

AGA 511

Métodos Computacionais em Astronomia

Segundo semestre de 2023

Informações gerais

Prof. Alex Cavaliéri Carciofi

Email: carciofi@usp.br

Monitora:

- Pâmela Querido

Site do e-disciplinas

- Algumas aulas gravadas

Google Collab

- O collab vai uma ferramenta essencial no curso!
- Esta semana colocarei um tutorial no Moodle para quem não tiver familiaridade com esta plataforma

Programa resumido

1. Programação (boas práticas, algoritmos científicos, etc.)
2. HPC - High Performance Computing - conceitos básicos
3. MPI - Message passing interface
4. OpenMP - Open Multi-Processing
5. CUDA - Compute Unified Device Architecture
6. OpenACC

Proposta Geral

Curso não tradicional:

- Poucas aulas expositivas (sempre às quartas)
- Aulas de segunda-feira (12h as 13h30) - Plantão no laboratório de informática do bloco B
- Focado em trabalhos práticos e estudos dirigidos
- Project-oriented (orientado por projeto)
- PBL - problem-based learning

Proposta Geral

1. **EP1**: agulha de Buffon (básico de método MC, números aleatórios, etc.)
2. Escolher um **projeto interessante**, mas curto o suficiente para que possa ser resolvido em poucas semanas (**próxima aula!**)
3. **EP2**: Escrever um programa tradicional (**serial**) em C++, Fortran ou Python que resolva este problema
 - * Exercício individual. 100% do código desenvolvido do zero pelo aluno
 - * Código bem estruturado: comentado, compartimentalizado, eficiente
 - * Entrega de monografia parcial (ver adiante)

Proposta Geral

4. Formação das **equipes de trabalho**, em que cada uma estudará uma das técnicas abaixo:
 - a) **MPI - C++ ou FORTRAN**
 - b) **OpenMP - C++ ou FORTRAN**
 - c) **CUDA - C++**
 - d) **OpenACC**
5. **Em equipes**: apresentação de uma visão geral sobre as técnicas para a turma (1h)
6. Individualmente:
 - refazer ambos os programas: Buffon (**EP3**) e projeto (EP4) incorporando técnicas de HPC
 - Apresentar projeto à turma (10 min)
 - Preparar TCD - Trabalho de Conclusão de Disciplina

Calendário

- 09/08 - Aula de Apresentação - Agulha de Buffon
- 16/08 - Apresentação de ideias para os trabalhos individuais
- 23/08 - Definição das equipes e dos projetos individuais
- 30/08 - **Aula 1** - Entrega do EP1
- 13/09 - **Aula 2**
- 20/09 - **Aula 3**
- 27/09 - **Aula 4 - Entrega do EP2**
- 04/10 - Reunião com a equipe de OpenMP
- 11/10 - Reunião com a equipe de MPI
- 18/10 - Reunião com a equipe de CUDA
- 25/10 - Reunião com a equipe de OpenACC - **Entrega do EP3**
- 01/11 - **Apresentação da Equipe OpenMP + trabalhos individuais**
- 08/11 - **Apresentação da Equipe MPI + trabalhos individuais**
- 22/11 - **Apresentação da Equipe CUDA + trabalhos individuais**
- 29/11 - **Apresentação da Equipe OpenACC + trabalhos individuais**
- 08/12 - **Entrega do TCD**

Avaliação

- EP 1 (**10%**) - O problema da agulha (Buffon) - versão serial
- EP 2 (**25%**) - Versão serial do projeto + monografia inicial
- EP3 (**10 %**) - Buffon em versão paralela e análise de Amdahl
- Apresentação em equipe (**20%**)
- Trabalho de Conclusão de Disciplina (**35%**)
 - Apresentação individual
 - Entrega de monografia

Frequência

- Entregáveis (EP1, EP2, EP3, etc...)
- Frequência nas reuniões com os grupos
- Frequência nas apresentações dos grupos e nas apresentações individuais

EP2 - Versões seriais do projeto (30% da nota)

- Entrega de uma pequena monografia (2+ páginas) com:
 - descrição do projeto escolhido
 - descrição detalhada do algoritmo empregado
 - Resultados (gráficos, tabelas)
- Entrega do código fonte:
 - Bem comentado
 - Compartimentalizado
- **Importante:** esta primeira entrega será uma prévia do TCD

Reuniões com as equipes

- Discussão sobre a técnica de paralelismo
- Prévia da apresentação para o grupo
- Lembrando: a apresentação da equipe vale 20% da nota final

Trabalhos Individuais (35% da nota)

- Entrega de uma monografia que será uma extensão da primeira, com os seguintes pontos adicionais:
 - Resumo da técnica de paralelismo empregada
 - Descrição da implementação da técnica de paralelismo
 - Testes de performance: lei de Amdahl, aceleração, etc.
- Entrega do código fonte:
 - Bem comentado
 - Compartimentalizado

