ADAPTIVE CROSS APPROXIMATION WITH A GEOMETRICAL PIVOT CHOICE: ACA-GP METHOD*

VLADISLAV A. YASTREBOV† AND CAMILLE NOÛS‡

Abstract. The Adaptive Cross Approximation (ACA) method is widely used to approximate admissible blocks of hierarchical matrices, or $\mathscr{H}$-matrices, from discretized operators in the boundary integral method. These matrices are fully populated, making their storage and manipulation resource-intensive. ACA constructs a low-rank approximation by evaluating only a few rows and columns of the original operator, significantly reducing computational costs. A key aspect of ACA's effectiveness is the selection of pivots, which are entries common to the evaluated row and column of the original matrix. This paper proposes combining the classical, purely algebraic ACA method with a geometrical pivot selection based on the central subsets and extreme property subsets. The method is named ACA-GP, GP stands for Geometrical Pivots. The superiority of the ACA-GP compared to the classical ACA is demonstrated using a classical Green operator for two clouds of interacting points.

Key words. Low-rank approximation, Adaptive Cross Approximation, Geometrical Pivots, Boundary Integral Method, Hierarchical matrices.

MSC codes. 65F35, 65F30, 15A03, 65R20, 65Y20, 65F99

Supplementary material: The source code of the ACA and ACA-GP methods is available at github.com/vyastreb/ACA, and the instance used for all studies presented in the paper is available at github.com/vyastreb/ACA/releases/tag/v0.1.0 and archive.softwareheritage.org. Supplementary material with all the data presented in the paper and all additional results is available at zenodo.org/14809517.

1. Introduction.

Low-rank approximation of matrices plays a crucial role in the numerical analysis of discretized differential or integral operators for boundary value problems in mathematical physics and various other domains. In the context of matrices arising from Boundary Integral (BIM) or Boundary Element (BEM) methods, they are often huge and dense, posing significant challenges in terms of storage and computational efficiency. Low-rank approximation techniques enable the reduction of these matrices to more manageable forms without sacrificing accuracy and improving efficiency of operations: various decompositions, matrix-vector product, inversion, etc. This simplification not only enhances computational performance but also makes it feasible to tackle with enhanced accuracy complex problems in scientific computing, engineering, and data science. Beyond mathematical physics, low-rank approximations are invaluable in fields such as machine learning, signal processing, and bioinformatics, where they facilitate efficient data compression, noise reduction, and the extraction of meaningful patterns from large datasets.

The method of Adaptive Cross Approximation (ACA) introduced in [5, 8, 7] has been successfully used to approximate admissible blocks of hierarchical matrices or $\mathscr{H}$-matrices [6] of discretized operators arising from the boundary integral method. Since the matrices associated with such operators are fully populated, their storage and manipulation are memory- and computationally intensive. The ACA permits not only to construct a low-rank approximation of the admissible blocks, but also it allows to control the approximation by evaluation only a few rows and columns of the original operator. This shortcut is especially beneficial when the computation of such entries is computationally expensive. The crucial point in an effective usage of the ACA is the choice of the so-called pivots which correspond

*Submitted to the editors February 6, 2025

†Centre des matériaux, Mines Paris, PSL University, 91003 Evry, France (vladislav.yastrebov@minesparis.psl.eu, www.yastrebov.fr).

‡Cogitamus Laboratory, Paris, France (camille.nous@cogitamus.fr, www.cogitamus.fr).

to the common entry of the evaluated row and the column of the original matrix. In this article we suggest combining a purely algebraic classical ACA method with a geometrical choice of the pivots. This choice is based on the distance between the points in the cloud of points which represent the geometry of the problem. The proposed method is called ACA-GP (GP stands for Geometrical Pivots). We demonstrate the efficiency of the ACA-GP on a classical Green operator for elasticity.

The concept of $\mathscr{H}$-matrices was introduced in [20], which is the most general class of hierarchical matrices with a strong admissibility. In $\mathscr{H}$-matrices, off-diagonal blocks can be low-rank and full-rank sub-matrices, a class of HODLR (hierarchical off-diagonal low-rank) matrices, as the name indicates, have only low-rank matrices at the off-diagonal blocks. So HODLR has a simpler structure. The inverse of a matrix with low-rank blocks can be based on $\mathscr{H}$-matrix algebra [6, 17] or, as it was done in [3], on the Woodbury matrix identity

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1}$$

which in the context of ACA decomposition has $C = I_{k\times k}$ and $C^{-1} = I$ and $UCV = UV^{\top}$. HSS (Hierarchical Semiseparable) matrices [10, 27] present a subclass of HODLR matrices and allow $\mathscr{O}(n)$-complexity solvers, an example can be found in [25]. Another type of matrix structure is a Block Low-Rank (BLR) matrix format, a flat non-hierarchical block matrix structure [2]. We can distinguish between a *weak* and *strong admissibility* of non-overlapping domains as was introduced in [1].

Exist numerous techniques to construct low-rank approximations of admissible blocks $K \in \mathbb{R}^{m\times n}$, $n < m$. They could be subdivided in two main classes: (i) a very general algebraic skeleton-based or pseudo-skeleton approximations [15], where particular rows and columns of the original matrix are selected, (ii) by direct decompositions of the matrix into more adapted for low-rank approximation forms such as SVD, LU and QR, and (iii) kernel approximations, where the underlying operator of two variables is approximated as a product of two separate operators acting on one variable. The first class is the most efficient in terms of computational resources, the second is the most general and accurate, the two can be applied to any matrix, the third class is more specific and is applicable when the matrix is defined by a smooth kernel operator. A survey could be found in [18]. The most popular methods are summarized below:

- **Interpolative Decomposition (ID)** based on matrix skeletonization [11], complexity $\mathscr{O}(mn\log k + k^2 n)$ or as described in [23] as at most $\mathscr{O}(kmn\log(n))$ but typically $\mathscr{O}(kmn)$.
- **Randomized algorithms** [13, 12, 21] with a fast randomized algorithm [23, 26] with the complexity $\mathscr{O}(nmlog(k) + nl^2)$ where l is of order k but $k < l < n$. The fast version is based on the ID and requires a knowledge of the matrix
- **Adaptive Cross Appoximation (ACA)** with full pivoting needs to scan the full matrix and Partial pivoting Adaptive Cross Approximation (ACA) and ACA+, seek for largest entries in a row or column at every approximation step; they work well for rather homogeneous matrix structure, but can fail when especial rows and columns are present in the matrix, complexity $\mathscr{O}(k(m+n))$.
- **Boundary Distance Low-Rank (BDLR)** [3], relies on the underlying sparse matrix graph to choose the desired rows and columns, complexity $\mathscr{O}(nk)$ however it can be used in a special context and requires the original sparse matrix from the finite element solver.
- **CUR** decomposition [24] at the boundary between classical decompositions and pseudo-skeleton approximations, where the selected rows and columns exhivit a "high statistical leverage", but it is mainly used in the context of data compression and is not directly related to matrices obtained from BIM, which is the main focus of this work.

- **Rank revealing** QR algorithm with a pivoted Gram-Schmidt algorithm to obtain r orthogonal basis vectors, used, for example, in [22], complexity $\mathscr{O}(mnk)$
- **Rank revealing** LU decomposition, complexity $\mathscr{O}(mnk)$
- **Singular Value Decomposition** (SVD) [14, e.g.], complexity for a full SVD construction $\mathscr{O}(mn^2 + n^3)$ and for a truncated SVD $\mathscr{O}(kmn)$.
- **Chebyshev low-rank approximation** for smooth kernels is an expansion of the kernel as a series of Chebyshev polynomials, complexity $\mathscr{O}(nk)$.
- **Multipole expansion** is another type of expansion of smooth kernels as series of spherical harmonics, it is used in the Fast Multipole Method [19], complexity $\mathscr{O}(kn)$.

In this short paper we develop a new method for a low-rank approximation of matrices arising from physical interaction between admissible (well separated) domains described by a smooth kernel operator. It is based on the psedo-skeleton approximation and is largely inspired by a purely algebraic partial pivot ACA method, but the choice of pivots is based on the combination of algebraic and geometrical properties of the problem. This paper is organized as follows. In Section 2 we present the ACA method and its proposed ACA-GP modification for point-wise interaction. In Section 3 we present the results of the ACA-GP method applied to a Green operator for a set of separate clouds of points. The performance of the method is compared with the classical ACA and the most computationally extensive but the most accurate Singular Value Decomposition (SVD) ensuring the best low-rank approximation. Finally, in Section 4 we draw some conclusions and propose future research directions.

2. Methods.

2.1. Low rank approximation of admissible blocks. Let us consider admissible blocks of a hierarchical matrix [6] of a discretized operator arising from the Boundary Integral Method (BIM) [9]. By admissible blocks we understand the blocks corresponding to subdomains $X, Y \subset \Omega$ which verify the following geometrical criterion [6]:

$$\min\{\mathrm{diam}(X), \mathrm{diam}(Y)\} \leq \eta \mathrm{dist}(X, Y), \tag{2.1}$$

where by $\mathrm{diam}(\bullet)$ we understand a certain measure of the set extension which can be precomputed with a reasonable (linear) complexity, for example as a the doubled distance from the center to the farthest element. Since the computation of the minimal distance between the sets has a high ($\mathscr{O}(nm)$) complexity, a "relaxed" distance condition shall be used. This relaxed distance will be denoted with a prime $\mathrm{dist}'(X, Y)$, meaning that this is an upper bound of the real Euclidean distance. This approximate distance can be also computed with a reasonable (linear) complexity, for example, as a distance between centers of subdomains $\bar{x}, \bar{y}$ of X and Y minus the estimated half-sum of diameters, i.e.

$$\begin{aligned}\mathrm{dist(X, Y)} \leq \mathrm{dist}'(X, Y) = \|\bar{x} - \bar{y}\| - (\mathrm{diam}(X) - \mathrm{diam}(Y))/2 \leq \\ \leq \|\bar{x} - \bar{y}\| - \min\{\mathrm{diam}(X), \mathrm{diam}(Y)\}.\end{aligned}$$

For a set of points, the center can be computed with a linear complexity as a barycenter:

$$\bar{x} = \frac{1}{|X|} \sum_{i \in X} x_i, \quad \bar{y} = \frac{1}{|Y|} \sum_{j \in Y} y_j, \tag{2.2}$$

where $|X|, |Y|$ denote the number of points in the sets X, Y, respectively. Then the admissibility criterion Equation (2.1) reduces to

$$\min\{\mathrm{diam}(X), \mathrm{diam}(Y)\} \leq \alpha \|\bar{x} - \bar{y}\|, \quad \alpha = \frac{\eta}{1 + \eta}. \tag{2.3}$$

Now let us assume that the components a_{ij} of the admissible block of the discrete BIM operator $A \in \mathbb{R}^{n\times m}$ can be computed as

$$a_{ij} = \int_{X_i}\int_{Y_j} \kappa(x,y)\,dx\,dy, \quad i = 1,\dots,n, \quad j = 1,\dots,m$$

where $\kappa(x,y)$ is the kernel operator. Since the subdomains are separated, there is no singular function evaluation. The objective is to evaluate a low-rank approximation A'_k of the operator A such that the rank k of the approximation ensures a controllable error $\|A - A'_k\|_F$ and that k preferably remains much smaller than the size of the original matrix $k \ll \min\{m,n\}$.

2.2. Adaptive Cross Approximation (ACA). The ACA method [8] is based on the general idea of a pseudo-skeleton representation of the matrix A [15]. The approximate low-rank matrix A'_k of rank k is represented as an outer product of submatrices U and V:

$$A'_k = \sum_{i=1}^{k} u_i v_i^T = UV^T, \qquad U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{m\times k} \tag{2.4}$$

which can be adaptively constructed until a required accuracy is reached:

$$\forall \varepsilon > 0, \exists k \quad \text{s.t.} \quad \|A - A'_k\|_F \le \varepsilon \|A\|_F, \tag{2.5}$$

where ε is the required accuracy ϵ and $\|\bullet\|_F$ is the Frobenius norm. Of course, since the matrix A is never computed fully nor the low-rank matrix A' is constructed (only its decomposed representation), all these norms are replaced by adapted expressions and will be presented in the following section. The data compression ratio is defined as

$$\frac{|U| + |V|}{n \times m} = \frac{k(n+m)}{n\times m} \xrightarrow{m=n} \frac{2k}{n}$$

which clearly demonstrates that this compression makes sense only if $k < \min\{n,m\}/2$.

