

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA  
DE MINAS GERAIS

ENGENHARIA DE COMPUTAÇÃO

---

**Paralelização de Métodos Numéricos para  
Resolver Equações Diferenciais Parciais**

---

*Orientando:*

Marcelo Lopes de Macedo  
FERREIRA CÂNDIDO

*Orientador:*

Prof. Dr. Luis Alberto  
D'AFONSECA

BELO HORIZONTE  
17 de fevereiro de 2019

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Aquisições Sísmicas</b>	<b>3</b>
2.1	O Que São Aquisições Sísmicas e Como Modelá-las . . . . .	3
2.2	A Equação da Onda . . . . .	4
2.3	O Método de Diferenças Finitas (MDF) . . . . .	4
<b>3</b>	<b>A arquitetura computacional atual e a necessidade de paralelização</b>	<b>5</b>
3.1	A arquitetura de von Neumann . . . . .	5
3.1.1	O gargalo de von Neumann . . . . .	6
3.2	O processador . . . . .	6
3.3	A memória principal . . . . .	6
3.4	A cache . . . . .	6
3.5	Disco rígido . . . . .	6
3.6	Conseguir mais em menos tempo . . . . .	6
3.6.1	A barreira da memória - <i>Memory wall</i> . . . . .	7
3.6.2	A barreira do paralelismo a nível de instruções - <i>ILP wall</i> . . . . .	7
3.6.3	A barreira no gasto de energia dos processadores - <i>Power wall</i> . . . . .	7
3.7	Paralelismo - a alternativa para se contornar as barreiras . . . . .	7
3.7.1	Arquiteturas de memória na computação paralela . . . . .	7
3.7.2	Modelos da computação paralela . . . . .	7
3.7.3	Desenhando programas paralelos . . . . .	8
<b>4</b>	<b>Paralelismo em prática</b>	<b>9</b>
4.1	OpenMP . . . . .	9
4.2	Pthreads . . . . .	9
4.3	MPI . . . . .	9
<b>5</b>	<b>Do serial ao paralelo</b>	<b>10</b>
5.1	A construção do código serial . . . . .	10
5.2	Primeiro contato do código com as <i>threads</i> - OpenMP . . . . .	10
5.2.1	Construindo o caminho para a OpenMP . . . . .	10
5.3	Um contato mais profundo do código com as <i>threads</i> - Pthreads . . . . .	10
5.3.1	Construindo o caminho para a Pthreads . . . . .	10
5.4	Realizando a mescla de Pthreads e MPI . . . . .	10
5.4.1	Contruindo o caminho para a mescla . . . . .	10
<b>6</b>	<b>Considerações finais</b>	<b>11</b>

# **Capítulo 1**

## **Introdução**

# Capítulo 2

## Aquisições Sísmicas

### 2.1 O Que São Aquisições Sísmicas e Como Modelá-las

No ramo da mineração não se pode tentar a esmo a descoberta de recursos minerais no subterrâneo de um local em que se já se suspeita sua existência. É necessário que, de alguma forma, se obtenha a forma dessa estrutura oculta para se saber os pontos onde se encontram as jazidas/poços desse recurso. A obtenção dos dados de como é essa estrutura se chama **aquisição sísmica**.

A forma de se realizar essa aquisição pode variar com o ambiente e os métodos adotados para coleta de dados e processamento dos mesmos. Os recursos minerais desejados podem se encontrar tanto em meios terrestres e/ou subaquáticos. Contudo, o meio em que a aquisição será realizada pouco importa nesse trabalho, o que será melhor explicado mais a frente.

Para a coleta dos dados a serem processados podemos citar dois exemplos de aquisição

buscar referências

1. **marítima**: um navio equipado com um canhão sonoro emite ondas sonoras cujas reflexões e refrações nas camadas terrestres submarinas são captadas por filas de hidrofones puxadas pelo mesmo navio.
2. **terrestre**: um explosivo é colocado (preferencialmente enterrado) em um terreno. Sua explosão gera uma onda sonora cujas reflexões são captadas por geofones distribuídos relativamente próximos, na superfície.

Costuma-se alterar a posição da fonte sonora na realização da aquisição para facilitar a modelagem do meio de propagação das ondas.

