# Computação Paralela e Distribuída

## Primeira entrega

## Relatório (OpenMP)

### Grupo XX:

### Beatriz Marques, XXXXX

### Carlos Carvalho, XXXXX

### Diogo Nunes, 85184

# Introdução

O objetivo da primeira entrega é paralelizar (num modelo de memória partilhada) a versão sequencial do projeto utilizando a ferramenta OpenMP.

Por forma a realizar este trabalho, o grupo tomou os seguintes passos:

1. Identificação das zonas de código computacionalmente pesadas;
2. Iterativamente sobre cada uma das zonas supramencionadas,
   1. Implementação da respetiva paralelização;
   2. Teste das alterações sobre os exemplos *input-output* fornecidos;
   3. Análise dos resultados produzidos (relatório da ferramenta OpenMP Profiler) sobre casos computacionalmente pesados;
   4. Conclusão sobre a relevância das alterações;
3. Análise e conclusão sobre resultados finais produzidos (relatório da ferramenta OpenMP Profiler) sobre os casos computacionalmente pesados;

Os passos referidos acima são desenvolvidos sequencialmente no resto do relatório.

# 1. Identificação das zonas de código computacionalmente pesadas

Pela formalização do problema, as únicas direções de escalabilidade a considerar são o número de partículas do sistema (**n**), o número de células (**c**) e o número de *time steps* (**t**)*.*

Por natureza, a computação de cada *time step* não pode ser paralelizada, já que o *time step* **t** depende do anterior, **t-1**. Por esta razão, o foco do trabalho está sobre as duas restantes variáveis.

Da versão sequencial do grupo, as seguintes zonas foram identificadas como computacionalmente pesadas (respetivo custo de execução entre parêntesis retos):

1. Inicialização das partículas [**n**]
2. Criação da matriz de células [**c**]
3. Inicialização da matriz de células [**c2**]
4. Para cada *time step* ***t****,*
   1. Determinação do centro de massa de cada célula [**n + c**]
   2. Computação da força gravítica apicada a cada partícula [**9n**]
   3. Cálculo da nova velocidade e posição de cada partícula [**n**]
   4. Inicialização dos valores para a próxima iteração [**n + c**]
5. Cálculo do centro de massa do sistema [**n**]
6. Libertação de memória [**c**]

# 2. Iterativamente sobre cada uma das zonas supramencionadas

## 2.1. Implementação da respetiva paralelização

Das zonas identificadas anteriormente, a (1.) inicialização de partículas não deve ser paralelizada devido à utilização de funções *random* (*thread safe*). É também de notar que em cada uma destas zonas é executado exatamente o mesmo conjunto de procedimentos, ou para cada partícula, ou para cada célula, dependendo do contexto – o que significa que é necessária muito pouca sincronização de acessos à memória.

Então, tomando partido deste *data parallelism*, cada uma das zonas (exceções descritas à frente) pode ser paralelizada utilizando a simples diretiva:

*#pragma omp parallel for*

Existem apenas duas zonas de exceção à afirmação acima, (4.1.) determinação do centro de massa de cada célula e o (6.) cálculo do centro de massa do sistema. A razão de exceção está no facto de ser necessário obter, para cada célula, os valores cumulativos de **x, y** e **m** (respetivamente, posição bidimensional e massa) de cada partícula que nela está inserida.

É, portanto, necessário que cada *thread* tenha uma cópia local destas variáveis, execute a operação cumulativa sobre as partículas que lhe são designadas, e que no fim, seja efetuada uma operação de *reduction* para obter o resultado correto.