The classical ACA method is based on the algorithm shown in Algorithm 2.1. The only liberty in the algorithm is the choice of the pivot row i_k (line 3) from which the pivot column j_k is selected such that it has the maximal entry of the residual column (line 10).The choice of the pivots is crucial for the convergence of the ACA method and classically the row indices i_k are selected such that the Vandermonde matrix corresponding to the system in which the approximation error is to be estimated is non-singular, see [6, 4]. In [16] a slightly different strategy is applied and called ACA+: starting with a random initial row and subsequent maximal component choice for the column and the maximal entry of this column for the row. The ACA+ method is claimed to be more robust than the classical ACA but as demonstrated in [6], in some situations it could be suboptimal.

Let us now demonstrate the ACA type matrix approximation in a more explicit form assuming that components a_{ij} are computed as $a_{ij} = 1/|x_i - y_j|$. The first rank $k = 1$ is constructed as

$$A'_{1ij} = \frac{a_{i_1,j}a_{i,j_1}}{a_{i_1,j_1}} = \frac{|x_{i_1} - y_{j_1}|}{|x_i - y_{j_1}|\,|y_j - x_{i_1}|}, \quad i = 1,\dots,n, \quad j = 1,\dots,m$$

where i_1, j_1 is the first pivot. The associated residual between the original matrix and this first rank approximation is given by

$$R^1_{ij} = A_{ij} - A'_{1ij} = \frac{1}{|x_i - y_j|} - \frac{|x_{i_1} - y_{j_1}|}{|x_i - y_{j_1}|\,|y_j - x_{i_1}|}. \tag{2.6}$$

We can represent points in the clouds with respect to the first pivot as $x_i = x_{i_1} + \Delta^1 x_i$ and $y_j = y_{j_1} + \Delta^1 y_j$, where $\Delta^1 x_{i_1} = 0$ and $\Delta^1 y_{j_1} = 0$. If we introduce the following notation for the vector connecting two pivot points $d_{11} = x_{i_1} - y_{j_1}$, then the residual becomes

$$R^1_{ij} = \frac{|d_{11} + \Delta^1 x_i||d_{11} - \Delta^1 y_j| - |d_{11} + \Delta^1 x_i - \Delta^1 y_j||d_{11}|}{|d_{11} + \Delta^1 x_i - \Delta^1 y_j||d_{11} + \Delta^1 x_i||d_{11} - \Delta^1 y_j|} \tag{2.7}$$

It is easy to see that for $i = i_1$ and for $j = j_1$ the numerator is equal to zero, i.e. the residual column and line corresponding to the pivot are zero. The second rank $k = 2$ approximation is constructed as

$$A'_{2ij} = A'_{1ij} + \frac{R^1_{i,j_2} R^1_{i_2,j}}{R^1_{i_2,j_2}}$$

where i_2, j_2 is the second pivot. The residual between the original matrix and this second rank approximation is given by

$$R^2_{ij} = A_{ij} - A'_{2ij} = R^1_{ij} - \frac{R^1_{i,j_2} R^1_{i_2,j}}{R^1_{i_2,j_2}} = \frac{R^1_{i,j} R^1_{i_2,j_2} - R^1_{i,j_2} R^1_{i_2,j}}{R^1_{i_2,j_2}}.$$

With this form it is easy to see that for $i = \{i_1, i_2\}$ and $j = \{j_1, j_2\}$ the residual is zero. Recursively, the $(k+1)$ rank approximation is constructed as

$$R^{k+1}_{ij} = A_{ij} - A'_{k+1\,ij} = R^k_{ij} - \frac{R^k_{i,j_{k+1}} R^k_{i_{k+1},j}}{R^k_{i_{k+1},j_{k+1}}} = \frac{R^k_{i,j} R^k_{i_{k+1},j_{k+1}} - R^k_{i,j_{k+1}} R^k_{i_{k+1},j}}{R^k_{i_{k+1},j_{k+1}}}.$$

Algorithm 2.1 Adaptive Cross Approximation (ACA)

Let $k \leftarrow 1$; $I, J \leftarrow \emptyset$; $\epsilon > 0$
repeat
  Find pivot row i_k by some rule
  Evaluate column $\tilde{v}_k \leftarrow A_{i_k,s}$
  for $l = 1, \ldots, k-1$ **do** ▷ Subtract previous columns
    $\tilde{v}_k \leftarrow \tilde{v}_k - (u_l)_{i_k} v_l$
  end for
  $I \leftarrow I \cup \{i_k\}$
  if $\tilde{v}_k$ does not vanish **then**
    $j_k \leftarrow \arg\max_{j \in S} |\tilde{v}_k|_j$
    $J \leftarrow J \cup \{j_k\}$
    Pivot $p_k \leftarrow \tilde{v_k}_{j_k}$
    Evaluate row $\tilde{u}_k \leftarrow A_{t,j_k}$
    for $l = 1, \ldots, k-1$ **do** ▷ Subtract previous rows
      $\tilde{u}_k \leftarrow \tilde{u}_k - (v_l)_{j_k} u_l$
    end for
    Evaluate residual norm $\|R_k\|_F$ and matrix norm $\|A'_k\|_F$
    Renormalize $u_k = \text{sign}(p_k)\tilde{u}_k/\sqrt{p_k}$, $v_k = \tilde{v}_k/\sqrt{p_k}$
    Update matrices $U \leftarrow [U|u_k]$, $V \leftarrow [V|v_k]$
    $k \leftarrow k + 1$
  end if
until $\|R_k\|_F \le \epsilon \|A'_k\|_F$ or $|I| = n$ or $|J| = m$

2.3. Geometrical insights in optimal pivot choice. The exclusively algebraic search for pivots used in ACA, based on maximizing pivot values, has the merit of being geometry-agnostic and easy to implement. However, it is suboptimal, as can be easily demonstrated through a relatively simple test. Take two random separate clouds X and Y of size $|X| = N$, $|Y| = M$ and compute all possible choices for the first pivot $\{i_1, j_1\}$ such that $\{i_1, j_1\} \in N \times M = \{1, 2, \ldots, n\} \times \{1, 2, \ldots, m\}$ for every pivot we can construct vectors $\tilde{u}_1 = a_{i,j_1}$ and $\tilde{v}_1 = a_{i_1,j}$ and approximate the matrix by

$$A'_1(i_1, j_1) = \tilde{u}_1 \tilde{v}_1^\top / p_{i_1,j_1}, \quad p_{i_1,j_1} = a_{i_1,j_1}.$$

Then we can select the optimal pivot $\{i_1^*, j_1^*\}$ as

$$\{i_1^*, j_1^*\} = \arg \min_{\{i_1,j_1\} \in N \times M} \left(\|A'_1(i_1, j_1) - A\|_F \right),$$

by definition

$$\|A'_1(i_1^*, j_1^*) - A\|_F \leq \|A'_1(i_1, j_1) - A\|_F, \forall i_1 \in N, j_1 \in M.$$

In the ACA i_1 is selected randomly and $j_1 = \arg\max_{j \in M} |A_{i_1,j}|$. We can visualize the error in the approximation for the two clouds for a fixed row or a column, i.e. $\forall i, \{i, j_1\}$ for cloud X and $\forall j, \{i_1, j\}$ for cloud Y. It makes sense to show the error for $i_1 = i_1^*$ and $j_1 = j_1^*$. Now we proceed to the error estimation of the next rank k by assuming that on the previous rank we selected the optimal pivot $\{i_{k-1}^*, j_{k-1}^*\}$. The scaled rows and columns are stored in matrices $U = \{u_1, u_2, \ldots, u_k\}$ and $V = \{v_1, v_2, \ldots, v_k\}$, with

$$u_l = \text{sign}(p(i_l, j_l)) \tilde{u}_l / \sqrt{|p(i_l, j_l)|}, \tag{2.8}$$

$$v_l = \tilde{v}_l / \sqrt{|p(i_l, j_l)|} \tag{2.9}$$

so that $A'_k = UV^\top$, where $\tilde{u}_l, \tilde{v}_l$ are the row and the column vectors of the residual matrix R_{k-1} and $p(i_l, j_l)$ is the pivot value:

$$\tilde{u}_l = R_{i,j_l}^{l-1} = a_{i,j_l} - a_{i,j_{l-1}}'^{l-1}, \tag{2.10}$$

$$\tilde{v}_l = R_{i_l,j}^{l-1} = a_{i_l,j} - a_{i_{l-1},j}'^{l-1}, \tag{2.11}$$

$$p(i_l, j_l) = \tilde{u}_l(i_l) = \tilde{v}_l(j_l). \tag{2.12}$$

The relative error for the pivot choice $\{i_l, j_l\}$ can be computed as

$$E_k(i_k, j_k) = \frac{\|A'_k(i_k, j_k) - A\|_F}{\|A\|_F}.$$

Again, we can find the optimal choice for the pivot

$$\{i_k^*, j_k^*\} = \arg \min_{\{i_k,j_k\} \in \{N \setminus I\} \times \{M \setminus J\}} E_k(i_k, j_k),$$

where $I = \{i_1, i_2, \ldots, i_{k-1}\}$, $J = \{j_1, j_2, \ldots, j_{k-1}\}$ contains rows and columns of previously selected pivots, thus $N \setminus I$ and $M \setminus J$ are sets of the remaining rows and columns. Again we can visualize the error in two clouds for a fixed row or a column, i.e. $\forall i, \{i, j_k^*\}$ for cloud X and $\forall j, \{i_k^*, j\}$ for cloud Y. Let's call this search strategy *genetic* as for every rank we select the best possible approximation, which is of course, unimaginable in practice as this requires construction of $n \times m$ full matrices $n \times m$ times to find the most optimal one, so, the complexity would be $\mathcal{O}(kn^2m^2)$ which is too prohibitive in practice. But here our objective is first to demonstrate the sub-optimality of the classical ACA pivot selection, and second to reveal the geometrical structure of the optimal pivot location. By pivot location we mean

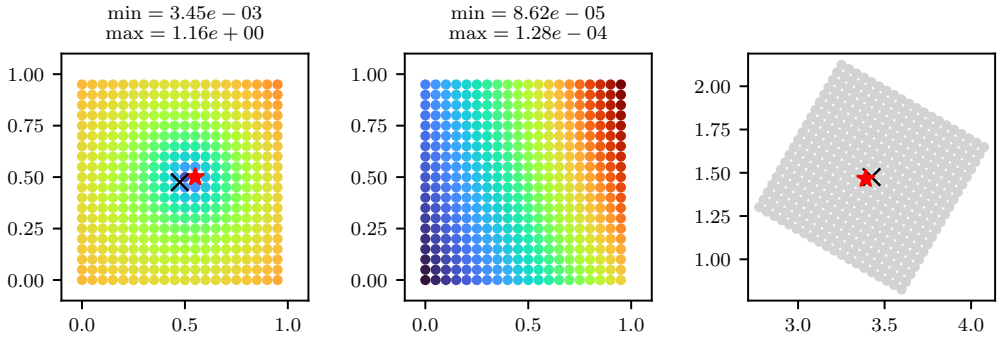

Fig. 1: Illustration of the optimal choice of the first pivot $\{i_1^*, j_1^*\}$ for the rank $k = 1$ for two clouds of points ($x_{i_1^*}, y_{j_1^*}$ are marked with stars). The optimal pivot points are located near the centers of the clouds. Left column: relative error above the SVD approximation: $\tilde{E}_1 = (E_1 - E_1^{\text{SVD}})/E_1^{\text{SVD}}$. Central column: interaction matrix's column $a_{i_1,j}$ colored according to their value. Right column: the location of the optimal column i_1 in the cloud X. Black crosses correspond to the barycenters of the clouds.

the geometrical coordinates of the optimal row i_k (point x_{i_k}) and column j_k (point y_{j_k}) in the clouds X and Y, respectively.

