

Introdução

Disciplina: Métodos Numéricos em Termofluidos

Professor: Adriano Possebon Rosa

Departamento de Engenharia Mecânica

Faculdade de Tecnologia

Universidade de Brasília

1 MNT

Em Métodos Numéricos em Termofluidos nós vamos estudar problemas básicos de Transferência de Calor e de Mecânica dos Fluidos a partir de uma abordagem numérica. Vamos resolver a equação do calor

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \quad (1)$$

e as equações de Navier-Stokes,

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

e

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} , \quad (3)$$

para um fluido incompressível.

Vamos utilizar o método das diferenças finitas para resolver o clássico problema da cavidade cisalhante.

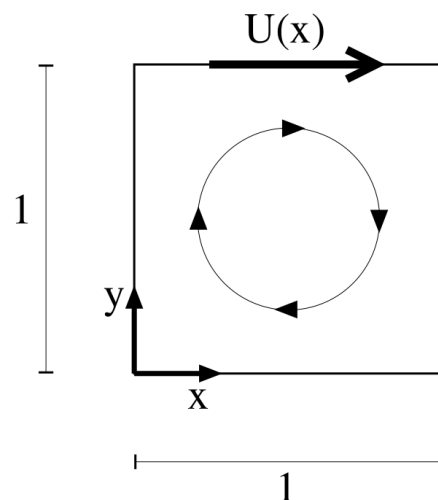


Figura 1: Cavidade cisalhante.

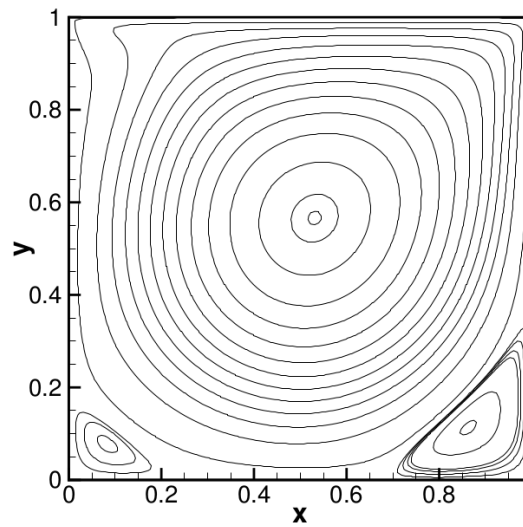


Figura 2: Linhas de corrente.

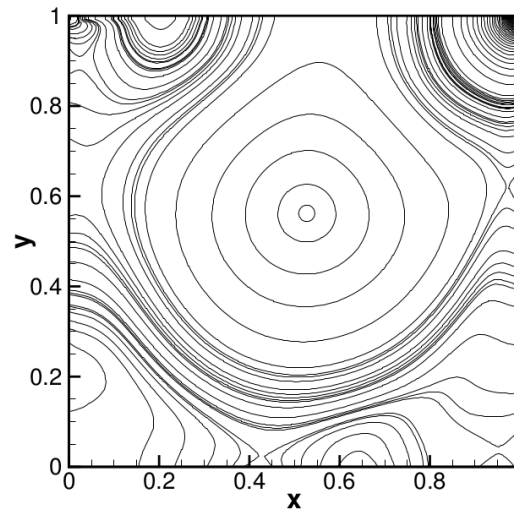


Figura 3: Pressão.

Pra quem é esse curso? Para quem já fez alguma matéria de Mecânica dos Fluidos (Mecânica dos Fluidos 1, Fenômenos de Transporte, Transporte de Calor e Massa) e quer aprender um pouco mais sobre Simulação Numérica de Fluidos.

Não é necessário nenhum conhecimento prévio de programação: vamos começar do zero.

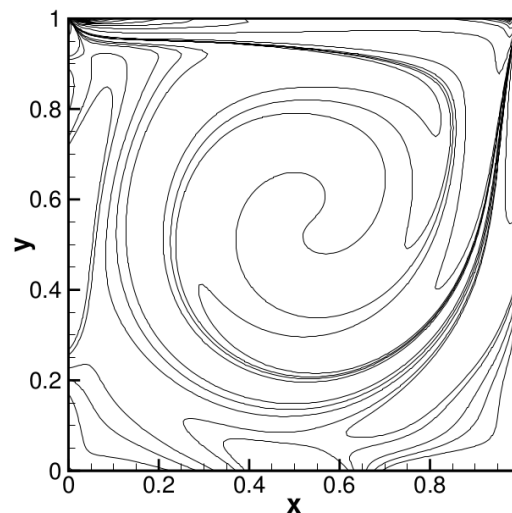


Figura 4: Vorticidade.

2 Objetivos Principais

Os objetivos principais neste curso são:

- aprender a programar, com foco em Métodos Numéricos;
- entender um pouco mais sobre Fluidos e Transferência de Calor;
- desenvolver um código próprio;
- ter uma noção introdutória de como funcionam os programas de CFD (Computational Fluid Dynamics);
- escolher um problema e escrever um artigo científico.

