

# Norm1

This exercise is about parallelizing the computation of the 1-norm of a matrix  $A \in \mathbb{R}^{m \times n}$  which is defined as

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_i^m |a_{ij}|$$

Assuming  $A$  is represented as a 2D array, this can be computed with a doubly nested loop in a column-major style (i.e., the outer loop is over columns) or row-major style (i.e., the outer loop is over rows). Because, in the C language, 2D arrays are stored in row-major format, the row-major code is likely to achieve better performance because of a better use of cache memories.

## 1 Package content

In the `norm1` directory you will find the following files:


- `main.c`: this file contains the main program that first executes a sequential reference code `norm1` and then calls the column-major and row-major sequential routines, respectively `norm1_colmajor` and `norm1_rowmajor`. **Only this file has to be modified for this exercise.**
- `aux.c`, `aux.h`: these two files contain auxiliary routines and **must not be modified.**


The code can be compiled with the `make` command: just type `make` inside the `norm1` directory; this will generate a `main` program that can be run like this:

```
$ ./main m n
```

where `m` and `n` are the number of rows and columns in the  $A$  matrix.

## 2 Assignment

-  Parallelize the `norm1_colmajor` and `norm1_rowmajor` routines. Make sure that the result computed by the two parallel variants is consistent with the reference sequential code.

-  Report, in the form of code comments directly in the `main.c` file, execution times for the sequential code and the two parallel variants using 1, 2 and 4 threads. Which parallel version is fastest?