To analyse the geometrical signature of the optimal pivot choice we analyse two clouds of points X and Y located at some distance and oriented in a random fashion. Random and structural point distributions within the clouds are studied based on the full genetic data generated. To make the error estimation more meaningful, we subtract the relative error of the SVD approximation for the corresponding rank k and normalize by the relative SVD error:

$$\tilde{E}_k = \frac{E_k - E_k^{\text{SVD}}}{E_k^{\text{SVD}}} = \frac{\|A'_k - A\|_F}{\|A'^{\text{SVD}}_k - A\|_F} - 1 \geq 0, \tag{2.13}$$

since, by definition the SVD is the optimal low-rank approximation, we can be sure that the error remains positive.

Based on the data analysis of the genetic ACA, the *first observation* we make is that the optimal choice of the first pivot is located near centers of the clouds X and Y, which is quite obvious. It can be readily seen in Figure 1 that the points near the center possess the minimal error.

The *second observation* is that starting from rank $k = 2$, minimal absolute values of the residual matrix's column corresponds to the maximal approximation errors. By residual matrix, as previously, we understand $R_{k-1} = A - A'_{k-1} = a_{ij} - \sum_{i=1}^{k-1} u_i v_i^\top$. So for every i_k^* we can evaluate all values $R_{k-1}(i_k^*, j)$; in the classical ACA we select the maximal value $j_k^* = \arg_j \max |R_{k-1}(i_k^*, j)|$, but as can be seen for $k = 1$ in Figure 1 (for which $R_0 = A$), and for higher ranks as illustrated in Figure 2, this choice is not optimal. Nevertheless, what is clear is that the minimal absolute values of the residual column correspond to the maximal errors of the approximation $\tilde{E}_k$ and thus the ACA by choosing the higher values of the residual safely avoids too bad approximation.

The *third observation* is that the optimal points concentrate in the center of the cloud. It could be readily seen in Figure 2: black circles correspond to the optimal pivots selected using genetic ACA. It contrasts with the classical ACA choice in which the maximal values of the residual correspond to points on the boundary of the cloud as can be easily seen in the central column of Figure 2.

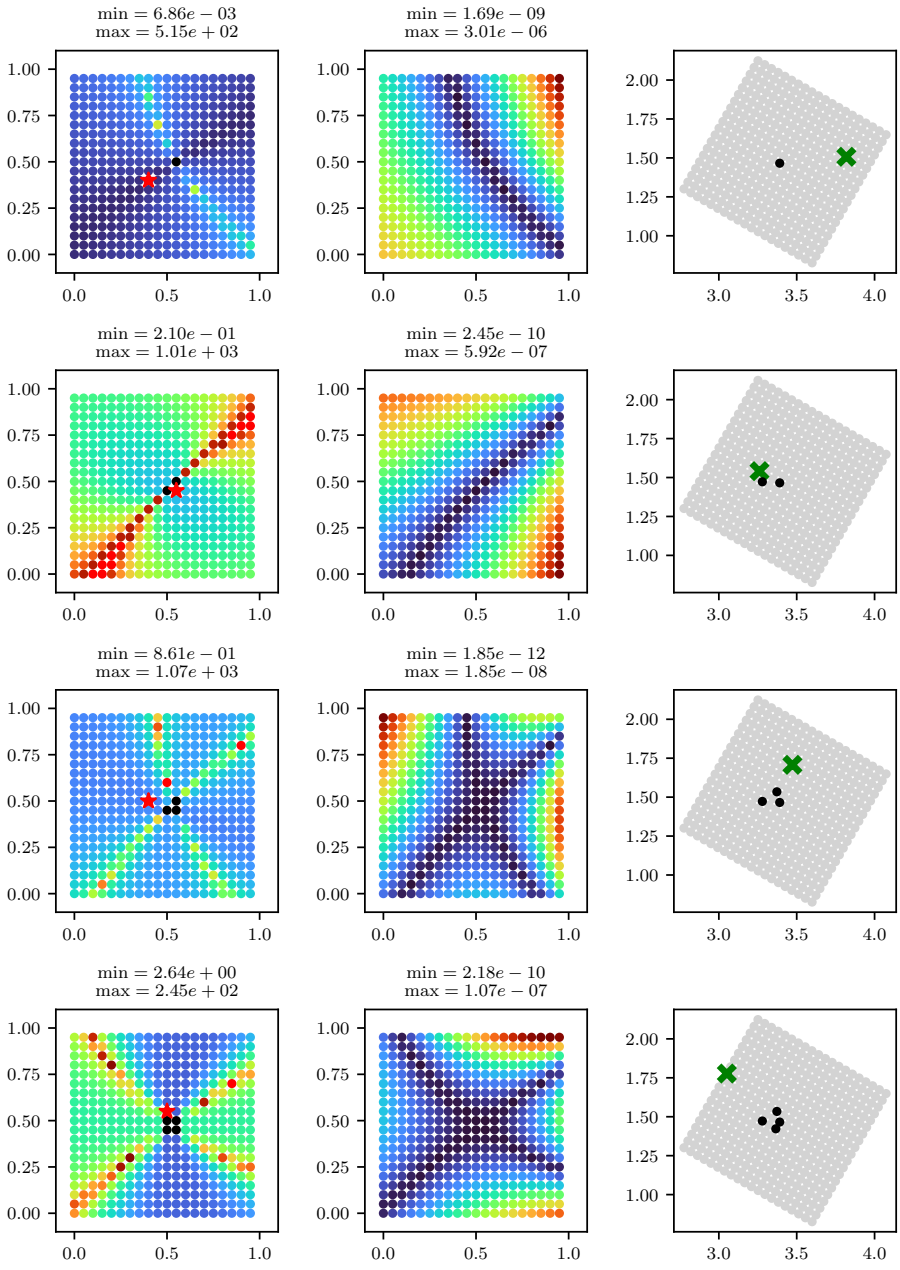

Fig. 2: Illustration of the error structure for ranks $k = \{2, 3, 4, 5\}$ starting from the top and a randomly selected $\{i_2, i_3, i_4, i_5\}$ (note that all pivots i_l^*, j_l^* with $l < k$ were assumed to be selected in an optimal way and are shown with black circles) in cloud X shown with a cross on the right column; the left column represents the relative difference in genetic ACA's error (relative error of the Frobenius norm of the matrix approximation) with respect to SVD $\tilde{E}_k = (E_g - E_{\text{SVD}})/E_{\text{SVD}}$; the central column shows the absolute value of the residual column corresponding to selected i_k; the clouds have $n = m = 400$ points.

The *fourth observation* is that the zones of extreme errors (minimal or maximal) lie on some curved lines which seem to pass through optimal pivots of the previous ranks. These curves, let's call them *extreme curves*, could be approximated by some function $\mathscr{E}_k : f(x) = 0$ such that error minimizing and maximizing pivots lie on these extreme curves: $f(x_i) \approx 0 \approx f(y_j)$. The sign $\approx$ reflects the discrete nature of points and the continuous nature of the curves. We can distinguish extreme curves of maximal $\mathscr{E}_k^{\max}$ and minimal $\mathscr{E}_k^{\min}$ errors. This observation is valid for $k > 1$. Regardless a seeming simplicity of these extreme curves and relatively simple origin algebraic equation for their construction, they cannot be easily determined for arbitrary domains. Therefore, the construction of the algorithm will combine such curves for ranks $k = 2$ and $k = 3$ with the third observation about the optimal points concentration in the center of the cloud. Nevertheless, we will make an attempt to reveal the geometrical structure of these extreme curves for simple domains and formalize the algorithm for the optimal pivot selection in the following sections.

2.4. Geometrical structure of the error distribution. To take profit of the observation that the optimal points concentrate in the center of the cloud, we introduce two central subsets I^c, J^c of the clouds such that

$$I^c = \left\{ i \in \{1, \dots, n\} \;\middle|\; |x_i - x_{i_1}| \leq \varepsilon_r \mathrm{diam}(X) \right\} \tag{2.14}$$

$$J^c = \left\{ j \in \{1, \dots, m\} \;\middle|\; |y_j - y_{j_1}| \leq \varepsilon_r \mathrm{diam}(Y) \right\}, \tag{2.15}$$

which contain indices of points close to the points forming the first pivot x_{i_1}, y_{j_1}. The parameter ε_r is a relative distance to the diameter of the cloud. The only condition on the selection of the central subsets is that they must contain sufficient number of points for the relevant range of ranks, for example, if the maximal rank is bounded by max(rank) then the number of points in the central subset must contain at least this number with a relatively small offset of the order of $\Delta \approx 5 - 10$: $|I_c| \geq \max(\text{rank}) + \Delta$. The algorithm for the central subset selection is presented in Algorithm 2.2. If the required number of points is not verified, the central ratio fraction is increased, for example, by 10 %.

The sensitivity of the approximation to this central fraction – the only parameter of the algorithm – will be evaluated in Section 3. The central subsets I^c, J^c will be used to construct *ad hoc* rules for the selection of optimal points. As will be shown, a selection algorithm based on extreme curves can be constructed for more or less square domains for the ranks $k = 2$ and $k = 3$, for which the geometrical structure of error is rather trivial. For higher ranks $k \geq 4$ or arbitrary domains a simpler procedure combining central subsets and algebraic considerations is adopted, which will be presented in Subsection 2.4.3.

2.4.1. Rank $k = 2$. Let us start the study from the second low-rank approximation $k = 2$, we assume that we selected the optimal pivot $\{i_1^*, j_1^*\}$ for the first rank. The objective is to select such point $x_{i_2^*}$ in cloud X and such point $y_{j_2^*}$ in cloud Y so that the newly approximated matrix $A_2' = (u_1, u_2)(v_1, v_2)^\top$ minimizes the Frobenius norm $\{i_2^*, j_2^*\} = \arg\min_{i,j} \|A - A_2'\|_F$. If a point in cloud X is selected in random fashion i_2, then there exist such j_2' which minimizes the aforementioned norm for given i_2. At the same time, for every choice of i_2, one can visualize the error $\tilde{E}_2(i_2, j)$, see Equation (2.13), for all points of the cloud Y as it was done before.

On the top of the clouds colored according to relative error $\tilde{E}_k$, we make a simple geometrical construction: a circle $\mathscr{C}_2(i_2)$ passing through points $x_{i_1^*}$, $y_{j_1^*}$ and the new (tentative) point x_{i_2}. According to our experiments, for any choice of i_2, the optimal $y(j_2')$ lies on the same circle with an eventual spatial shift related to the spatial distribution of points. Therefore, the extreme curve for i_2: $\mathscr{E}_2^{\min}(i_2)$ can be approximated by $\mathscr{C}_2(i_2)$. Let us construct

Algorithm 2.2 Central subset selection

Let first pivot $\{i_1\}$ be selected
Let $I^c \leftarrow \emptyset$; $\varepsilon_r > 0$
while $|I^c| < \max(\text{rank}) + \Delta$ **do**
  for $i = 1, \dots, n$ **do**
    if $|x_i - x_{i_1}| \leq \varepsilon_r \text{diam}(X)$ **then**
      $I^c \leftarrow I^c \cup \{i\}$
    end if
  end for
  if $|I^c| < \max(\text{rank}) + \Delta$ **then**
    $\varepsilon_r \leftarrow 1.1\varepsilon_r$
  end if
end while

a *conjugate circle* passing through the point $y_{j_1^*}$: $\mathscr{C}_2^{\perp}(y_{j_1^*})$ (or for shorter notation $\mathscr{C}_2^{\perp}(j_1^*)$) in such a way that it has the same radius as $\mathscr{C}_2$, it intersects the point $y_{j_1^*}$ orthogonally to the circle $\mathscr{C}_2$, and the center $c_2^{\perp}$ verifies $(c_2^{\perp} - y_{j_1^*}) \cdot (x_{i_1^*} - y_{j_1^*}) \geq 0$. Constructed in such a way, this conjugate circle represents the extreme curve of maximal errors $\mathscr{E}_2^{\max}(i_2) = \mathscr{C}_2^{\perp}$. Interestingly, regularly structured clouds of points possess the same geometrical structure of the error as randomly distributed clouds with a relatively high and uniform density. In Figures 3 and 4 we show the error distribution for randomly selected i_2 along with the introduced geometrical constructions: the circle $\mathscr{C}_2(i_2)$ and the conjugate circle $\mathscr{C}_2^{\perp}(y_{j_1^*})$. Only two configurations are shown, more examples are provided in Supplementary material [29].