Muitos algoritmos que vamos desenvolver aqui já estão implementados no Python. Então por que aprender a desenvolver esses algoritmos?

Primeiro porque eles formam a base para o desenvolvimento de programas muito mais complexos, que não estão disponíveis. Segundo, porque o objetivo aqui é desenvolver habilidades, aprender a programar.

Não vamos nos aprofundar muito na teoria, neste curso. Vamos focar mais na parte de implementação.

3 Plano de Ensino

Vamos dar uma olhada no Plano de Ensino do nosso curso.

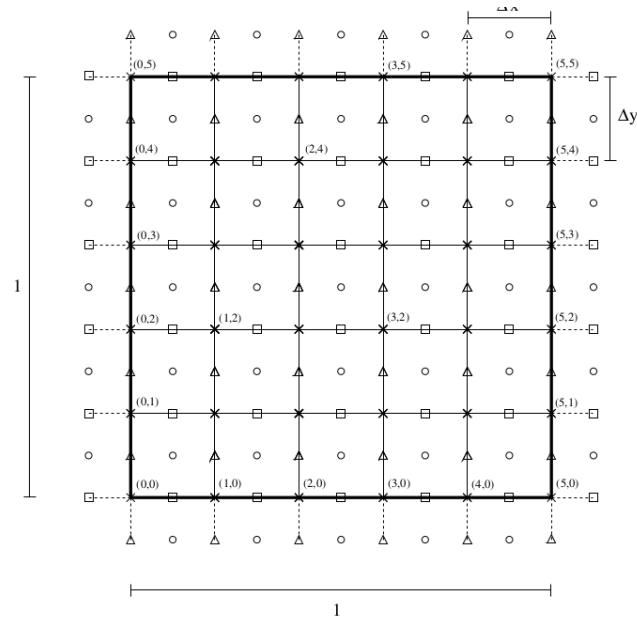


Figura 5: Domínio computacional.

4 Atividades

Teremos dois tipos de atividades que valem nota:

- Trabalhos: serão 4 ao longo do semestre;
- Artigo Científico: 1, no final do curso.

Trabalhos iguais terão nota zero. Vocês podem discutir as questões dos trabalhos, claro, mas os trabalhos são individuais.

Os códigos poderão ser anexados ao relatório do trabalho, no próprio pdf, ou enviados em arquivos .py separados.

As datas para a entrega desses trabalhos serão combinadas ao longo do curso. Já dá pra ter uma noção pelo Cronograma.

O Artigo Final será entregue e apresentado na última semana.

5 Linguagens de Programação

- C
- C++
- Fortran
- Matlab
- Julia

- R
- Lua
- Java
- Ruby
- Python

Existem diversas linguagens, cada uma com seus pontos positivos e negativos. Você pode escolher qualquer linguagem de programação para utilizar aqui no nosso curso, fique à vontade.

A linguagem não é o mais importante. A lógica de programação e o algoritmo são os pontos principais.

Vou usar o **Python** aqui no nosso curso.

Pontos positivos: open source, é relativamente mais fácil de aprender, códigos mais curtos, fácil de entender um código pronto. Python é uma linguagem usada em vários tipos de aplicações, não apenas em simulação numérica.

Pontos negativos: é muito lenta (por ser uma linguagem interpretada) quando comparada a outras linguagens como Fortran, C, C++ e Julia (compiladas). Mas nós vamos ver como resolver esse problema....

Pra dar sorte, o primeiro programa tem que ser escrever *Hello World* na tela.

```
print("Hello World!")
```

Outro exemplo:

```
x = 3.0
print(x)
x = x + 4
print(x)
```

Vamos ver uma introdução rápida ao Python, focando principalmente no que nós vamos usar aqui no curso.

6 Dicas

Aprender a programar não é fácil.

Só tem um jeito de aprender: **programando. E errando.** Se tudo der certo vamos errar muito aqui nesse curso!!

Não copie e cole os códigos: escreva-os você mesmo.

Não se limite às tarefas/trabalhos propostos: procure fazer sempre mais.

O feito é melhor que o perfeito: faça primeiro um código básico que funcione, depois trabalhe nesse código para melhorá-lo.

Não deixe para fazer os trabalhos na última hora: faça um pouco por dia.



Quando estiver resolvendo um problema, **programar é a última etapa**. Primeiro você deve entender bem o problema, e fazer um esquema no papel. Só depois disso você começa a produzir o código.

7 Tarefa desta Semana

Instalar o Python no seu computador e deixar pronto pra rodar!!!! Vou deixar essa parte com vocês. Tem vários tutoriais sobre como instalar o Python. Uma boa ideia é instalar o Spyder e o Anaconda, que são programas para rodar códigos em Python. Dê uma olhada também no Google Colab: você pode rodar os seus códigos na nuvem, sem precisar instalar nada no seu computador.