Data-sparse matrix computations
Lecture 7: Finishing Rank Structured Matrices, Beginning Krylov Subspace Methods

Lecturer: Anil Damle
Scribers: Lijun Ding, Mike Sosa

September 12th, 2017

1 Rank Structred Matrices

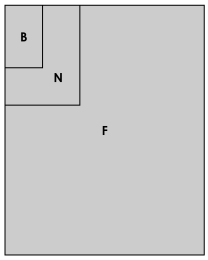
1.1 Recap

Last time, we showed the first step of how to factorized a matrix with entries representing some physical quantity on a plane into a block diagonal form.

More specifically, suppose we have a set of points $\{x_i \in \mathbb{R}^2\}_{i=1,\dots,n} \subset \Omega$ and a matrix K with (i,j)th entry to be

$$K_{ij} = K(x_i, x_j).$$

for some function $K : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ which is symmetric, i.e., $K(x,y) = K(y,x)$ and can be approximated well using Taylor expansion. The points $\{x_i \in \mathbb{R}^2\}_{i=1,\dots,n}$ are assumed to live uniformly in the domain. We first divide the region Ω into three parts. Like shown in the following graph.



Here B is the set points that we are currently interested in. N are neighbor points and F are points in the far field. We will also use B, N, F to refer to the indices of the corresponding points. Then there is a permutation matrix P such that

$$P^* K P = \begin{bmatrix} K_{BB} & K_{BN} & K_{BF} \\ K_{NB} & K_{NN} & K_{NF} \\ K_{FB} & K_{FN} & K_{FF}. \end{bmatrix}$$

By previous lectures, we know K_{FB} is approximately low rank and we can use interpolative decomposition on this matrix to obtain

$$K_{FB}\Pi \approx K_{FS}[IT]$$

for some $S \subset B$ which is called skeleton, some permutation matrix Π and some matrix T. We denote $R = B - S$ to be the set of redundant points.

Then we can show that there are upper triangular matrices U_T, U and lower triangular matrices L, L_T and permutation matrix P such that

$$L U_T P^* K P L_T U \approx \begin{bmatrix} X_{RR} & & & \\ & X_{SS} & X_{SN} & X_{SF} \\ & X_{NS} & X_{NN} & K_{NF} \\ & K_{FS} & K_{FN} & K_{FF} \end{bmatrix} =: Z(K, B).$$

The accuracy of approximation is at the level of accuracy using ID.

1.2 A Recursive Skeletonization Factorization

We define

$$V_B = PU_T^{-1}L^{-1}$$

and

$$W_B = U^{-1}L_T^{-1}P^*.$$

Then

$$Z(K, B) = V_B^{-1}KW_B^{-1}.$$

In the following we will have V, W with other subscripts to mean similar operations performing on the set indicated in the subscripts.

We now start describing the procedure of factorizing K into a block diagonal form where each block will be of small size, e.g., 64×64 or 128×128.

Assume we have a quad-tree on Ω with levels $0 \to L$. The levels is determined so no leaf level box has more than some fixed number of points.

1. We will first specify what we should do on the leaf level.

   (a) Take a leaf box, say Ω_l with index I_l. Use the procedure in recap or last lecture to compute $Z(K, I_l)$. Let N_l be neighbor points that touches Ω_l and F_l as everything else.

   (b) Select the next leaf box $I_{l'}$, where $I_{l'}$ is some other box at the leaf level. Define $N_{l'}$ as "active points" in neighbor boxes and $F_{l'}$ as "active" points in far field. A point is active if it has never been redundant. If we go back to last lecture or recap, the skeleton set S is still active while R is redundant.

   It should be noted that $Z(K, I_l)_{F_{l'}, I_{l'}}$ might not be just $K(F_{l'}, I_{l'})$. This happens if $I_{l'} \subset N_l$ and $F_{l'} \cap N_l \neq \emptyset$, a.k.a that $I_{l'}$ is in the near field. Thus we don't have much reason to say $Z(K, I_l)_{F_{l'}, I_{l'}}$ is low rank. But it turns out to be fine.