Now, we need to develop a set of *ad hoc* rules which would help us to select more or less optimal points. In order to keep these rules simple to implement, we propose an algorithm shown in Algorithm 2.3. It aims to iteratively select an optimal pair of points from two clouds to keep small the approximation error for a rank-2 matrix. The process starts by randomly selecting a point from one of the clouds $x_{i_2} \in X$ within the central subset $i_2 \in I^c \setminus \{i_1\}$, after which a corresponding point in the central subset $J_c \setminus \{j_1\}$ of the cloud Y is chosen such that it lies close to the extreme circle and maximizes the residual component $|R^1_{j,i_2}|$. The algorithm refines the selection by iteratively searching for the best candidate, balancing proximity and error minimization, until the optimal point is found. As can be seen, this algorithm combines geometric and algebraic considerations and has a linear complexity. Moreover, contrary to the classical ACA it requires evaluation of only a small fraction of R^1_{ij} components and thus could be even more advantageous than the classical ACA, especially for situations where the evaluation of the kernel is expensive. The complexity of the algorithm remains $\mathscr{O}(|Y|)$, but the number of operations is of the order $\beta|J^c| \approx \beta\lfloor\varepsilon_r|Y| - 1\rfloor$ with the factor $\beta \sim 2$–4.

2.4.2. Rank $k = 3$. Third rank approximation is more trivial than the second one because the underlying geometry of errors remains stable with respect to the choice of the x_{i_3} point under the condition that the latter is not located too close to $\mathscr{C}_2(i_2^*)$ which becomes now the extreme curve of maximal errors $\mathscr{E}_3^{\max}$. On the other hand, the conjugate extreme surface $\mathscr{C}_2^{\perp}(j_1^*)$ becomes the minimal error curve $\mathscr{E}_3^{\min}$, i.e. the extreme curves exchange their roles:

$$\mathscr{E}_3^{\max} = \mathscr{E}_2^{\min} = \mathscr{C}_2(i_2^*), \quad \mathscr{E}_3^{\min} = \mathscr{E}_2^{\max} = \mathscr{C}_2^{\perp}(j_1^*).$$

The error structure shown in Figure 5 remains practically independent of the choice of i_3 if it is selected far from the $\mathscr{E}_3^{\max}$ extreme curve (see supplementary material for more examples, [29]). On the contrary, the selection of x_{i_3} close to $\mathscr{C}_2(i_2^*)$ as shown in Figure 6 leads to a significant increase in the error (see the minimal value in the figure) and can

Algorithm 2.3 Selection of optimal points for rank $k = 2$

Require: Central subsets I^c, J^c (excluding the first pivot)
1: Select a point x_{i_2} randomly such that $i_2 \in I^c$.
2: Initialize $l \leftarrow 1$.
3: Copy the central subset $J^c_l \leftarrow J^c$.
4: **repeat**
5: Select a point $y_{j^l_2}$ such that $j^l_2 \in J^c_l$ and minimizes the distance to the circle $\mathscr{C}_2(x_{i_2})$:

$$j^l_2 \leftarrow \arg\min_{j_2 \in J^c_l} \operatorname{dist}(y_{j_2}, \mathscr{C}_2(x_{i_2}))$$

6: Evaluate $|R^1_{i_2,j^l_2}| = |a_{i_2,j^l_2} - a'_{1(i_2,j^l_2)}|$.
7: **if** $l > 1$ **and** $|R^1_{i_2,j^l_2}| \le |R^1_{i_2,j^{l-1}_2}|$ **then**
8: Stop iterations and select $j_2 = j^{l-1}_2$.
9: **else**
10: $l \leftarrow l + 1$
11: Update the copy of the central subset $J^c_l \leftarrow J^c_l \setminus \{j^l_2\}$
12: **end if**
13: **until** convergence

alter the geometry of the error distribution. Therefore, we assume that if one selects x_{i_3} far from $\mathscr{C}_2(i^*_2)$, i.e. close to the circle passing through $x_{i^*_1}$ and orthogonal to the circle $\mathscr{C}_2(i^*_2)$: $\mathscr{C}^\perp_2(i^*_1)$ (our assumption), and then one selects y_{j_3} close to $\mathscr{C}^\perp_2(j^*_1)$, then the error will be minimal.

The Algorithm 2.4 for selecting the points i_3 and j_3 is rather similar to the one used for i_2, j_2. The main difference is that the i_3 point is selected in the central subset close to the $\mathscr{C}^\perp_2(x_{i^*_1})$ and j_3 is selected in an iterative way close to $\mathscr{C}^\perp_2(y_{j^*_1})$. The algorithm is as follows:

- Select $i_3 \in I^c \setminus \{i_1, i_2\}$ such that

$$i_3 = \arg\min_{i \in I^c \setminus \{i_1,i_2\}} \operatorname{dist}(x_i, \mathscr{C}^\perp_2(x_{i^*_1})).$$

- Select trial $j^1_3 \in J^c \setminus \{j_1, j_2\}$ such that

$$j^1_3 = \arg\min_{j \in J^c \setminus \{j_1,j_2\}} \operatorname{dist}(y_j, \mathscr{C}^\perp_2(y_{j^*_1})).$$

- Evaluate $|R^2_{i_3,j^1_3}| = |a_{i_3,j^1_3} - a'^2_{i_3,j^1_3}|$.
- Iterate to find j_3 such that

$$j^{l+1}_3 = \arg\min_{j \in J^c \setminus \{j_1,j_2,j^1_3,\ldots,j^l_3\}} \operatorname{dist}(y_j, \mathscr{C}^\perp_2(j^*_1))$$

before $|R^2_{i_3,j^{l+1}_3}| \le |R^2_{i_3,j^l_3}|$ and select $j_3 = j^l_3$ as the optimal point.

The complexity of the algorithm is $\mathscr{O}(|X| + |Y|)$, but the number of operations is of the order $|I^c| + \beta|J^c| \approx \lfloor \varepsilon_r |X| - 2 \rfloor + \beta \lfloor \varepsilon_r |Y| - 2 \rfloor$ with the factor $\beta \sim 2$–4. Note that, contrary to the second rank selection, this algorithm for the selection of the 3rd rank pivot is not restricted to square-shaped domains and, in principle, can be applied to arbitrary domains.

2.4.3. Higher ranks $k \ge 4$. For higher ranks establishing a simple deterministic procedure relying on extreme curves seems too tedious: trivial curves do not ensure extreme values of error even though visually it can be noticed that they also form curves with more

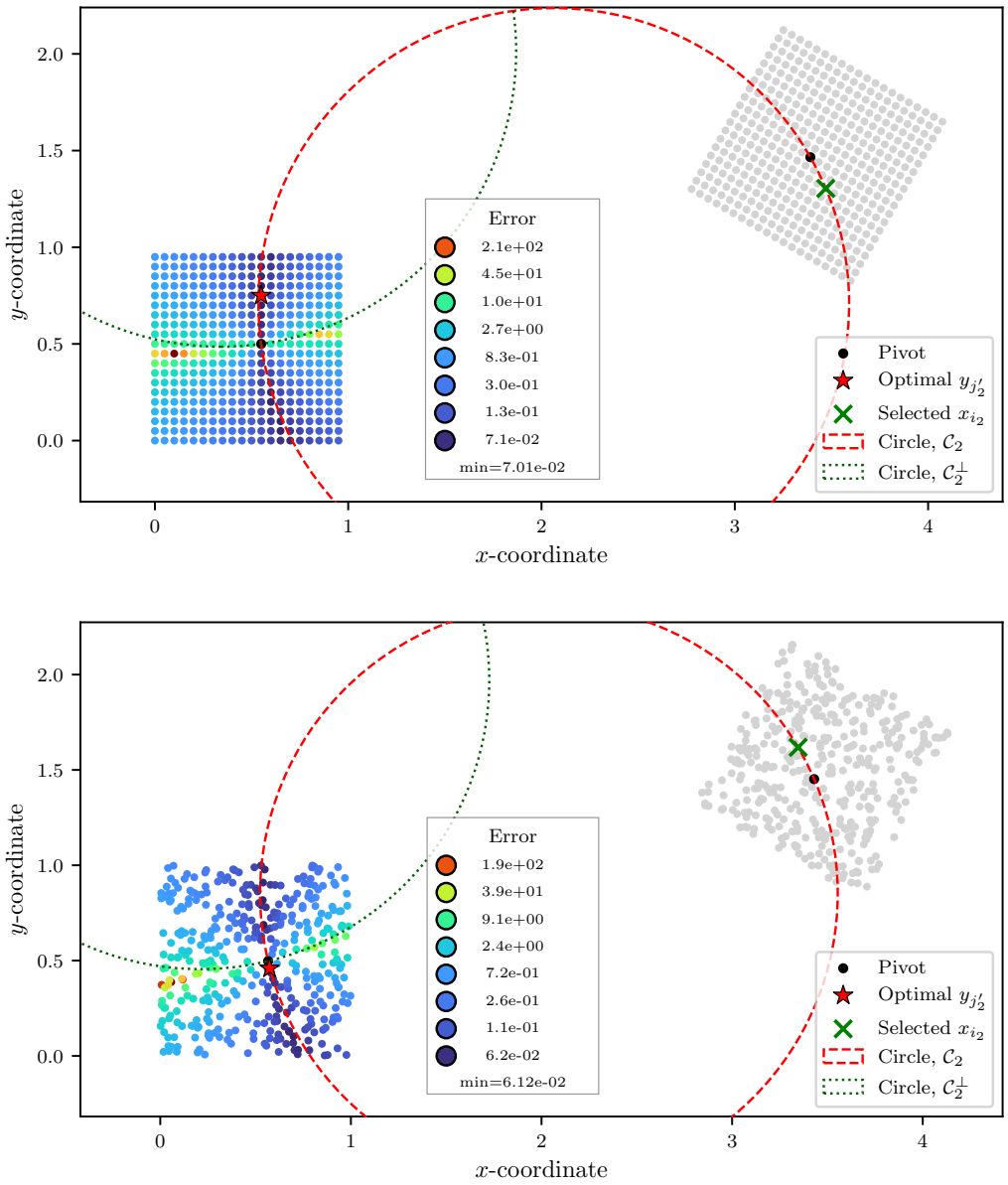

Fig. 3: Illustration of the error structure for the rank $k = 2$ for two clouds of structured (upper panel) and random (lower panel) points within square boxes. A randomly selected point x_{i_2} in X cloud is marked with a green cross and the corresponding optimal $y_{j'_2}$ is shown with a star, the corresponding circle $\mathscr{C}_2(i_2)$ is shown with a red dashed line, the conjugate circle $\mathscr{C}_2^\perp$ is shown with a green dotted line; black circles highlight the first optimal pivot $x(i_1^*)$ and $y(j_1^*)$.

or less constant curvature. Therefore, an even simpler approach combining geometrical consideration and algebraic analysis could be used. The algorithm is presented in Algorithm 2.5 and consists in a two step evaluation of the maximal residual component over the central

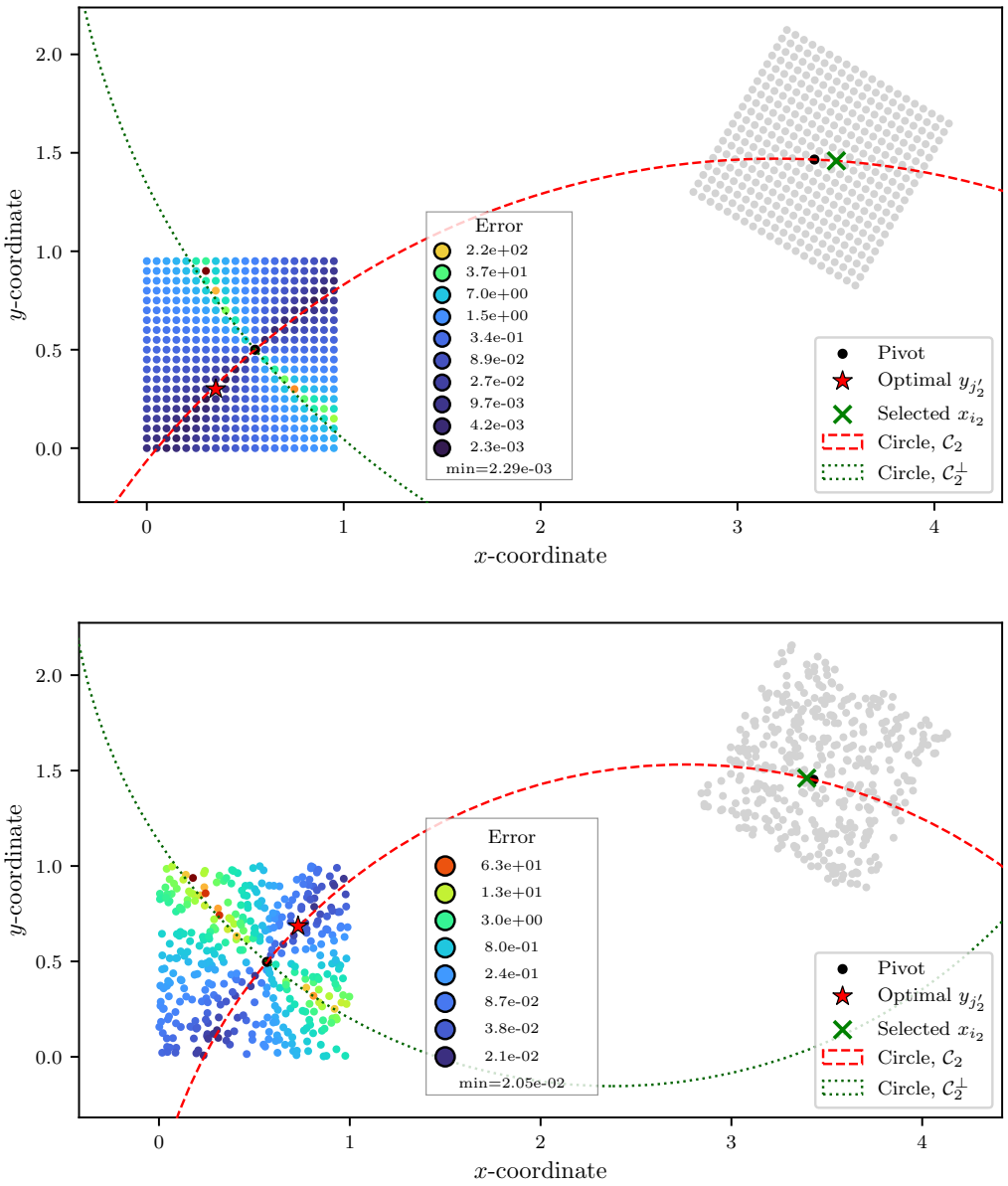

Fig. 4: Illustration of the error structure for the rank $k = 2$ for two clouds of structured (upper panel) and random (lower panel) points within square boxes. A randomly selected point x_{i_2} in X cloud is marked with a green cross and the corresponding optimal $y_{j'_2}$ is shown with a star, the corresponding circle $\mathscr{C}_2(i_2)$ is shown with a red dashed line, the conjugate circle $\mathscr{C}_2^{\perp}$ is shown with a green dotted line; black circles highlight the first optimal pivot $x(i_1^*)$ and $y(j_1^*)$.

subsets. As previously, we introduce adjusted central subsets by excluding previously selected points:

$$I_k^c = I^c \setminus \{i_1, \ldots, i_{k-1}\}, \quad J_k^c = J^c \setminus \{j_1, \ldots, j_{k-1}\}.$$

Algorithm 2.4 Rank $k = 3$ Selection Algorithm

Require: Central subsets I^c, J^c (excluding previously selected points)

Select a point i_3 such that

$$i_3 \leftarrow \arg\min_{i \in I^c} \operatorname{dist}(x_i, \mathscr{C}_2^{\perp}(x_{i_1^*}))$$

Initialize $l \leftarrow 1$

Copy the central subset $J_l^c \leftarrow J^c$

repeat

  Select a point j_3^l such that

$$j_3^l \leftarrow \arg\min_{j \in J_l^c} \operatorname{dist}(y_j, \mathscr{C}_2^{\perp}(y_{j_1^*}))$$

  Evaluate $|R^2_{i_3,j_3^l}| = |a_{i_3,j_3^l} - a'_{2(i_3,j_3^l)}|$.

  if $l > 1$ **and** $|R^2_{i_3,j_3^l}| \le |R^2_{i_3,j_3^{l-1}}|$ **then**

    Stop iterations and select $j_3 = j_3^{l-1}$

  else

    $l \leftarrow l + 1$

    Update the copy of the central subset $J_l^c \leftarrow J_l^c \setminus \{j_3^l\}$

  end if

until convergence

Then for a randomly selected trial $i^t \in I_k^c$ we evaluate the components of the residual matrix corresponding to the columns $j \in J_k^c$:

$$R^{k-1}_{i^t j} = a_{i^t j} - \sum_{l=1}^{k-1} u_l(i^t) v_l(j).$$

Among all evaluated components we select the column j_k with the maximal absolute value of the residual component:

$$j_k = \arg\max_{j \in J_k^c} |R^{k-1}_{i^t j}|.$$

Then we evaluate the components of the central rows $i \in I_k^c$:

$$R^{k-1}_{i j_k} = a_{i j_k} - \sum_{l=1}^{k-1} u_l(i) v_l(j_k).$$

and select the row i_k with the maximal absolute value of the residual component:

$$i_k = \arg\max_{i \in I_k^c} |R^{k-1}_{i j_k}|.$$

In such a way, the new pivot i_k, j_k is selected. This algorithm will be used for all ranks $k \ge 4$.

2.5. A New Method: ACA-GP. The previous sections suggest a methodology for an informed choice of pivots which aim at minimizing the approximation errors. In this chapter the full algorithm for a new adaptive low-rank approximation is presented and discussed. In essence, we suggest combining the purely algebraic ACA method with geometrical considerations for the choice of the pivots. This choice is based on combined knowledge of

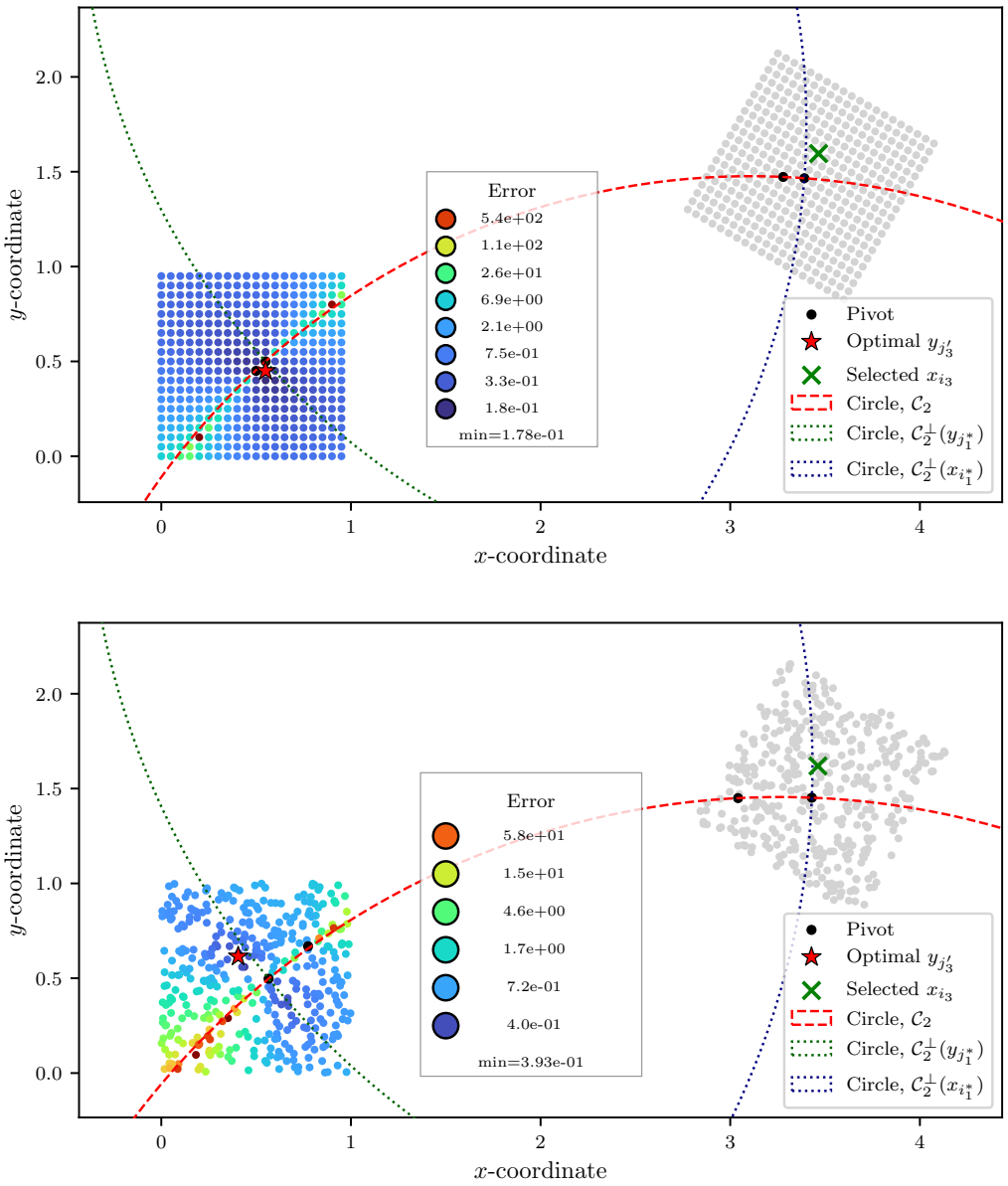

Fig. 5: Error structure for a random choice of i_3 for a structured (upper panel) and random (lower panel) clouds.

spatial distribution of points within the cloud and on the algebraic properties of the residual vectors. The proposed method is named ACA-GP (GP for Geometrical Pivots).

A detailed algorithm is presented in Algorithm 2.6 with associated complexity evaluations for every step. We assumed that $m \leq n$. The following notations were used: two sets of integers $M = \{1, 2, \ldots, m\}$, $N = \{1, 2, \ldots, n\}$ indicate IDs of points (elements) in two separate clouds, whereas sets $I_k = \{i_1, i_2, \ldots, i_k\}$, $J_k = \{j_1, j_2, \ldots, j_k\}$ correspond to pivots (rows and columns) used for skeleton construction of a rank-k approximation, so $N \setminus I_k$,

