred thousand unknowns. This report presents matrix LU factor solutions for electromagnetic scattering problems for problem sizes to one million unknowns with thousands of right hand sides that run in mere days on PC level hardware. This EM solution is accomplished by utilizing the numerical low rank nature of spatially blocked unknowns using the Adaptive Cross Approximation for compressing the rank deficient blocks of the system Z matrix, the L and U factors, the right hand side forcing function and the final current solution. This compressed matrix solution is applied to a frequency domain EM solution of Maxwell's equations using standard Method of Moments approach. Compressed matrix storage and operations count leads to orders of magnitude reduction in memory and run time.						
15. SUBJECT TERMS Computational electromagnetics; Matrix compression						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)	
U	U	U	UU	51	19b. TELEPHONE NUMBER (Include area code) (301) 621-0390	