   What we get after step 2 is

$$V_{l'}^{-1}Z(K, I_l)W_{l'}^{-1} \approx Z(Z(K; I_l); I_{l'}) := Z(K, I_l, I_{l'})$$

   The matrix $Z(K, I_l, I_{l'})$ looks like

$$Z(K, I_l, I_{l'}) = \begin{bmatrix} X_{R_l R_l} & & \\ & X_{R_{l'} R_{l'}} & \\ & & X \end{bmatrix}.$$

   Here $R_l, R_{l'}$ are "inactive" and redundant. With quad trees these boxes are disjoint as well.

2. We now define how to proceed to go up the tree.

   (a) We loop over all boxes at leaf level according to previous steps. Let's say we have $Z(K, I_1, \ldots, I_m)$ where m are boxed on the leaf level. After this procedure. We get a block diagonal matrix with

$$Z(K, I_1, \ldots, I_m) = \begin{bmatrix} X_{R_l R_l} & & & \\ & X_{R_2 R_2} & & \\ & & \ddots & \\ & & & X \end{bmatrix}.$$

   There is one large matrix X left after we have done the leaf level and is what we need to deal with at a higher level of the hierarchical tree.

   (b) We take all boxes at level $L - 1$. Say Ω_p define

$$I_{\Omega_p} = \cup_{\sigma \text{ s.t. } \Omega_\sigma \text{ is a children of } \Omega_p} S_\sigma$$

(c) We can then do the same procedure as the leaf level.

3. We then loop over all levels $L, L-1, \ldots, 1$. Note that for skeleton sets at a certain level which are not neighbors or neighbor of a common box, then the entries with indeces cross the boxes are always the corresponding entries in K. For example, suppose at level level we have B_1, B_2 and skeleton set S_1, S_2, as long as B_1 and B_2 are not neighbors or not both are neighbors of a third box B_3, then the entries after we multiply V and Ws are still $K_{S_1 S_2}$. See [2, Figure 5] for more detail explanation on this.

Define an order of boxes such that if $l \leq l'$, l is on the same or lower level of the tree. Let M be the total boxes not including the root.Then we get

$$D = \begin{bmatrix} X_{R_1 R_1} & & & \\ & \ddots & & \\ & & X_{R_M R_M} & \\ & & & X_{I_0 I_0} \end{bmatrix} \approx P^*(\Pi_{l=[M]'} V_i^{-1}) K (\Pi_{l \in [M]} W_i^{-1}) P$$

and so

$$F = (\Pi_{l=[M]} V_i) P K P^* (\Pi_{l \in [M]'} W_i) \approx K.$$

The multiplication $\Pi_{[M]} V_i$ means multiply according to order $V_1 \ldots V_M$ and $\Pi_{[M]'} V_i^{-1}$ means the reverse order $V_M^{-1} \ldots V_1^{-1}$. The number M is at the order $\mathcal{O}(N)$.

The final matrix F can be called the skeletonization factorization of K. It is "easy" to factor/solve systems and $F^{-1} = K^{-1}$ and $F^{-1} \approx K^{-1}$ since each submatrices should be easy to factorize or solve and there are $\mathcal{O}(N)$ of them.

Finally, if $|S_l| = \mathcal{O}((L - l + 1)^q$ where l is the levels of boxes and $|S_l| = \mathcal{O}(\log^q(N))$ at level $L - 2$. Then the algorithm is $O(N)$ for factorizing, solve and applies the matrix to a vector. See [2, Theorem 3.3, Corollary 3.4] for more detail explanation on this.

2 Krylov Subspace Methods

2.1 Introduction

Alexei Krylov, was a Russian mathematician among many other things. After serving in the Russian Navy, Krylov began his work as an applied mathematician. His work spanned a wide variety of fields from maritime engineering, differential equations, physics, and most importantly numerical analysis. In 1931, Krylov published a paper on what we call a "Krylov Subspace"[1]. The paper dealt with eigenvalue problems and the computation of the characteristic polynomial coefficiants of a matrix. Limited in his time due to the lack of computation power, Krylov's work lay unnoticed until it became valuable for modern iterative methods for finding eigenvalues, or solving linear systems. Today many Krylov Subspace methods exist, including GMRES, Arnoldi, Lanczos, and BiCGSTAB.