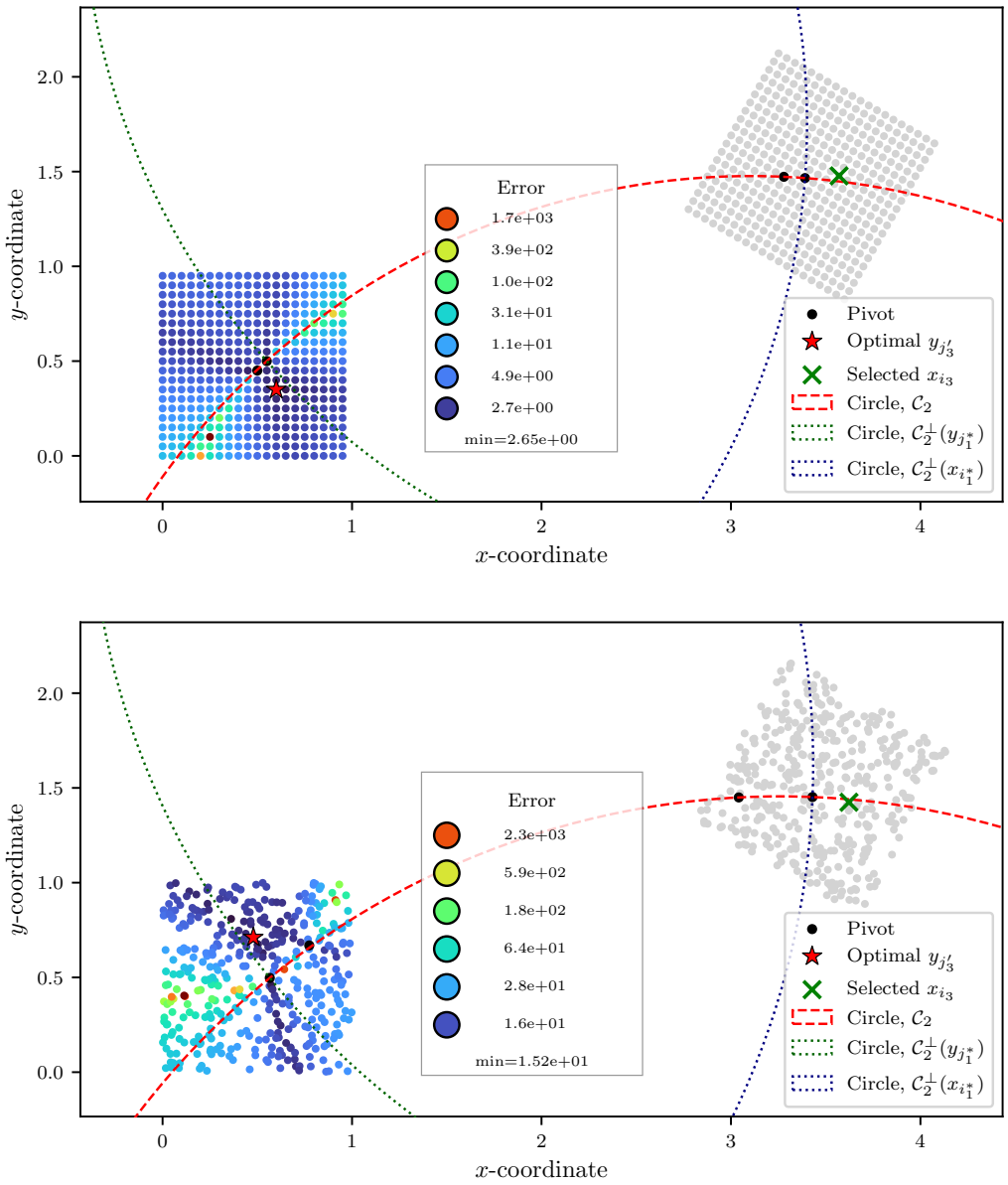

Fig. 6: An example, of the error structure for a bad choice of x_{i_3}, i.e. near the extreme circle $\mathscr{C}_2(i_2^*)$ resulting in a significant increase in the error: upper panel for structured and lower panel for random clouds.

$M \setminus J_k$ correspond to the sets of the remaining columns and rows, respectively. The sets I^c, J^c are the central subsets which are defined by Equation (2.15). The core of the method remains unchanged with respect to the ACA methodology, it is briefly described below to help the interpretation of the algorithm. The matrix A'_k mentioned in Algorithm 2.6 corresponds to the matrix approximation at the k-th step/rank and is defined as $A'_k = U_k V_k^T$ with $U_k = \{u^1, u^2, \ldots, u^k\}$ and $V_k = \{v^1, v^2, \ldots, v^k\}$, but of course A'_k is never evaluated,

Algorithm 2.5 Pivot Choice for Higher Ranks $k \geq 4$

Require: Central subsets I^c, J^c (excluding previously selected points)

1: Select a trial point i^t randomly such that $i^t \in I_k^c$

2: Compute $R_{i^t j}^{k-1}$ for all $j \in J^c$:

$$R_{i^t j}^{k-1} = a_{i^t j} - \sum_{l=1}^{k-1} u_l(i^t) v_l(j)$$

3: Select the column j_k such that

$$j_k \leftarrow \arg\max_{j \in J^c} |R_{i^t j}^{k-1}|$$

4: Compute $R_{i j_k}^{k-1}$ for all $i \in I^c$:

$$R_{i j_k}^{k-1} = a_{i j_k} - \sum_{l=1}^{k-1} u_l(i) v_l(j_k)$$

5: Select the row i_k with the maximal absolute value:

$$i_k \leftarrow \arg\max_{i \in I_k^c} |R_{i j_k}^{k-1}|$$

only matrices U_k, V_k are fully computed and only the necessary operations are performed to construct the next approximation A'_{k+1}. The rows and columns u^i, v^i for $i = 1, \ldots, k$ of the matrices U_k, V_k are constructed from scaled rows and columns of the residual matrices R^{i-1} with $R^0 = A$ in the following way. For the first rank $k = 1$ the rows and columns are constructed as

$$u_i^1 = \operatorname{sign}(p_1) A_{i,j_1} / \sqrt{p_1}, \quad v_j^1 = A_{i_1,j} / \sqrt{p_1},$$

where $p_1 = A_{i_1,j_1}$ is the first pivot (see Algorithm 2.6). The division by pivot is split into two divisions for u and v vectors to avoid division by a too small number. For higher ranks $k > 1$, the rows and columns are constructed as

$$u_i^k = \operatorname{sign}(p_k) \tilde{u}_i^k / \sqrt{p_k}, \quad v_j^k = \tilde{v}_j^k / \sqrt{p_k},$$

where

$$\tilde{u}_i^k = R_{i j_k}^{k-1} = a_{i,j_k} - \sum_{l=1}^{k-1} v_{j_k}^l u_i^l, \quad \tilde{v}_j^k = R_{i_k,j}^{k-1} = a_{i_k,j} - \sum_{l=1}^{k-1} u_{i_k}^l v_j^l,$$

where $p^k = \tilde{u}_{i_k}^k = \tilde{v}_{j_k}^k$ is the k-th pivot.

To evaluate the Frobenius norm of the approximate matrix at the k-th step/rank (lines 35), we use the recursive formula from [8]:

$$\|A'_k\|_F = \sqrt{\|A'_{k-1}\|_F^2 + 2 \sum_{j=1}^{k-1} u_k^\top u_j v_j^\top v_k + R_k^2}, \quad \text{where} \quad R_k^2 = \|u_k\|_F^2 \|v_k\|_F^2 \tag{2.16}$$

The key steps of the entire procedure is the selection of to-be-evaluated rows and columns of the original matrix A, which is done according to algorithms presented in the previous section, namely Algorithms 2.3 to 2.5. The selection of the first pivot is made such that the

Algorithm 2.6 Adaptive Cross Approximation with Geometrical Pivots (ACA-GP) for $m \leq n$

Require: Two clouds $X = \{x_i\}_{i=1}^n$, $Y = \{y_j\}_{j=1}^m$, Kernel $\kappa(x, y)$, global tolerance $\epsilon > 0$ **OR** maximal rank $k_{\max}$
pivot tolerance $\epsilon_p > 0$, central fraction $\varepsilon_r > 0$
// First pivot
Find the center of X: $\bar{x} \leftarrow 1/n \sum_{i \in N} x_i$ ▷ $\mathscr{O}(n)$
Find the center of Y: $\bar{y} \leftarrow 1/m \sum_{j \in M} y_j$ ▷ $\mathscr{O}(m)$
Find the first pivot's row: $i_1 \leftarrow \arg \min_{\substack{i \in N \\ (x_i - \bar{x}) \cdot (\bar{y} - \bar{x}) > 0}} \|x_i - \bar{x}\|$ ▷ $\mathscr{O}(n)$
Find the first pivot's column: $j_1 \leftarrow \arg \min_{\substack{j \in M \\ (y_j - \bar{y}) \cdot (\bar{x} - \bar{y}) > 0}} \|y_j - \bar{y}\|$ ▷ $\mathscr{O}(m)$
$I \leftarrow \{i_1\}$, $J \leftarrow \{j_1\}$.
Pivot $p_1 = \kappa(x_{i_1}, y_{j_1})$
$u^1 \leftarrow \{\mathrm{sign}(p_1)\kappa(x_i, y_{j_1})/\sqrt{p_1}\}, i \in N$, ▷ $\mathscr{O}(m)$
$v^1 \leftarrow \{\kappa(x_{i_1}, y_j)/\sqrt{p_1}\}, i \in M$, ▷ $\mathscr{O}(n)$
$U \leftarrow u^1$, $V \leftarrow v^1$
Compute residuals: $\|R\|_F = \|U\|_F \|V\|_F$, $\|A'\|_F = \|R\|_F$ ▷ $\mathscr{O}(n+m)$
Construct central subsets I_c, J_c according to Algorithm 2.2.
// Main loop
$r \leftarrow 2$
repeat
  if $r = 2$ **then**
    Use Algorithm 2.3 **or** Algorithm 2.5 to select i_2, j_2 ▷ $\mathscr{O}(m)$ or $\mathscr{O}(n+m)$
  else if $r = 3$ **then**
    Use Algorithm 2.4 to select i_3, j_3 ▷ $\mathscr{O}(m)$
  else
    Use Algorithm 2.5 to select i_r, j_r ▷ $\mathscr{O}(n+m)$
  end if
  Update central subsets $I_c \leftarrow I_c \setminus \{i_r\}$, $J_c \leftarrow J_c \setminus \{j_r\}$
  Update pivot sets $I \leftarrow I \cup \{i_r\}$, $J \leftarrow J \cup \{j_r\}$
  Pivot $p_r = \kappa(x_{i_r}, y_{j_r}) - \sum_{l=1}^{r-1} u_l(i_r) v_l(j_r)$
  if $|p_r| < \epsilon_p$ **then return** $U, V, k \leftarrow r - 1$
  end if
  Evaluate column $\tilde{u}^r := \kappa(x_i, y_{j_r}), i \in N$ ▷ $\mathscr{O}(n)$
  Evaluate row $\tilde{v}^r := \kappa(x_{i_r}, y_j), j \in M$ ▷ $\mathscr{O}(m)$
  for $l = 1, \ldots, r-1$ **do** ▷ Substract A'_{r-1} columns and rows, $\mathscr{O}(rn)$
    $\tilde{u}^r := \tilde{u}^r - (v_l)_{j_r} u_l$
    $\tilde{v}^r := \tilde{v}^r - (u_l)_{i_r} v_l$
  end for
  Compute skeleton row and column: $u^r \leftarrow \mathrm{sign}(p_r)\tilde{u}^r/\sqrt{p_r}$, $v^r \leftarrow \tilde{v}^r/\sqrt{p_r}$
  Update outer-product matrices: $U \leftarrow [U|u^r]$, $V \leftarrow [V|v^r]$
  Residual norm $\|R_r\|_F = \|u^r\|_F \|v^r\|_F$ ▷ $\mathscr{O}(n+m)$
  Compute the norm of the approx. matrix $\|A'_r\|_F$ Equation (2.16) ▷ $\mathscr{O}(r(n+m))$
  $r \leftarrow r + 1$
until $\|R_r\|_F \leq \epsilon \|A'_r\|_F$ or $J \equiv M$ or $r = k_{\max}$

pivot is the closest point to the barycenter of the cloud and lies in the half-plane passing

through the barycenter on the side containing the homologue cloud, see Algorithm 2.6:

$$i_1 = \arg \min_{i \in N, (x_i - \bar{x}) \cdot (\bar{y} - \bar{x}) \geq 0} \|x_i - \bar{x}\| \tag{2.17}$$

$$j_1 = \arg \min_{j \in M, (y_j - \bar{y}) \cdot (\bar{x} - \bar{y}) \geq 0} \|y_j - \bar{y}\|, \tag{2.18}$$

where $\bar{x}, \bar{y}$ are the barycenters of clouds X, Y, respectively.

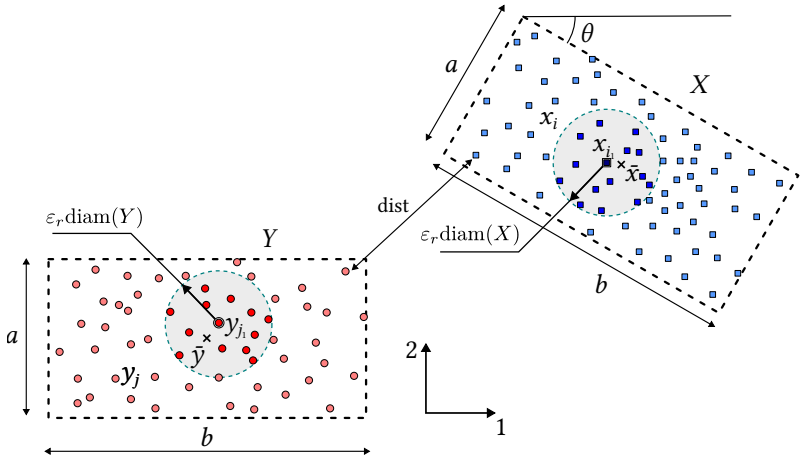

Fig. 7: Set-up of the experiment: two clouds of points X and Y are generated, each in a rectangle $a \times b$ ($\xi = a/b$).

