Journal of Computational Physics 334 (2017) 145–169

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A Tensor-Train accelerated solver for integral equations in complex geometries

Eduardo Corona [a], Abtin Rahimian [b,*], Denis Zorin [b]

[a] Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, United States
[b] Courant Institute of Mathematical Sciences, New York University, New York, NY 10003, United States

ARTICLE INFO

Article history:
Received 19 November 2015
Received in revised form 1 October 2016
Accepted 27 December 2016
Available online 28 January 2017

Keywords:
Integral equations
Tensor Train decomposition
Preconditioned iterative solver
Complex geometries
Fast multipole methods
Hierarchical matrix compression and inversion

ABSTRACT

We present a framework using the Quantized Tensor Train (QTT) decomposition to accurately and efficiently solve volume and boundary integral equations in three dimensions. We describe how the QTT decomposition can be used as a hierarchical compression and inversion scheme for matrices arising from the discretization of integral equations. For a broad range of problems, computational and storage costs of the inversion scheme are extremely modest $\mathcal{O}(\log N)$ and once the inverse is computed, it can be applied in $\mathcal{O}(N \log N)$.

We analyze the QTT ranks for hierarchically low rank matrices and discuss its relationship to commonly used hierarchical compression techniques such as FMM and HSS. We prove that the QTT ranks are bounded for translation-invariant systems and argue that this behavior extends to non-translation invariant volume and boundary integrals.

For volume integrals, the QTT decomposition provides an efficient direct solver requiring significantly less memory compared to other fast direct solvers. We present results demonstrating the remarkable performance of the QTT-based solver when applied to both translation and non-translation invariant volume integrals in 3D.

For boundary integral equations, we demonstrate that using a QTT decomposition to construct preconditioners for a Krylov subspace method leads to an efficient and robust solver with a small memory footprint. We test the QTT preconditioners in the iterative solution of an exterior elliptic boundary value problem (Laplace) formulated as a boundary integral equation in complex, multiply connected geometries.

1. Introduction

We aim to efficiently and accurately solve equations of the form

$$a\sigma(x) + \int_{\Omega} b(x)K(x, y)c(y)\sigma(y)\, d\Omega_y = f(x), \qquad \text{for all } x \in \Omega, \tag{IE}$$

where Ω is a domain in $\mathbb{R}^3$ (either a boundary or a volume). When $a \neq 0$, the integral equation is Fredholm of the second kind, which is the case for all equations presented in this work. A large class of physics problems formulated as PDEs may be

* Corresponding author.
E-mail addresses: coronae@umich.edu (E. Corona), arahimian@acm.org (A. Rahimian), dzorin@cs.nyu.edu (D. Zorin).

http://dx.doi.org/10.1016/j.jcp.2016.12.051
0021-9991/

cast in this equivalent form. $K(x, y)$ for these type of problems is a *kernel function* derived from the fundamental solution of the PDE. The advantages of integral equation formulations include reducing dimension of the problem from three to two and improved conditioning of the discretization.

The kernel function $K(x, y)$ is typically singular as x approaches y but is smooth otherwise. For the purposes of this paper, we also assume it is not highly oscillatory.

A discretization of Eq. (IE) produces a linear system of equations

$$\mathsf{A}\sigma = \mathsf{f}, \tag{LS}$$

where A is a dense $N \times N$ matrix. Krylov subspace methods such as GMRES [71] coupled with the rapid evaluation algorithms such as FMM [34] are widely used to solve this system of equations. However, the performance of the iterative solver is directly affected by the eigenspectrum of Eq. (IE).

The eigenspectrum of the system, while typically independent of the resolution of the discretization, can vary greatly, depending on the geometric complexity of Ω and the kernel K in particular. When the spectrum is clustered away from zero, the system is solved in a few iteration using a suitable iterative method. However, this is not the case for a number of problems of interest (e.g., when different parts of the boundary approach each other). Such problems may either be solved by constructing an effective preconditioner for the iterative solver or by using *direct solvers*, in which the system is solved in a fixed time independent of the distribution of the spectrum.

There have been a number of efforts to develop robust, fast direct solvers with linear complexity for systems given in Eq. (LS). When Ω is a contour in the plane, extremely efficient $\mathcal{O}(N)$ algorithms such as [55] exist. These algorithms may be extended to volumes in 2D and surfaces in 3D, producing direct solvers with complexity $\mathcal{O}\left(N^{3/2}\right)$ [25,43,24]. More recently, approaches that aim to extend linear complexity to Hierarchically Semi-Separable (HSS) matrices have been developed [18, 42]. Furthermore, a general inverse algorithm has been proposed for FMM matrices [2].

For 2D problems, these types of methods have excellent performance and remain practical even at high target accuracies, e.g., 10^{-10}. However, for volume or boundary integral equations in 3D, especially in complex geometries, algorithmic constants in the complexity of these methods grow considerably as a function of accuracy. In particular, the compressed form of the inverse typically requires a very large amount of storage per degree of freedom, limiting the range of practical target accuracies or problem sizes that one can solve. This also precludes efficient parallel implementation, due to the need to store and communicate large amounts of data.

Basic iterative solvers (requiring only matrix–vector multiplication) and direct solvers, represent two extremes of the spectrum of *preconditioned iterative solvers*, as a direct solver can be viewed as a preconditioned solver with a perfect preconditioner requiring one iteration to converge. These also represent two extremes in the tradeoff between memory and time spent for computing the preconditioner as well as the cost of each solve. A direct solver requires a lot of memory and precomputation time for the inverse matrix, but solving a system for a specific right-hand side is typically very fast. On the other extreme, a non-preconditioned iterative solver requires only a matrix–vector multiplication function, either requiring no precomputation, or compression of the matrix (but not computing its inverse). However, each solve in this case may require a large number of iterations.

Varying the accuracy of the approximate inverse preconditioner leads to intermediate solvers, with reduced storage and time required for precomputation, but increased time needed for solves. By varying this accuracy, we can find a "sweet spot" for a particular type of problem and let the practitioner strike a reasonable trade-off between precomputation time and solve time, within the available memory budget.

1.1. Contributions and outline

We present an effective and memory-efficient preconditioned solver for Eq. (LS) based on the quantized tensor train decomposition (QTT) [64,50]. The QTT decomposition allows for the compact representation and fast arithmetic of structured matrices by recasting them in tensor form.

In this work, we frame this process in the context of hierarchical compression and inversion for matrix A. We show that for a range of target accuracies, QTT decomposition achieves significantly higher compression by finding a common basis for all interactions at a given level of the hierarchy. We argue this makes it a natural fit for the solution of integral equations, and discuss how, for certain problems, it can achieve superior performance versus commonly used hierarchical compression techniques such as FMM and variants of $\mathcal{H}$-matrices.

We prove rank bounds for the QTT ranks of A for translation-invariant systems, as well as for a family of non-translation invariant systems obtained from volume integral equations in Section 3, and provide evidence that this behavior extends to boundary integrals in complex geometries. In our experiments in Section 5, we find that the inverse A^{-1} is also highly compressible, displaying comparable rank behavior to that of A in all cases considered.

In our presentation of the QTT inversion process in Section 4, we contribute novel strategies to precondition the QTT inversion algorithms, representing the inverse as a product of matrix factors in the QTT form to achieve faster computation.

Finally, we perform an extensive series of numerical experiments to evaluate the performance and experimental scaling of the QTT approach for volume and boundary integral equations of interest. In both instances, we confirm key features of QTT that distinguish it from existing fast direct solver techniques: