A03 - Assignment 3

Student: Deidier Simone - 133020

Answers to the theory questions

1. To determine the acceleration (*speed-up*) obtained by parallelising the code, it is necessary to compare the execution time of the parallel code with that of the sequential code. Speed-up is defined as:

$$\text{Speed-up} = \frac{T_{\text{sequential}}}{T_{\text{parallel}}}$$

Where $T_{\text{sequential}}$ is the execution time of the sequential code, and T_{parallel} is the execution time of the parallel code. On my laptop, the $T_{\text{sequential}}$ is 1.973315 *seconds*, using $n = 4$ processes, the T_{parallel} is 0.747056 *seconds*, so the speed-up is $\frac{1.973315}{0.747056} \approx 2.641$, using $n = 13$ processes, the T_{parallel} is 1.088252 *seconds*, so the speed-up is $\frac{1.973315}{1.088252} \approx 1.813$.

2. This type of parallelisation is known as ***domain decomposition parallelisation***. In this approach, the problem domain (*e.g. a grid of points in a numerical simulation*) is divided into smaller sub-domains, each assigned to a parallel process.
3. *Point-to-Point communication* involves only two processes: a *sender* and a *receiver*, common examples include send and receive operations, it is used to transfer data directly from one process to another, it is more flexible, but can be more complex to handle in applications with many processes. *Collective communication*, on the other hand, involves a group of processes, often all processes in a communicator, common examples include *broadcast*, *scatter*, *gather* and *reduce* operations. It is used for operations requiring the participation of several processes, such as the distribution of data or the collection of results, and is simpler to use for common operations involving several processes, but may be less flexible than point-to-point communication.
4. In the code `int* a, b;` the variable b is of type **int**, in fact a is declared as a pointer to an integer (*int**), while b is declared as an integer (*int*). The code `int* a, b;` does not mean that both variables a and b are pointers to integers, if we wanted to declare both variables as pointers to integers, we should write `int *a, *b;`.