A geometric cross approximation method

Jianming Zhang & Xingshuai Zheng
State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, China

Abstract

The matrices resulting from the discretisation of non-local operators occurring in the boundary element method (BEM) are fully populated and require special compression techniques for efficient treatment. Among the several standard techniques, the $\mathcal{H}$-matrix representation is used to approximate the dense stiffness matrix in admissible blocks which can be approximated by low-rank matrices. This paper presents a geometric cross approximation (GCA) algorithm to assemble the low-rank matrices. Compared with the adaptive cross approximation (ACA), the GCA determines the skeleton points from the two interacting groups of nodes corresponding to an admissible block by their topological spatial relations directly and, thus, has a remarkable non-iterative nature and requires the spatial geometric information of the nodes, only. Numerical examples are presented to further demonstrate the feasibility and effectivity.
Keywords: $\mathcal{H}$-matrix, dense matrix, low-rank matrices, adaptive cross approximation, skeleton points, topological spatial relations.

1   Introduction

In the applied and engineering sciences, boundary element method (BEM) plays an important role, especially in the problems with infinite domain. However, the resulting matrices are in general dense and require computer memory consumption scale quadratically with respect to the degrees of freedom. The efficient treatment of dense matrices requires special compression techniques to reduce the storage requirement and speed up the arithmetic (e.g., inversion).

Several fast solutions of boundary integral equation have been developed in the last two decades. Well known are panel clustering [1, 2], multipole expansions [3–8], interpolation [9] and (adaptive) cross approximation [10–13].

Even the wavelet technique can also be used to compress the resulting dense matrix, when the underlying geometry can be described by a few of smooth maps [14].

The adaptive cross approximation (ACA) algorithm is proposed by Bebendorf [10] and Bebendorf and Rjasanov [11]. As can be seen in Eq. (1), this method takes the single-variable function which is the result of fixing some source points or field points of the kernel function as the basis of the interpolation space. The selected points are called skeleton points. Hence, the low rank approximation can be assembled when we have found an optimal set of the skeleton points. Obviously, the optimal set of the skeleton points is not unique.

$$\kappa(P,Q) \approx \{\kappa(P,Q_1) \quad \dots \quad \kappa(P,Q_k)\} \begin{bmatrix} \kappa(P_1,Q_1) & \cdots & \kappa(P_1,Q_k) \\ \vdots & & \vdots \\ \kappa(P_k,Q_1) & \cdots & \kappa(P_k,Q_k) \end{bmatrix}^{-1} \begin{Bmatrix} \kappa(P_1,Q) \\ \vdots \\ \kappa(P_k,Q) \end{Bmatrix} \qquad (1)$$

This algorithm determines the skeleton points similarly to Gram-Schmidt orthogonalization. It uses only entries from the original matrix for the approximation of each block. This algorithm can be regarded as the algebraic counterparts of panel clustering and fast multipole methods. There is no need for the expansion of the kernel function. Compared with the FMM, the ACA is the easier parallelization of the algorithm [15] and much simpler to be implemented. Meanwhile, as an advantage over the FMM, an approximation of the global matrix can be used for preconditioning besides the near-field matrix [16]. The hybrid cross approximation (HCA) which combines the ACA algorithm and the interpolation-based separation of the kernel function is proposed by Börm and Grasedyck [13]. Likewise, an algorithm which determines the skeleton points using random sampling technique is proposed by Martinsson and Rokhlin [17].

Our contribution is a new non-iterative method that implements the low rank approximation of each admissible block in the original matrix of BEM based on the geometric topological spatial relations. It is well known that the matrices of BEM are derived from the boundary integrals of the kernel functions. Compared with the constant element which can be directly performed analytical integration, it is not to avoid using numerical integration techniques when the higher order elements are used. Thus, it needs to perform the numerical integrations which correspond to the entries of the selected column or row in the matrices of the BEM in each iteration procedure of the ACA. The goal of the proposed method is separating the procedure of determining the skeleton points from the integral operation of the kernel function. The skeleton points are determined in advance from the perspective of the geometric characteristics. Then, the boundary integrals which correspond to the skeleton points in the far-field can be computed together with those for the near-field interaction. Integrals for both near-field and far-field interactions are computed in a unified framework in the same way as the traditional BEM. It will be convenient and effective to use higher order elements in the simulation.

