# HPC 4MA 2021/2022

**Members**

PHAM Tuan Kiet

VO Van Nghia

**Date**

12 Dec, 2021

# Contents

# Chapter 1

# OpenMP

## 1.1 Optimization techniques

### 1.1.1 Naive dot

We first mention here the original `naive_dot` function. This function serves as an anchor (or base case) for performance comparision as well as for making sure we have the right result when using other techniques.

```
  for (i = 0; i < M; i++)
    for (j = 0; j < N; j++)
      for (k = 0; k < K; k++) C[i + ldc * j] += A[i + lda * k] * B[k +
   ldb * j];
```

Below is the output of `naive_dot` for `M = 1`, `K = 2` and `N = 2`:

```
## ( 1.00  1.50 )
##
## ( 1.00  1.50 )
## ( 1.50  2.00 )
##
## Frobenius Norm   = 5.550901
## Total time naive = 0.000001
## Gflops           = 0.013333
##
## ( 3.25  4.50 )
```

As

$$\begin{pmatrix} 1 & 1,5 \end{pmatrix} \begin{pmatrix} 1 & 1,5 \\ 1,5 & 2 \end{pmatrix} = \begin{pmatrix} 3,25 & 4,5 \end{pmatrix}$$

The result of this function is correct. We move on to the next technique.