



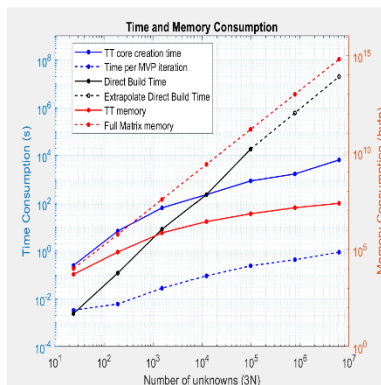
Tensor-Train Accelerated Solution of 3D Vector Volume Integral Equation Solutions with $\log N$ Complexity

Chris Nguyen⁽¹⁾ and Vladimir Okhmatovski⁽¹⁾

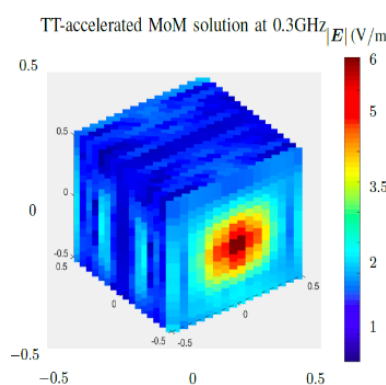
⁽¹⁾ University of Manitoba, Winnipeg, MB, Canada, www.umanitoba.ca

This study explores the use of tensor train decomposition to model the matrix, excitation, and solution vectors of the dense matrix equation resulting in the Method of Moments (MoM) discretization of the full-wave 3D Volume Integral Equation, representing them as a product of $O(\log(N))$ small matrices (tensors). This quantized tensor train (QTT) decomposition illustrates an $O(\log(N))$ efficiency in both CPU time and memory usage. To solve the matrix equation given the QTT-represented system of linear algebraic equation (SLAE) matrix and vectors, we implement an iterative GMRES scheme, enabling fast matrix-vector product evaluations with $O(\log(N))$ CPU time and memory consumption. At present, the $O(\log(N))$ performance is confined to SLAEs with purely Toeplitz matrices, relevant to scattering problems involving homogeneous dielectric scatterers composed of cubic voxels.

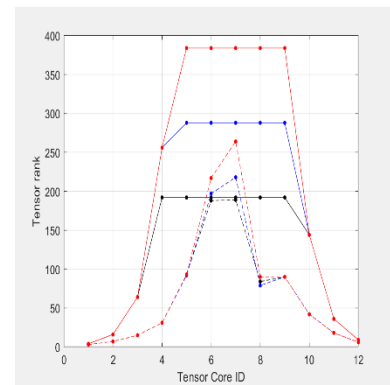
We uniformly partition the cube into N sub-volumes (cells or voxels), which leads to simultaneous equations written in the matrix form as $3N \times 3N$ matrix equation (Z matrix). The implementation incorporates the iterative TT-cross approximation which effectively reduces the cost compared to conventional TT-SVD. As an example for demonstration, consider a cube dielectric scatterer with 1m sides centered at the origin, having relative permittivity $\epsilon_r = 2$ and illuminated by the z-polarized incident plane wave traveling in negative x direction with 1V/m magnitude with the operating frequency is set at 0.3 GHz. We discretize our domain to 8^4 elements (voxels). QTT reduces the memory cost from 2.42 GB to 2.74 MB and the total time to solve from 883.11 to 407.25 seconds with the solution error at $1.07e-4$. The trend of memory and time of the TT approaches $\log(N)$ complexity. One draw back of QTT is the *curse of rank* of which occurs while increasing our frequencies to accommodate the physical nature of problem. One such approach is doing TT for each individual block (9 blocks in total) in our system of $3N \times 3N$ matrix. This significantly reduced our TT ranks. Afterwards, we can apply the recursive Schur complement method for block-structured matrix to find the inverse (in TT format) of our overall system.



(a) Time and Memory consumption



(b) TT solution



(c) TT ranks of compressed Z matrix