3. Results: ACA-GP for Two Interacting Clouds. We demonstrate the performance of the algorithm on the problem of two interacting separate clouds of points $X = \{x_i\} \subset \mathbb{R}^{n \times d}$, $Y = \{y_j\} \subset \mathbb{R}^{m \times d}$ where $d = 2$ is the space dimension. For the sake of simplicity, points in every cloud X, Y are generated within a rectangle $a \times b$ such that $a \leq b$, and $\xi = a/b$ is the aspect ratio of the rectangle, $\xi \in (0, 1]$ (see Fig. 7). Therefore, the size of every cloud can be approximated by $\mathrm{diam}(Y) \approx \sqrt{a^2 + b^2} = b\sqrt{1 + \xi^2}$. For simplicity, we set $b = 1$, so the distances are measured in units of b. We consider a Poisson point process (uniform distribution) for both clouds X, Y to generate the points. The cloud Y is centred at zero and the rectangle is aligned with the axes, whereas the cloud X is displaced and rotated by a random angle θ (uniform distribution in $[-\pi, \pi]$) in such a way that the criterion of admissibility Equation (2.1) is satisfied. The approximate distance between the clouds is estimated as

$$\mathrm{dist}'(X, Y) = \|\bar{x} - \bar{y}\| - \min\{\mathrm{diam}(X), \mathrm{diam}(Y)\} = \|\bar{x} - \bar{y}\| - b\sqrt{1 + \xi^2}, \tag{3.1}$$

whereas the true distance can be computed as

$$\mathrm{dist}(X, Y) = \min_{i \in N, j \in M} \|x_i - y_j\|. \tag{3.2}$$

The kernel is given by a smooth kernel $\kappa(x, y) = 1/\|x - y\|$ and we assume that the matrix entries are simply

$$a_{ij} = \kappa(x_i, y_j)$$

as it was considered in [16].

The tests are carried out for two aspect ratio $\xi = \{0.5, 1\}$ and for true distances dist $= \{1.5, 2.5, 5.0\}$, the number of points in the clouds is $n, m = 400$. We used $N_s = 1000$ realization of randomly oriented clouds with randomly distributed points. The first objective is to study the effect of the central subset fraction ε_r introduced in Subsection 2.4. The rule of thumb for its selection can be easily constructed. If we suppose that the domain is a circle of diameter $\mathrm{diam}(Y)$, so to have the number of points in the central subset of radius $\varepsilon_r \mathrm{diam}(Y)$ equal to max(rank), we need to verify the condition:

$$\varepsilon_r \gtrsim \sqrt{\frac{\max(\mathrm{rank})}{|Y|}},$$

which is a double of the non-conservative estimation for the needed value. The double is used to account for arbitrary shape of the domain and the fact that the number of points in the central subset should be bigger than the maximal rank of the approximation to ensure optimal performance. So, one has to carefully choose the maximal allowed rank in the algorithm to avoid a too large central subset fraction and excessive computational cost. In the current case for $\max(\mathrm{rank}) = 10$, we get $\varepsilon_r \gtrsim 0.15$.

The effect of the central subset fraction ε_r is not trivial and affects differently different ranks of the approximation. The relative size of the central subset is varied in the interval $\varepsilon_r \in [0.1, 0.5]$. The lower value was chosen such to ensure that a sufficient number of points is available in each central subset even though in Poisson's process it is always possible to have zero points within a central subset. In Figure 8 we show for $\xi = 1$ and dist $= 1.5$ how the accuracy of the approximation varies with the central subset size for the first 10 ranks. The very first rank is, of course, independent on the central ratio, the second rank is also neutral to the variation of the parameter ε_r. Starting from rank $k = 3$ and up to $k = 7$, the increase of ε_r decreases the gain factor, but for ranks $k = 8, 9$ the gain factor is higher for larger ε_r. However, a simple strategy consisting in increasing the central subset individually for those ranks does not work because the overall approximation depends on the whole history of pivot selections. Therefore, a selected central subset fraction should be kept for all ranks to ensure a controllable effect. The value of $\varepsilon_r = 0.25$ presents a reasonable tradeoff between the gain factor for all considered ranks. However, if one is interested in the lower ranks only (up to $k = 6$), one can choose the smallest $\varepsilon_r = 0.1$ to ensure the best accuracy and less computational cost.

The main results are presented in Figures 9 and 10 for different aspect ratios ξ, true distances dist and central subset sizes ε_r. We plot the relative Frobenius norm of the residual matrix $\|A - A'\|_F / \|A\|_F$ for the three methods: ACA, ACA-GP and SVD; the log-mean values and log-standard deviations are identified as:

$$E_{\log} = \log_{10} \frac{\|A - A'_k\|_F}{\|A\|_F}, \quad \bar{E}_{\log} = \frac{1}{N_s} \sum_{s=1}^{N_s} E_{\log}^{(s)} \tag{3.3}$$

$$\sigma_{\log} = \sqrt{\frac{1}{N_s} \sum_{s=1}^{N_s} (E_{\log}^{(s)} - \bar{E}_{\log})^2}, \quad E = 10^{\bar{E}_{\log} \pm \sigma_{\log}}. \tag{3.4}$$

The accuracy gain factor of the ACA-GP to the ACA compared to the SVD approximation is also plotted:

$$\mathrm{gain} = \frac{\|A - A'_{\text{ACA}}\|_F - \|A - A_{\text{SVD}}\|_F}{\|A - A'_{\text{ACA-GP}}\|_F - \|A - A_{\text{SVD}}\|_F}. \tag{3.5}$$

For aspect ratio $\xi = 1$, the results are computed over 1000 realizations, for $\xi = 0.5$ the results are computed over 500 realizations. Many more results can be found in the supplementary material [29]. The key observations are the following:

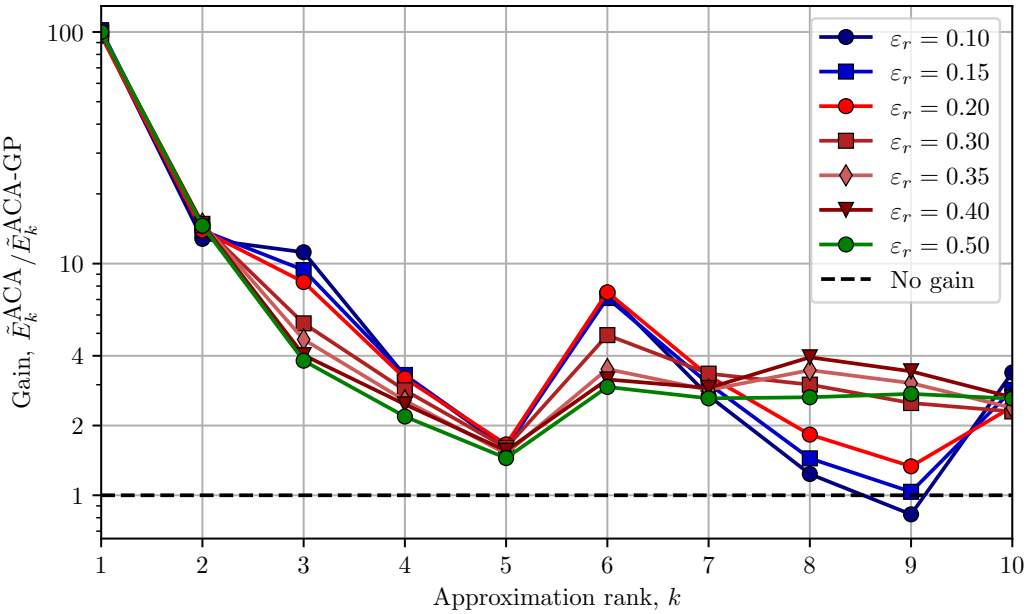

Fig. 8: The effect of the relative central subset fraction ε_r on the accuracy of the low-rank approximation: the gain factor $\tilde{E}_k^{\text{ACA-GP}}/\tilde{E}_k^{\text{ACA}}$ for all ranks are plotted.

- The ACA-GP method strongly outperforms the classical ACA method for most of the considered ranks.
- The average error of the ACA-GP method and its standard deviation are significantly lower than those of the ACA method.
- For square-shaped domains and small ε_r, the first 3 ranks, the ACA-GP method matches very closely in its performance with the SVD approximation if the extreme curves are used through Algorithms 2.3 and 2.4.
- For higher ranks $k \geq 4$ and arbitrary domains (here, rectangular) and properly selected ε_r, the ACA-GP's error represents the geometrical mean of the ACA and SVD errors.
- The ACA-GP is the most efficient up to rank $k = 3$ and for rank $k = 6$, for intermediate ranks $k = \{4, 5\}$, the error decay rate remains small.
- The domain aspect ratio ξ does not affect the performance of the ACA-GP method if the central subset-based selection is used Algorithm 2.5, eventually in combination with Algorithm 2.4.

4. Conclusions. In the context of matrix construction for Boundary Integral or similar methods, the low-rank approximation of the interaction matrix is a key ingredient for the effective use of hierarchical matrices. Within this approach, the interaction between well separated or admissible domains can be efficiently approximated by low-rank matrices. In this work, we proposed a new adaptive low-rank approximation method, called ACA-GP, which combines the algebraic ACA method with a geometry-aided choice of pivots. This new method, similarly to the ACA method, does not require the full original matrix to construct its low-rank approximation and relies on the algorithms with much lower complexity than the SVD. The new method demonstrates a much higher accuracy and has a smaller error dispersion than the classical ACA method.

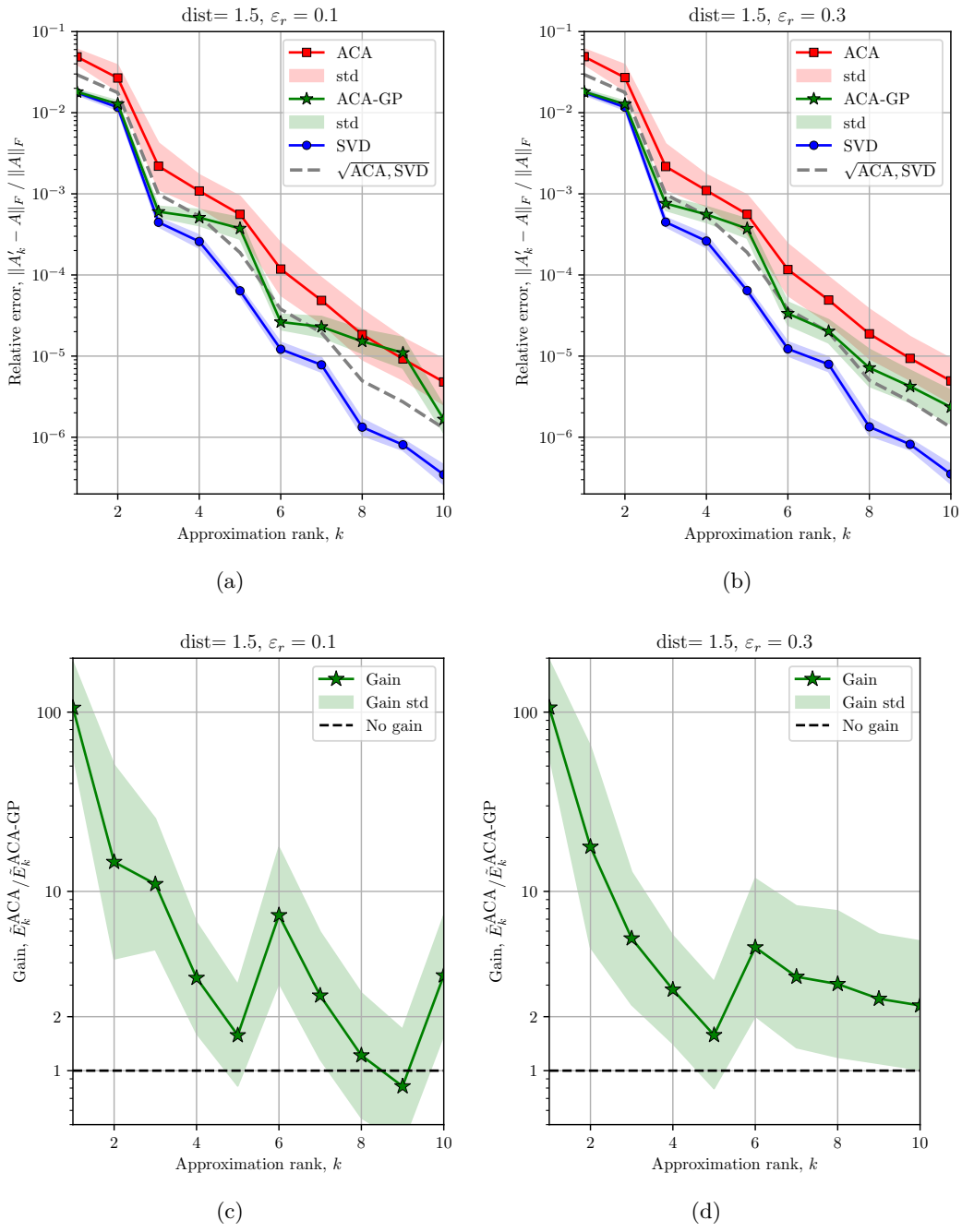

Fig. 9: (a-b) relative error in matrix approximation by ACA, ACA-GP and SVD methods for square domains ($\xi = 1$): log-mean values and log-standard deviations are computed over 1000 realizations; (c-d) corresponding log-mean gain and log-standard deviation between the ACA-GP compared to the ACA method with the offset of SVD accuracy (Equation (3.5)) is also shown; (a,c) corresponds to $\varepsilon_r = 0.1$ and (b,d) to $\varepsilon_r = 0.3$. The true distance is $\text{dist}(X, Y) = 1.5$.