The rest of this article is organized as follows: in Section 2, we introduce a simple model problem and describe the $\mathcal{H}$-matrix format and the low rank approximation in short. In Section 3, we introduce the geometric cross approximation algorithm and provide numerical examples that show this

algorithm is effective in Section 4. Finally, the paper ends with conclusions in Section 5.

2 The $\mathcal{H}$-matrix format

2.1 Model problem: integral equation

Large dense matrices are usually derived from the BEM without additional structure. These matrices arise after the discretisation of the boundary integral equation

$$A[u](x) = \int_{\Gamma} K(x,y)u(y)d\Gamma(y) \tag{2}$$

where $\Gamma \subset R^3$ is the boundary of the computational domain and $K : \Gamma \times \Gamma \to R$ is the kernel function. The kernel function might be the single or double layer kernel for the Laplacian:

$$g^{SLP}(x,y) := \frac{1}{4\pi\|x-y\|}, \qquad g^{DLP}(x,y) := \frac{\langle x-y, n(x)\rangle}{4\pi\|x-y\|^3} = \frac{\partial g^{SLP}(x,y)}{\partial n(x)}$$

The kernel functions $g^{SLP}(x,y)$ and $g^{DLP}(x,y)$ are asymptotically smooth. In fact, $\mathcal{H}$-matrix is based on the typical kernel functions, singularities only occur at the diagonal and the function is smooth everywhere else[12].

A standard Galerkin discretisation of $A[u](x)$ for a basis $(\varphi_j)_{j\in I}, I = \{1,...,n\}$, $V_n := span\{\varphi_1,...,\varphi_n\}$, yields a matrix A with entries

$$a_{i,j} = \int_{\Gamma}\int_{\Gamma} \varphi_j(x)K(x,y)\psi_i(y)d\Gamma(x)d\Gamma(y) \tag{3}$$

The support of the kernel function $K(x,y)$ is in general not local. So, a dense matrix **A** arises. The computational complexity for computing and the storage requirement for storing a dense matrix are quadratic to the number of the degrees of freedom. Hence, an efficient approximation method has to be developed.

2.2 Low-rank approximation

The rank of a matrix **A** is defined as the number of linearly independent columns or rows of **A**. For a matrix $\mathbf{A} \in C^{m\times n}$, the rank of **A** is bounded by the minimum value of the **m** and **n**. Although the matrices in the BEM usually have full rank, they can often be approximated by matrices having a much lower rank. For example, a matrix $\mathbf{A} \in C^{m\times n}$, the rank of **A** is **k**. A low rank approximation of **A** can be represented as a factorisation of the form $A = UV^T$ with matrices $U \in R^{n\times k}, V \in R^{m\times k}$ (cf. Fig. 1).

The storage requirement in low rank approximation is $k(m+n)$ in contrast to the quadratic cost mn for standard full matrix. Hence, if the k is small and the condition $k(m+n) < mn$ is satisfied, **A** matrix is said to be of low rank. These matrices are favorably stored in the outer-product form.
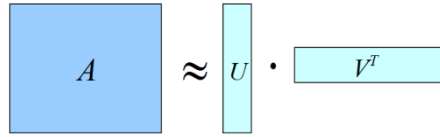
Figure 1:   The schematic diagram of the low rank approximation.

The operational complexity in the matrix-vector multiplication is also efficiently improved. In the standard representation, the number of dominant arithmetic operations is *mn*. In contrast to this, only $2k(m+n)\text{-}k\text{-}n$ additions and multiplications of real numbers are necessary in the outer-product form.

The best possible low rank approximation of the matrix $\mathbf{A} \in C^{m \times n}$ can be obtained by its singular value decomposition (SVD). However, the cost of computing an SVD for a general matrix is expensive. For this reason, some cross approximation algorithms are applied [9–13]. In Section 3, we will construct the low rank approximation of the matrix $\mathbf{A}$ (cf. (3)) by a non-iterative cross approximation algorithm.

3   The geometric cross approximation algorithm

The cross approximation algorithm can be regarded as the procedure of searching the skeleton points from the two admissible interacting groups of nodes. The higher approximation accuracy can be obtained if the columns or rows in the two low rank matrices which are computed by the skeleton points have the larger linear independence. In the BEM, the kernel functions are asymptotically smooth, and are the decay functions of the distance between the source point and the field points. Thus, we try to determine the skeleton points based on the relative geometric relationship between the boundary nodes.
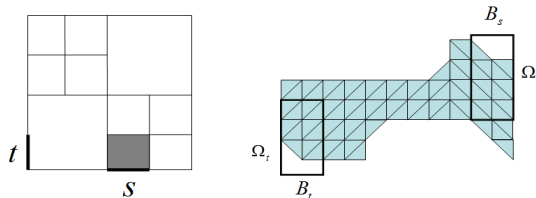

Figure 2:   The admissible block $t \times s$ corresponds to a subset $\Omega_t \times \Omega_s$.

