Project 2 on *Randomized Nyström*
HPC for numerical methods and data analysis

1 Some important information

- The grade of the project (report plus oral exam) has weight 0.6 for the final grade of the course, that is:
  - Weight for the grade - report: 0.3
  - Weight for the grade - oral exam: 0.3
- Oral exam (individual, 5 mins with 3 slides to present the project + 5 mins Q&A): during the January exam session.
- Students work in groups of two on the project and submit a common report. A student can do the project alone only if justified and after approval by the professor. In this case, you can consider only one sketching operator and one of the two data sets.
- Inform the professor by email about the composition of your group by December 3, 2024.
- Deadline to submit the report (one per group) + slides (individual) on moodle: January 6, 2025, 11:59PM CEST
- The Python and MPI code used for the implementation of the algorithms should be submitted as well as a .zip or a .tar archive.
- All questions should be addressed to Prof. Laura Grigori.

2 Randomized Nyström low rank approximation

The goal of this project is to study the randomized Nyström algorithm for computing a rank-k approximation of a matrix $A \in \mathbb{R}^{n \times n}$ that is symmetric positive semidefinite. Given a sketching matrix $\Omega \in \mathbb{R}^{n \times l}$, where l is the sketch dimension, $l > k$, the randomized Nyström approximation relies on the following formula:

$$A_{Nyst} = (A\Omega)(\Omega^T A\Omega)^+(\Omega^T A), \tag{1}$$

where $(\Omega^T A\Omega)^+$ denotes the pseudoinverse of $\Omega^T A\Omega$. This formula provides an approximation of A of rank at most l. Several solutions are possible for computing a rank-k approximation from A_{Nyst}, that we denote as $[\![A_{Nyst}]\!]_k$. One consists of computing a rank-k decomposition of the matrix $B = \Omega^T A\Omega$, while another one consists of computing a rank-k truncation of the Nyström approximation A_{Nyst}. The second approach is considered for this project. Different solutions are proposed in the literature, see one solution in the slides from the lecture on *Randomized algorithms for low rank matrix approximation*, where the algorithm relies on the Cholesky decomposition or the eigenvalue decomposition of B. Other reference is [4]. Two different data sets should be used to validate your considered randomized Nyström low rank approximation, that are described later on. Hence this project should allow you to identify a randomized algorithm that is numerically stable for the considered data sets and scales reasonably well in parallel.

2.1 Data sets

Two different data sets should be used in this project. The first data set is synthetic and is described in section 5 of [4]. Consider the polynomial decay matrix and the exponential decay matrix with the parameters provided in [4].

The second data set is described for example in [1] (section 4). It uses MNIST or YearPredictionMSD datasets [3, 2], that can be downlowded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. The radial basis function $e^{-\|x_i - x_j\|^2/c^2}$ is used to build a dense matrix A of size $n \times n$ from n rows of the input data. The parameter c should be varied and can be chosen as 100 for the MNIST dataset and 10^4 as well as 10^5 for the YearPredictionMSD dataset. The dimension n should be taken depending on what your code can support in terms of memory consumption.

A Julia code for generating this data can be found for example at `https://github.com/matthiasbe/block_srht` as well as potentially other useful codes for the project. Some of those matrices will be generated and used during the exercice sessions.

3 Content of the report (10 pages maximum, not including appendix)

The report should contain the following elements, that will guide the approach to use in the project.

Section 1: A description of randomized Nyström and of sketching matrices (Points: 1)

Your report should contain

- A presentation of the randomized Nyström low rank approximation algorithm considered in the project and the algebra on which the algorithm relies. The report should clearly explain how the rank-k approximation is obtained from the formula (1).
- A short description of the oblivious subspace embedding property and of the sketching matrices Ω used in the project. Two different sketching matrices should be used, among the different choices discussed during the Lecture from October 29, that is Gaussian and block SRHT (block subsampled randomized Hadamard transform).

Section 2: An investigation of the numerical stability of randomized Nyström (Points: 1.25)

An investigation of the numerical stability of randomized Nyström. For the data sets described in section 2.1, you should provide graphs that display the error of the low rank approximation in terms of nuclear norm (also known as trace norm) and computed as the sum of the singular values of the considered matrix. For the matrix A, this is computed as $\|A\|_* = \sigma_1(A) + \ldots \sigma_n(A)$). The error to be studied experimentally is $\|A - [\![A_{Nyst}]\!]_k\|_*/\|A\|_*$.

The discussion should compare as well the accuracy obtained for the two different sketching matrices by taking into consideration the sketching dimension.

This investigation should be done on the data sets provided in 2.1. Additional data could be provided but is not required.

Section 3: A presentation of the parallelization of randomized Nyström (Points: 1.25)

Describe how you parallelize randomized Nyström and provide pseudo-code of the parallel algorithm. For the parallelization, the matrix A should be distributed among processors by using a two-dimensional block distribution while the matrix Ω should be distributed using a block row distribution. You can consider that the number of processors is a power of 2 such that you can easily distribute the matrices among processors. For example, when $P = \sqrt{P} \times \sqrt{P} = 9$, the matrices A and Ω are distributed as:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}, \quad \Omega = \begin{pmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{pmatrix}$$

Section 4: A presentation of the sequential runtimes obtained by randomized Nyström (Points: 0.5)

A presentation of the sequential runtime obtained by the developed algorithm with the two different sketching matrices. Discuss if you observe any advantage in using a faster sketching operator with respect to the sketch dimension l that you might need in order to obtain an accurate low rank approximation.

Note that you can rely on optimized libraries for operations as matrix-vector multiplication, matrix-matrix multiplication, Cholesky factorization, sequential Householder QR, eigenvalue decomposition or singular value decomposition, Walsh-Hadamard transform.

As in the previous project, you should state the type of computer used for the experiments and the version of Python and other used libraries. You should plot the sequential runtimes obtained by the algorithms. They should be the average of a certain number of runs (for example 3 or 5). You should explain the plots, and explain if you were expecting these results and why.

Section 5: A discussion of the parallel performance (Points: 1)

You should present the parallel runtimes obtained by the developed algorithms on helvetios cluster. The number of processors can be small, up to 32 or 64 processors. The report should describe the machine used for the experiments and take into account the instructions given for reporting sequential times, that apply here as well. The report then should compare the runtime performance of the parallel algorithm with the two different sketching matrices, the scaling when increasing the number of processors, and their advantages and disadvantages should be explained in terms of parallel performance and numerical stability.

Generative AI: usage of generative AI, Large Language Models as chatGPT, needs to be acknowledged in the report. Explain the usage, as improving the english and formulation, generating some code (specify which algorithms and if this concerns their initial version), debugging the code, generating figures (specify which figures), generating text (specify which parts).

References

[1] O. Balabanov, M. Beaupere, L. Grigori, and V. Lederer. Block subsampled randomized hadamard transform for low-rank approximation on distributed architectures. In *ICML'23: Proceedings of the 40th International Conference on Machine Learning*, number 66, pages 1564–1576, 2023.

[2] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[4] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. *Advances in Neural Information Processing Systems*, 30, 2017.