Esse colhimento dos dados consistirá em traços, que são o gráfico das amplitudes das ondas sonoras captadas pelos hidrofones/geofones, como se pode ver na imagem ???. A partir desses traços, detecta-se, por análise técnica, onde se encontram as interfaces entre as camadas do domínio analisado e do que são feitas essas. Nisso consiste o processamento dos dados colhidos.

Vale lembrar que, quanto aos métodos de processamento utilizados, nesse trabalho simularemos um problema direto, ou seja, estipulando o meio (nesse caso, não-homogêneo) da aquisição, veremos como as ondas se propagam nele. Para tal, usaremos um método matemático específico. Nesse trabalho, como já foi dito no Capítulo 1, é o de Diferenças Finitas.

conferir  
se  
pro-  
cede

colocar  
isso  
no  
Ca-  
pí-  
tulo  
1

## 2.2 A Equação da Onda

Para que seja possível avaliar matematicamente um fenômeno ondulatório produzido por uma fonte em um domínio é necessário se ter uma fórmula matemática para o que se entende por onda. Tal fórmula, que nos servirá durante todo esse trabalho, principalmente na parte do código é a **equação da onda**, dada por

$$\frac{\partial^2 u}{\partial t^2} = \left(\frac{1}{v^2}\right) \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x, y, t) \quad (2.1)$$

onde  $x$  e  $y$  são variáveis espaciais e  $t$ , temporal. A constante (no caso desse trabalho)  $v$  representa a velocidade da frente de onda. Trata-se de uma **equação diferencial parcial hiperbólica** com solução analítica, mas que pode ser resolvida de forma fácil numericamente, utilizando, por exemplo, o **método de diferenças finitas**, a ser mais explicado na próxima seção.

Caso o leitor se interesse por estudar ou revisar mais sobre Ondulatória, pode conferir em Cândido [mfcandido2018].

## 2.3 O Método de Diferenças Finitas (MDF)

Uma equação diferencial parcial, que geralmente é considerada em um domínio contínuo, pode ser discretizada. Isso é feito para que a equação possa ser representada e resolvida computacionalmente.

No caso do método de diferenças finitas para esse trabalho, basta transcrever cada termo da equação para o equivalente na fórmula de diferenças finitas para derivações de segundo grau, que é

$$\frac{\partial^2 u}{\partial x^2} \quad \left| \quad \frac{u_{i-1} - 2u_i + u_{i+1}}{2\Delta x^2} \right.$$

## Capítulo 3

# A arquitetura computacional atual e a necessidade de paralelização

Para compreender a questão da paralelização envolvida nesse trabalho é necessário entender de onde e porque ela veio. Para tal, é necessário se apresentar as principais peças que constituem um computador moderno, os problemas que surgiram no processo de evolução da *arquitetura computacional* atual e como isso culminou no paralelismo [LLNL:parcomp].

Antes de iniciar esse processo, é também necessário se explicar o que se entende por arquitetura computacional. Trata-se da área do conhecimento que estuda a interface entre *software* e *hardware*, desde o mais baixo nível, no qual o processador manipula as informações (instruções e dados) entregues a ele, para toda operação realizada no computador; passando pelas políticas de manipulação de dados nas memórias cache (e seus níveis), de acesso aleatório (RAM - *random access memory*) e de armazenamento não-volátil (discos rígidos, por exemplo); chegando também à interação dos computadores com os demais periféricos que por ventura estão nele conectados, realizando *inputs* (entradas) e/ou *outputs* (saídas), também conhecidas pela abreviação "I/O", como teclado, *mouse*, monitor, etc [Catsoulis].

Ainda no âmbito da arquitetura de computadores, são estabelecidas análises e metas de performance. Por exemplo, tenta-se determinar se um processador A é mais rápido que um B (sendo B diferente de A) comparando-se o número de instruções que cada um processa, ou quantos ciclos (que serão explicados mais a frente) podem ser dados em um segundo [wiki:comparch]. Questão semelhante encontra-se na computação de alta performance<sup>1</sup>: quão mais rápido um trecho de código executa ao ser paralelizado.