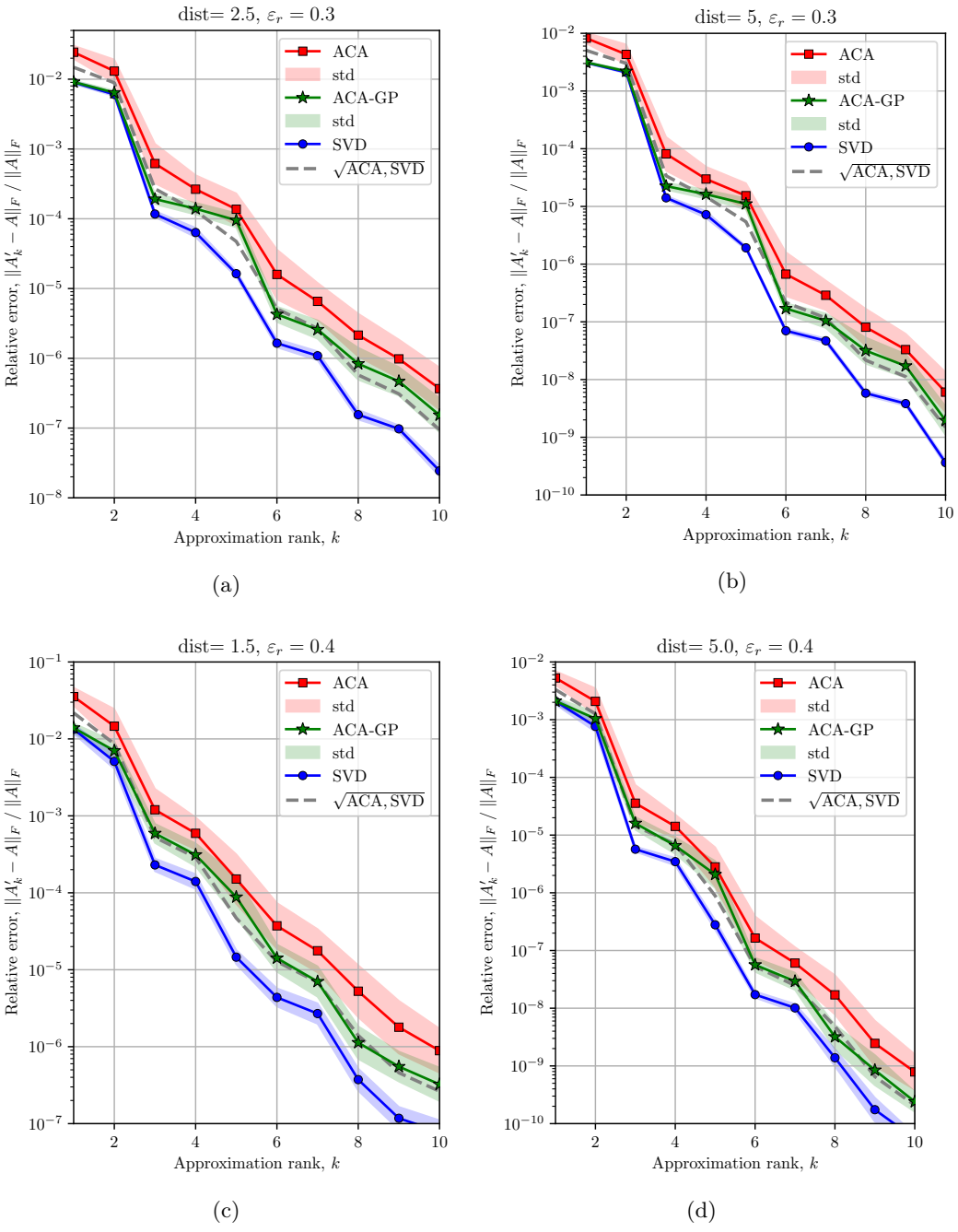

Fig. 10: Relative error in matrix approximation by ACA, ACA-GP and SVD methods for square domains $\xi = 1$ (computed over 1000 realizations) for dist $= 2.5$ in (a) and dist $= 5.0$ in (b) for $\varepsilon = 0.3$; relative errors for rectangular domains $\xi = 0.5$ (computed over 500 realizations) are shown in (c-d) for dist $= 1.5$ and dist $= 5.0$ with $\varepsilon = 0.4$.

The pivot selection procedure is the following. For the first pivot, the central points/elements from the interacting domains are selected. The choice of the pivots for ranks 2 and 3 is based on specific geometrical error structure revealed here, however, it can be well constructed for square-shaped domains only. Nevertheless, this structure is rather independent on the distribution of the points within the interacting domains.For higher ranks, adopting a simple geometric procedure, which constraints the choice of pivots to the central part of the domain, allows to improve the accuracy of the approximation and even reduce the computational cost compared to the classical ACA method. The method was tested on two randomly oriented interacting clouds of points of different aspect ratios. The performance of the ACA-GP method matches closely the SVD performance for the first three ranks (especially for square-shaped domains) and ensures approximation error equal in average to the geometric mean of the ACA and SVD for higher ranks.

In terms of the implementation, the ACA-GP method is slightly more complex and contains one user-defined parameter: the central subset fraction ε_r, which controls the size of the subdomain from which geometrical points corresponding to rows and columns should be selected. The optimal value of this parameter depends on the required rank of the approximation, which in its turn cannot be known in advance because of the key idea of the ACA – the adaptivity – the approximate matrix construction is stopped when the approximate error reaches a certain threshold. Nevertheless, keeping the central fraction as small as possible is beneficial for its performance for lower ranks, which should be often enough in practice. In the worst case scenario, the central subset can be increased when it runs out of available points. Another possible drawback of the focus on the central subset is that in real life, the domains can have a ring-like structure, which therefore does not contain the central subset and could be tricky to handle. In such situation, it would be relevant to switch to the classical ACA method.

In perspective, we plan to adapt the ACA-GP for the collocation BIM method, where the area associated with each domain should be taken into account. We also plan to study in detail the geometrical structure of extreme curves and make an attempt to suggest a simple algorithm for its identification for arbitrary domains.

Our Python implementation of the ACA and ACA-GP is shared on GitHub [28] as well as the data and the scripts used for the experiments [29].

Acknowledgement. This work was partially presented at the World Congress of Computational Mechanics held in Vancouver in the summer of 2024, with fees partially covered by the Comité National Français de Mécanique (CNFM), which is gratefully acknowledged.

REFERENCES

[1] S. Ambikasaran, *Fast algorithms for dense numerical linear algebra and applications*, PhD thesis, Stanford University, 2013.
[2] P. Amestoy, C. Ashcraft, O. Boiteau, A. Buttari, J.-Y. l'Excellent, and C. Weisbecker, *Improving multifrontal methods by means of block low-rank representations*, Journal on Scientific Computing, 37 (2015), pp. A1451–A1474.
[3] A. Aminfar, S. Ambikasaran, and E. Darve, *A fast block low-rank dense solver with applications to finite-element matrices*, Journal of Computational Physics, 304 (2016), pp. 170–188.
[4] M. Bauer and M. Bebendorf, *Block-adaptive cross approximation of discrete integral operators*, Computational Methods in Applied Mathematics, 21 (2021), pp. 13–29.
[5] M. Bebendorf, *Approximation of boundary element matrices*, Numerische Mathematik, 86 (2000), pp. 565–589.
[6] M. Bebendorf, *Hierarchical matrices*, Springer, 2008.
[7] M. Bebendorf and R. Grzhibovskis, *Accelerating galerkin BEM for linear elasticity using adaptive cross approximation*, Mathematical Methods in the Applied Sciences, 29 (2006), pp. 1721–1747.
[8] M. Bebendorf and S. Rjasanow, *Adaptive low-rank approximation of collocation matrices*, Computing, 70 (2003), pp. 1–24.
[9] M. Bonnet, *Boundary Integral Equation Methods for Solids and Fluids*, Wiley, 1999.

[10] S. Chandrasekaran, M. Gu, X. S. Li, and J. Xia, *Some fast algorithms for hierarchically semiseparable matrices*, Tech. Report LBNL-62897, Lawrence Berkeley Nat. Lab., 2008.
[11] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin, *On the compression of low rank matrices*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1389–1404.
[12] A. Deshpande and S. Vempala, *Adaptive sampling and fast low-rank matrix approximation*, in International Workshop on Approximation Algorithms for Combinatorial Optimization, Springer, 2006, pp. 292–303.
[13] A. Frieze, R. Kannan, and S. Vempala, *Fast Monte-Carlo algorithms for finding low-rank approximations*, Journal of the ACM (JACM), 51 (2004), pp. 1025–1041.
[14] G. H. Golub and C. F. Van Loan, *Matrix computations*, JHU press, 2013.
[15] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear algebra and its applications, 261 (1997), pp. 1–21.
[16] L. Grasedyck, *Adaptive recompression of $\mathscr{H}$-matrices for BEM*, Computing, 74 (2005), pp. 205–223.
[17] L. Grasedyck and W. Hackbusch, *Construction and arithmetics of H-matrices*, Computing, 70 (2003), pp. 295–334.
[18] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen, 36 (2013), pp. 53–78.
[19] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, Journal of Computational Physics, 73 (1987), pp. 325–348.
[20] W. Hackbusch, *Sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices*, Computing, 62 (1999), pp. 89–108.
[21] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM review, 53 (2011), pp. 217–288.
[22] W. Y. Kong, J. Bremer, and V. Rokhlin, *An adaptive fast direct solver for boundary integral equations in two dimensions*, Applied and Computational Harmonic Analysis, 31 (2011), pp. 346–369.
[23] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, *Randomized algorithms for the low-rank approximation of matrices*, Proceedings of the National Academy of Sciences, 104 (2007), pp. 20167–20172.
[24] M. W. Mahoney and P. Drineas, *Cur matrix decompositions for improved data analysis*, Proceedings of the National Academy of Sciences, 106 (2009), pp. 697–702.
[25] P. G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, Journal of Computational Physics, 205 (2005), pp. 1–23.
[26] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert, *A fast randomized algorithm for the approximation of matrices*, Applied and Computational Harmonic Analysis, 25 (2008), pp. 335–366.
[27] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li, *Superfast multifrontal method for large structured linear systems of equations*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1382–1411.
[28] V. A. Yastrebov, *Python implementation of the Adaptive Cross Approximation (ACA) and the Adaptive Cross Approximation with Geometrical Pivot selection (ACA-GP) algorithms for an adaptive low-rank approximation of BIM matrices*, 2025. License: BSD-3-Clause, version 0.1.0, GitHub:github.com/vyastreb/ACA, SWHID: archive.softwareheritage.org/swhid.
[29] V. A. Yastrebov, *Supplementary material to ACA-GP paper*. Zenodo: 10.5281/zenodo.14809517, 2025.