Usually, for the existence of low-range approximations in the case of asymptotically smooth kernels we have to subdivide the coefficient matrix $\mathbf{A}$ into admissible blocks that fulfill the admissibility condition. The algorithm that

constructs a partition of the coefficient matrix can be referred in [9–13]. Here, we just introduce a new cross approximation algorithm to approximate the admissible blocks by low rank matrices.

$$\max\{diam \, B_t, diam \, B_s\} \leq \eta dist(B_t, B_s) \tag{4}$$

This algorithm requires that the strong admissibility condition holds (cf. (4)). In other words, it will be effective only on those parts of the domain that are far from the singularity as well.

3.1 The geometric octree structure for clustering nodes

In this section, we will introduce the geometric octree structure for clustering the boundary nodes corresponding to the admissible block in the boxes B_s and B_t, respectively. Firstly, we fix axially parallel boxes B_s and B_t which bound the relational nodes of the BEM (Figs 2–3) and hold the strong admissibility condition (eqn. (4)).
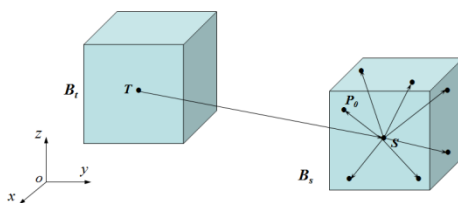

Figure 3:   The boxes B_t and B_s in the three dimensions.

For simplicity, we take the box B_s as example. We construct an octree structure in the box B_s with the depth to be 3. The box of the octree parallel to the coordinate axes x, y and z, is the minimal box bounding all the boundary nodes and centered by the point S. This box is called the cell of level 0. Then, start dividing the parent cell into eight equal child cells of level 1. Continue dividing in this way, that is, take a parent cell of level k and divide it into eight child cells of level k+1. A cell having no child cells is called a leaf. Therefore, we can get sixty-four leaves in this octree structure. In the same way, we also construct the geometric octree structure in the box B_t.

3.2 The right choice of nodes in the leaves

In the preceding section, we construct the geometric octree structures and cluster all the nodes by the leaves in the boxes B_s and B_t. Obviously, there may be more than one node or no in some leaves. But, only one 'best' node will be recorded in a leaf where the extra nodes are existent. In the course of this section, we will introduce the right choice of nodes if there is more than one node in the leaves.

Firstly, we denote the center of the box B_t by T and the center of one of the leaves in the box B_t by O. As can be seen in Fig. 4, we denote the vector drawn from the center T to the center O by the vector A. The vectors can be obtained by drawing from the center O to the nodes in this leaf. For example, a node in this
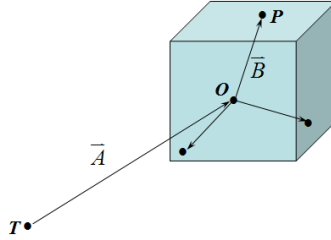
Figure 4:   The right choice of node in the leaf.

leaf is denoted by P. So, we can obtain the vector $\boldsymbol{B}$ drawn from the center O to the node P. we define the quantity λ about the node P as follows

$$\lambda = \bar{A} \cdot \bar{B} \tag{5}$$

The quantity λ is the dot product between the vectors A and B. Obviously, there is the quantity λ about every node in this leaf. Then, the node which has the maximum λ will be recorded as the best choice in this leaf. Taking the same way in all of the leaves, we can obtain the right choice of nodes in all the leaves.