2.2 The Problem

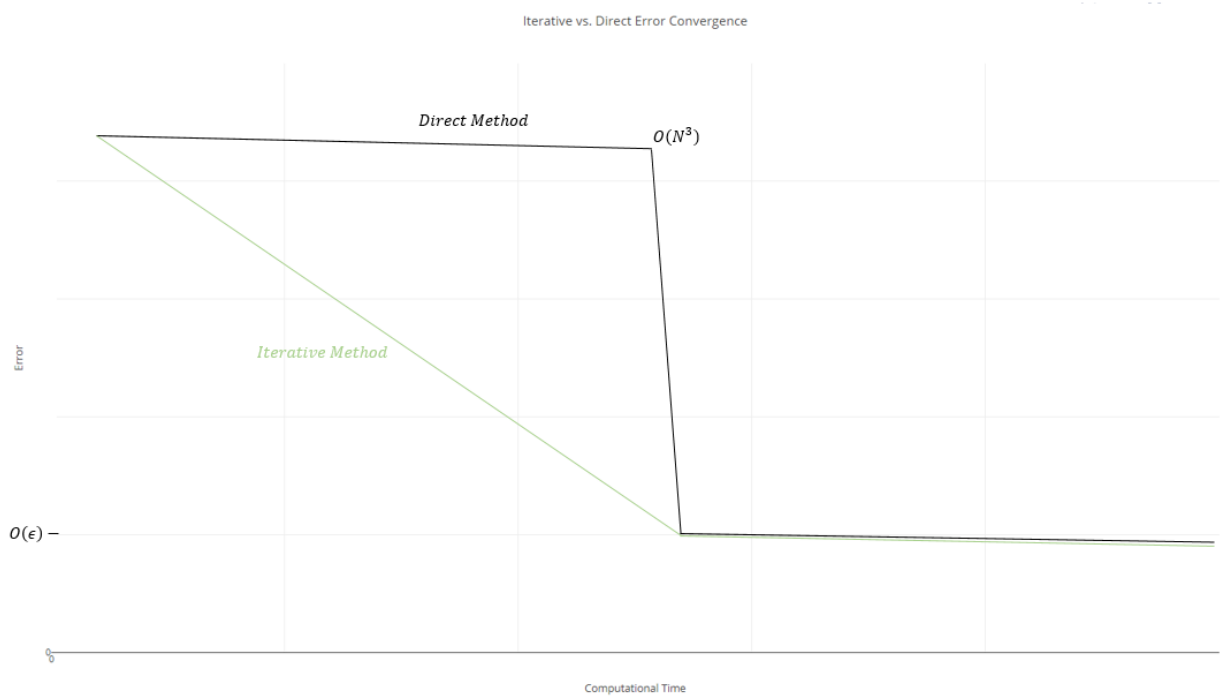
Assume we have the following.

1. A black box such that inputting a vector x we are given back Ax

We have no idea what A is, only that some process inside this black box computes Ax and returns it to us. Given this, the problem at hand is, can we solve $Ax = b$ for a vector b? Additionally are we able to find eigenvalues/eigenvectors of A? Using Krylov subspace methods, the answer to these two questions is Yes.

2.3 Iterative methods

Given A,b to solve $Ax = b$ we can cook up a sequence of $x^k \to X$ as $k \to \infty$. The method by which we do this is iterative, as opposed to a direct method. The difference between the two can be seen in the picture below.



The convergence for direct methods is non-existant up until a certain point at which the method converges to machine epsilon precision denoted $O(\epsilon)$. Iterative methods on the other hand converge, but at a much slower rate. The shape of the line of convergence is dependent on the problem at hand.

2.4 The Solution

The solution to this problem is to use Krylov Subspaces. We can look for the solution x^k which lies in the kth krylov subspace, denoted $x^k \in \mathscr{K}_k(A, b) \equiv span\{b, Ab, A^2b, \ldots, A^{k-1}b\}$. More details will be covered in the writeup of the next lecture.

References

[1] Mike Botchev. A.n. krylov: a short biography. `http://ta.twi.tudelft.nl/users/vuik/burgers/krylov.html`.

[2] Victor Minden, Kenneth L Ho, Anil Damle, and Lexing Ying. A recursive skeletonization factorization based on strong admissibility. *Multiscale Modeling & Simulation*, 15(2):768–796, 2017.