Uma imagem com fogo de artifício, escuridão, espaço

Descrição gerada automaticamente

**Simulação e Modelação Computacional em Engenharia Física**

Relatório Laboratório #01

Grupo:

Daniela Marques, 59939, MIEF

Tiago Neves, 59249, MIEF

Rui Filipe, 60461, MIEF

**Introduction (**About one page)

Objective of the simulation - DANI

What physical phenomena are we trying to simulate, with a concise description - DANI

**Code**

Explain the code structure - DANI

How it works (what information it needs and it’s given to the code) - DANI

Any decision or considerations you had to decide for the v0 code – RUI

* Se variar a temperatura ou o campo devo começar com a rede inicializada sempre da mesma forma ou no novo valor de t ou h começar com a rede como terminou o valor anterior? Vantagens e desvantagens
* Que números de ciclos iniciais se devem não contabilizar para efeitos de obtenção de valores? 10%, valor fixo, outros?
* Tamanho da rede, fronteiras abertas ou fechadas?

Eventually, preliminary results obtained so far - NEVES

**Computing Complexity**

Computer complexity analysis - RUI

Profiler measurements - NEVES

What the group takes from that information - NEVES

1. **transitionFunctionValues**:
   * This function iterates over a fixed range (-6 to 6) and performs constant time operations within the loop. Therefore, its time complexity is O(1).
2. **init**:
   * This function initializes an array with a given size. As it only involves creating an array and filling it with a constant value, its time complexity is O(n^3), where n is the size of the grid.
3. **cycle**:
   * This function contains nested loops iterating over the grid size three times. Within the innermost loop, there are constant time operations. So, the overall time complexity is O(n^3), where n is the size of the grid.
4. **sim**:
   * This function contains a loop iterating over the number of cycles. Inside the loop, it calls the **cycle** function, which has a time complexity of O(n^3). Therefore, the overall time complexity of this function is O(num\_cycles \* n^3).
5. **changingT**:
   * This function iterates over the temperatures array and calls the **sim** function for each temperature. Since the **sim** function has a time complexity of O(num\_cycles \* n^3), the overall time complexity of this function is O(num\_cycles \* n^3 \* points), where points is the size of the temperatures array.
6. **changingH**:
   * This function iterates over the fields and temperatures arrays and calls the **sim** function for each field. Similar to **changingT**, the overall time complexity of this function is O(num\_cycles \* n^3 \* pointsT \* pointsH), where pointsT is the size of the temperatures array, and where pointsH is the size of the fields array.
7. **hysterisis**:
   * This function calls **changingH** and plots the results. Its time complexity depends on the time complexity of **changingH**, which is O(num\_cycles \* n^3 \* pointsT \* pointsH).

Overall, the dominant factors affecting time complexity in this code is the size of the grid (size^3) and the number of cycles (num\_cycles).

**Optimization**

What were and why the optimization improvements were implemented - RUI

How much did those code changes improve the execution time - RUI

Justify what parameters and meta parameters will be used in the final simulation - RUI

**Results**

Report of the results obtained with v1 code version for the more demanding input size - NEVES

Analyze and compare them with the expected results - NEVES

Short information on what were the main difficulties of the project, and what could eventually be improved - NEVES