3.3 The priority order of all the leaves for determining the skeleton points

In the preceding sections, we cluster all the nodes in the boxes by the octree structure and obtain the right choice of nodes in the leaves. In the course of this section, we will determine the skeleton points by the priority order of all the leaves shown in Fig. 5.
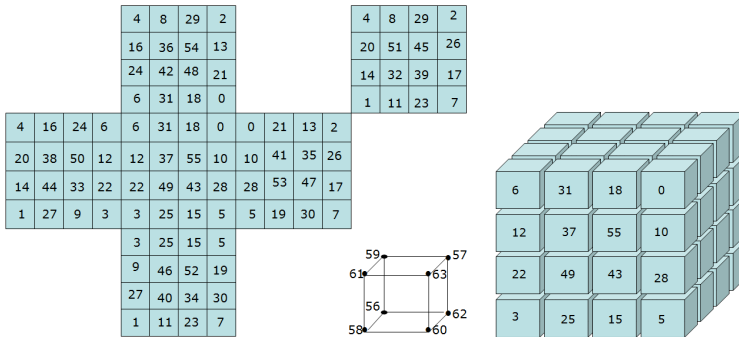

Figure 5:   The priority orders of all the leaves in the octree structure.

   We number the sixty-four leaves in a special way (Fig. 5). In this way, we can ensure that the serial numbers of two leaves are adjacent but the distance between them is almost farthest. We sequentially search the 'best' nodes in the leaves by serial number which denotes the priority order of the leaves. At last,

we can take the 'best' nodes in the box B_t as the skeleton points whose number does not excess a specified maximum number.

The same as the procedure of determining the skeleton points in the box B_t, we can obtain the skeleton points in the box B_s. Take the minimum value of the numbers of skeleton points in boxes B_t and B_s as the rank k of the low rank matrices U and V (cf. Sect. 2.2). We take the same number of skeleton points according to the rank k in the boxes B_t and B_s by their priority orders, respectively.

By the above method, we can obtain a set of skeleton points in boxes B_t and B_s, respectively. The distances between the skeleton points are farthest as possible, so that the columns or rows in low-rank matrices which correspond to the skeleton points can be linearly independent as large as possible.

3.4 Assembling the low rank approximation

In Section 3.3, we have obtained the skeleton points in the two boxes. Then, the low rank approximation represented as the out-product form (cf. Sect. 2.2) can be obtained as follows:

Firstly, the low rank matrix U is assembled as its columns derived from the matrix $\mathbf{A}$ which correspond to the skeleton points in the box B_s. Secondly, we construct the low rank matrix V^T as its rows derived from the matrix $\mathbf{A}$ which correspond to the skeleton points in the box B_t. At last, we obtain the low rank approximation of the matrix $\mathbf{A}$. However, the amount of storage required for the low rank approximation can still be reduced, for the low rank matrices U and V usually have orthonormal columns.

Hence, we assemble the low rank approximation based on the pseudo-skeleton representation as follows:

$$UV^T = A_{1:m,\,j_{1:k}} A_k^{-1} A_{i_{1:k},\,1:n} \tag{6}$$

where $A_k := A_{i_{1:k},\,j_{1:k}}$. The parts $A_{1:m,\,j_{1:k}}$ and $A_{i_{1:k},\,1:n}$ are derived from the original matrix $\mathbf{A}$ the same as the matrices U and V. In this way, we construct the matrix V

$$V^T = A_k^{-1} A_{i_{1:k},\,1:n} \tag{7}$$

where A_k^{-1} assembled by the SVD of the matrix A_k.

4 Numerical examples

The purpose of this section is to validate the new algorithm. In this section, we apply the GCA and the ACA in the examples with the single layer potential for comparison.

Here, we consider the matrix $\mathbf{A}$ with entries

$$A_{ij} := \frac{1}{\left\| x_i - y_j \right\|} \quad i \in m, j \in n$$

where the vertices $x_i \in X$, $y_j \in Y$, are chosen as in the following figures.
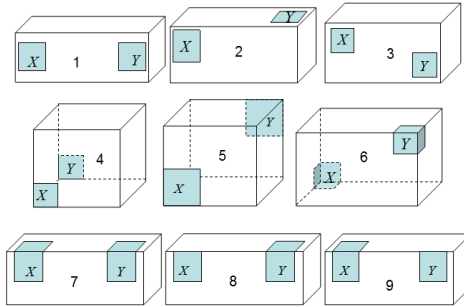
Figure 6:   The geometries for the examples with the single layer potential.

Table 1:   Numerical results for the examples with the GCA and ACA.