Essa organização focada em um processador, memórias cache, RAM e de armazenamento não-volátil vieram da **arquitetura de von Neumann**, que possui esse nome por conta de seu idealizador, John von Neumann.

### 3.1 A arquitetura de von Neumann

- von Neumann e o conceito de programa armazenado;
- a forma da arquitetura (componentes e ligações);

<sup>1</sup>**computação de alta performance** (ou **supercomputação**) é um ramo da computação que usa supercomputadores e estuda técnicas de paralelismo visando alcançar velocidades de processamento maiores (do que as de computadores normais, como os pessoais) para a resolução de problemas computacionais complexos, como simulações, modelagens e análises computacionais. Esse ramo também pode, no lugar de focar em processamento mais rápido, resolver problemas de dimensões maiores, que não poderiam ser resolvidos em tempo hábil por computadores comuns [techo:hpc]

conferir  
se  
deve  
tratar  
como  
atual

- o contraste da arquitetura idealizada com as anteriores.

### **3.1.1 O gargalo de von Neumann**

- o gargalo existente entre o processador e a memória, que não consegue acompanhá-lo.

## **3.2 O processador**

- introdução ao que é um processador;
- apresentação de um processador MIPS simples;
  - entradas;
  - módulos internos;
  - saídas;
- incrementos que os processadores receberam ao longo da história (ILP).

## **3.3 A memória principal**

- introdução do que é uma memória;
- apresentação básica dos tipos de memória (ênfase na RAM);
- apresentação básica das propriedades de uma memória RAM.

## **3.4 A cache**

- O princípio da localidade no tempo e no espaço;
- A necessidade da existência de uma cache, visto o princípio da localidade;
- Explicação do que é uma cache;

## **3.5 Disco rígido**

- O que são memórias voláteis e exemplos delas;
- A necessidade de memórias não-voláteis e o HD;
- Outras diferenças entre o HD e as outras memórias envolvidas no trabalho

## **3.6 Conseguir mais em menos tempo**

- A necessidade de se aumentar a velocidade de processamento;
- introdução aos problemas encontrados na computação serial

### 3.6.1 A barreira da memória - *Memory wall*

Conferir se existe mesmo a memory wall

Revisar o que é a memory wall e completar o itemize abaixo

- 

### 3.6.2 A barreira do paralelismo a nível de instruções - *ILP wall*

- Explicar do que se trata o paralelismo a nível de instruções;
- Explicar os conceitos de superpipeline e superescalar;
- Explicar que, ao se estender muito um pipeline, temos problemas;
- Explicar os problemas da superescalaridade

### 3.6.3 A barreira no gasto de energia dos processadores - *Power wall*

- Introduzir a lei de Moore;
- Explicar que houve evolução na taxa de clock ao longo do tempo;
- Explicar que essa evolução foi amortecida nos últimos anos devido ao gasto energético e às altas temperaturas que os processadores alcançaram

## 3.7 Paralelismo - a alternativa para se contornar as barreiras

- Explicar no que consiste o paralelismo;
- em seguida, explicar porquê ele é uma saída possível

### 3.7.1 Arquiteturas de memória na computação paralela

- Explicar as arquiteturas:
  - de memória compartilhada;
  - memória distribuída;
  - híbrida

### 3.7.2 Modelos da computação paralela

Conferir se serão esses modelos a serem apresentados

- Shared Memory Model;
- Threads Model;
- Distributed Memory / Message Passing Model;



- Data Parallel Model;
- Hybrid Model

### 3.7.3 Desenhando programas paralelos

ler a respectiva seção e completar aqui

-

# **Capítulo 4**

## **Paralelismo em prática**

**4.1** OpenMP

**4.2** Pthreads

**4.3** MPI

# Capítulo 5

## Do serial ao paralelo

### 5.1 A construção do código serial

### 5.2 Primeiro contato do código com as *threads* - OpenMP

#### 5.2.1 Construindo o caminho para a OpenMP

### 5.3 Um contato mais profundo do código com as *threads* - Pthreads

#### 5.3.1 Construindo o caminho para a Pthreads

### 5.4 Realizando a mescla de Pthreads e MPI

#### 5.4.1 Construindo o caminho para a mescla

## **Capítulo 6**

### **Considerações finais**