Project 1 on *Orthogonalization techniques for a set of vectors*
HPC for numerical methods and data analysis

1 Goal of the project

The goal of this project is to study three different algorithms for the orthogonalization of a set of vectors. Those algorithms are Classical Gram-Schmidt (CGS), Cholesky-QR and TSQR. They are described in the Lecture Notes provided for the course and in the slides from Week 3 and Week 4. Such an algorithm will be needed in the second project where you will need to orthogonalize a set of vectors in parallel. Hence this project should allow you to identify an orthogonalization algorithm that is numerically stable and scales well in parallel.

2 Grading

The grade of the project has a weight of 0.3 for the final grade of the course. Each student works individually on the project and must submit a report and the code written in Python and MPI.

3 Deadline

The code and the report have to be submitted through Moodle before November 5, 2024, 11:59PM CEST. Strict deadline, no extension provided.

4 Orthogonalization algorithms

Consider a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, where $m \gg n$. The goal is to compute the thin QR factorization of $\mathbf{A}$, that is $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{R}^{m \times n}$ is orthogonal and $\mathbf{R} \in \mathbb{R}^{n \times n}$ is upper triangular. The matrix $\mathbf{A}$ is available at once. In this project, the three algorithms allow to compute the QR factorization of $\mathbf{A}$ and thus orthogonalize n vectors stored as columns of $\mathbf{A}$. Typically the number of vectors is in between 50 and a few hundreds, while m can be much larger.

5 Access to computers

During this class you have access to helvetios computer. For submitting your jobs, use the following option:

```
--account=math-505
```

6 Questions

Address all questions related to the content of the project to the Professor, either during the class, the breaks, in the forum, or by email.

7 Content of the report (8 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 short description of the three algorithms for computing the QR factorization. (Points: 0.75)

Your report should contain:

- A brief introduction of the project. Explain what is a QR factorization and why it is useful.
- For each of the algorithms, explain the algebra. For Classical Gram-Schmidt, explain what are the projectors used. For Cholesky-QR, explain the connection with the Cholesky factorization of $\mathbf{A}^T\mathbf{A}$ and why this is useful. For TSQR, explain why it minimizes communication, state the reduction tree used, state the algebra and the dimensions of intermediate $\mathbf{Q}$ factors.

Section 2: A presentation of their parallelization. (Points: 1.25)

For the parallelization, the matrix $\mathbf{A}$ should be distributed among processors by using a block row distribution. That is for example for 4 processors, the matrix is decomposed into 4 block rows as $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ \mathbf{A}_4 \end{bmatrix}$ and each processor i owns a block $\mathbf{A}_i$ with $i = 1 : 4$. Precise if you use 1-indexed matrices or 0-indexed as in Python and if processors are numbered from 1 to P or 0 to $P - 1$.

For each algorithm you should give their pseudo-code and explain it. You should also explain the communication routines used and why they are used. The Python and MPI code used for the implementation of the algorithms should be submitted as well.

Section 3: An investigation of their numerical stability. (Points: 1.25)

For a theoretical investigation, the bounds on the loss of orthogonality provided in the lectures should be used. For the numerical investigation, you should provide plots measuring the loss of orthogonality as $\|I - Q^TQ\|_2$ and the condition number of the basis Q. When it is possible, these quantities, along with the condition number of the input vectors, should be provided at each iteration of the algorithm when a new vector is orthogonalized. However this will not be possible for Cholesky-QR or TSQR for example. In this case you should present only the results obtained at the end of the algorithm.

This investigation should be done on at least two different matrices. One of these matrices will be generated by uniformly discretizing a parametric function, also used in [1], that we denote $C \in \mathbb{R}^{m\times n}$. For all floating (and thus rational) numbers $0 \leq x, \mu \leq 1$, the function is defined as

$$f(x, \mu) = \frac{\sin(10(\mu + x))}{\cos(100(\mu - x)) + 1.1}$$

and the associated matrix is

$$C \in \mathbb{R}^{m\times n},\ \ C(i, j) = f\left(\frac{i-1}{m-1}, \frac{j-1}{n-1}\right),\ i \in 1, \ldots, m, j \in 1, \ldots, n. \tag{1}$$

You could use for example $m = 50000$ and $n = 600$.

Other matrices could be obtained for example from the SuiteSparse Matrix Collection, `https://sparse.tamu.edu/about` . You can pick any square matrix, and then select only the first columns. The dimensions of the test matrices should be chosen such that you can compute the required quantities.

Your report should state the theoretical bounds for the loss of orthogonality of the three algorithms. You should explain why these measures are important. For each matrix you should plot or put in a table the loss of orthogonality/condition number. You should explain these plots, reference them, and make connection with the theory.

Section 4: A presentation of the sequential runtimes obtained by the three algorithms. (Points: 0.5)

Note that you should rely on optimized libraries for operations as matrix-vector multiplication, matrix-matrix multiplication, Cholesky factorization, sequential Householder QR. The report should contain:

- State the type of computer used for the experiments and the version of Python and other used libraries.
- Plot sequential runtimes obtained by the algorithms for the different matrices. They should be the average of a certain number of runs (for example 3 or 5). Explain the plots.
- Explain if you were expecting these results and why.

Section 5: A discussion of their parallel performance. (Points: 1.25)

You should present the parallel runtimes obtained by the three 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 runtimes, that apply here as well. The report then should compare the runtime performance of the three algorithms, their scaling when increasing the number of processors, and their advantages and disadvantages should be explained in terms of parallel performance and numerical stability.

Overall in the project pay attention to:

- The text flows and is clearly organized into sections and paragraphs. Any pseudo code, figures, or code included within the text is referenced.
- The plots make use of color that helps the reader undertand better its objectives.
- The axis of the plots are labelled, the (psuedo) code has a title. The axis are in log scale if appropriate.
- Have clear references to either papers or lectures.
- The project is well organized, there is a title, name of the author, sections, etc.

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 and L. Grigori. Randomized gram–schmidt process with application to gmres. *SIAM Journal on Scientific Computing*, 44(3):A1450–A1474, 2022.