Examples	m	n	GCA		ACA	
			k	Rel.Err	k	Rel.Err
1	1600	1600	4	2.5×10^{-5}	4	1.1×10^{-5}
2	3600	3600	5	8.4×10^{-5}	5	1.3×10^{-5}
3	1600	1600	5	6.0×10^{-6}	4	8.8×10^{-5}
4	3600	3600	4	8.4×10^{-6}	4	2.0×10^{-5}
5	3600	3600	3	1.2×10^{-6}	3	3.9×10^{-5}
6	2700	2025	4	5.2×10^{-7}	4	7.3×10^{-5}
7	3200	3200	6	6.7×10^{-5}	6	2.4×10^{-5}
8	3200	1600	4	6.8×10^{-5}	4	1.0×10^{-4}
9	1600	3200	4	3.2×10^{-6}	5	4.3×10^{-5}

The numerical results for the examples shown in Fig. 6 are listed in Table 1. In the Table, m and n denote the numbers of the vertices in the domains X and Y, respectively. The "Rel.Err" giving the relative error $\left\| I - \tilde{A}^{-1}A \right\|_2$ and the corresponding rank k obtained by the GCA and the ACA are listed in columns 4, 5, 6, 7, respectively. It is seen that the new algorithm can compute the low rank efficiently.

5   Conclusions

A new low rank approximation algorithm is proposed in this paper. This algorithm determines the skeleton points based on their topological spatial relations which just needs the spatial geometric information of the nodes. As it has been detailed, this algorithm can determine the skeleton points in the admissible interacting blocks in advance, and the boundary integrals which correspond to the skeleton points in the far-field can be computed together with those for the near-field interaction. Integrals for both near-field and far-field

interactions are computed in a unified framework in the same way as the traditional BEM. It will be convenient and effective to use higher order elements in the simulation. Numerical examples have shown that the new algorithm can compute the low rank approximation efficiently. Concerning its advantages stated above, our method deserves further consideration. The low rank approximation of the kernel functions with the double layer potential is ongoing work.

Acknowledgement

References

[1]   W. Hackbusch, Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. Numer. Math., 1989; 54: 463-491.
[2]   S. Sauter. Variable order panel clustering. Computing., 2000; 64: 223-261.
[3]   V. Rokhlin. Rapid solution of integral equations of classical potential theory. J. Comput. Phys., 1985; 60: 187-207.
[4]   L. Greengard, V. Rokhlin. A new version of the fast multipole method for the Laplace in three dimensions. Acta Numer., 1997; 6: 229-269.
[5]   J.M. Zhang, Masa. Tanaka, M. Endo. The hybrid boundary node method accelerated by fast multipole method for 3-D potential problems. Int. J. Num. Meth. Eng., 2005; 63: 660-680.
[6]   J.M. Zhang, Masa. Tanaka. Adaptive spatial decomposition in fast multipole method. J. Comput. Phys., 2007; 226: 17-28.
[7]   J.M. Zhang, Masa. Tanaka. Systematic study of thermal properties of CNT composites by a fast multipole hybrid boundary node method. Eng. Anal. Bound. Elem., 2007; 31: 388-401.
[8]   J.M. Zhang, C. Zhuang, X.Y. Qin, G.Y. Li, X.M. Sheng. FMM-accelerated hybrid boundary node method for multi-domain problems, Eng. Anal. Bound. Elem., 2010; 34: 433-439.
[9]   S. Börm, L. Grasedyck. Low-rank approximation of integral operators by interpolation. Computing., 2004; 72: 325-332.
[10] M. Bebendorf. Approximation of boundary element matrices. Numer. Math., 2000; 86: 565-589.
[11] M. Bebendorf, S. Rjasanov. Adaptive Low-Rank Approximation of Collocation Matrices. Computing., 2003; 70: 1-24.
[12] E. Tyrtyshnikov. Incomplete cross approximation in the mosaic-skeleton method. Computing., 2000; 64: 367-380.
[13] S. Börm, L. Grasedyck. Hybrid cross approximation of integral operators. Numer. Math., 2005; 101: 221-249.
[14] W. Dahmen, R. Schneider. Wavelets on manifolds I: Construction and domain decomposition. SIAM J. Math. Anal., 1999; 31: 184-230.

[15] M. Bebendorf, R. Kriemann. Fast parallel solution of boundary integral equations and related problems. Comp. Vis. Sci., 2005; 8: 121-135.
[16] M. Bebendorf. Hierarchical LU decomposition-based preconditioners for BEM. Computing., 2005; 74: 225-247.
[17] P.G. Martinsson, V. Rokhlin, M. Tygert. A randomized algorithm for the decomposition of matrices. Appl. Comput. Harmon. Anal., 2011; 30: 47-68.