Resolução numérica de problemas

O motivo para escrever um programa, qualquer programa, é **resolver um problema**. A tarefa de escrever um problema pode ser dividida em três partes:

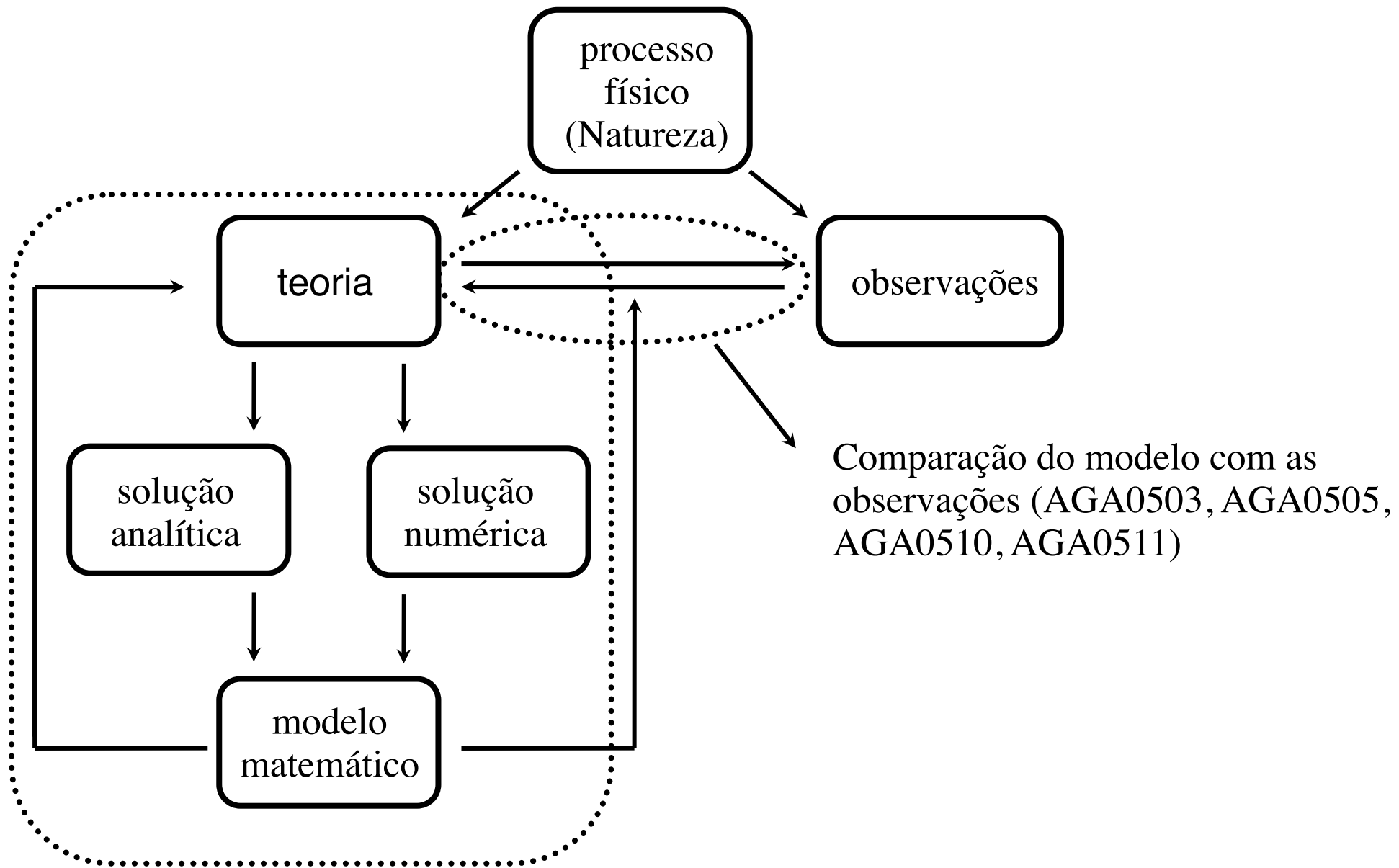
1. **Especificar o problema de forma clara** (difícil!)
2. Analisar o problema e **reduzi-lo em seus elementos fundamentais**
3. **Codificar o programa** de acordo com o plano desenvolvido no passo 2

Frequentemente há um quarto passo que, via de regra, é o mais difícil de todos:

4. Testar o problema exaustivamente, e repetir passos 2 e 3 até que o programa funcione corretamente em todas as situações previstas.

Método numérico: conjunto de regras escritas sob a forma de uma sequência de operações elementares que levam à uma solução do problema.

Em seu nível mais fundamental (ver capítulo 2), utiliza-se somente as quatro operações aritméticas $+$, $-$, \times , \div



Criação de um modelo matemático (física, cálculo, fismat, AGA0503, AGA0510)