

# Capítulo 1

## Documentação

Todo projeto de engenharia precisa ser bem documentado. Neste sentido, apresenta-se neste capítulo a documentação de uso do software para Cálculo da queda de pressão adimensional devido a migração de finos ocasionada pela injeção de água de baixa salinidade, considerando as equações da modelagem matemática fenomenológica de migração de finos honrando a formação de pontes e a demora na liberação de partículas. Esta documentação tem o formato de uma apostila que explica passo a passo como usar o software.

### 1.1 Documentação do usuário

A seguir encontra-se o manual do usuário, um guia que explica, passo a passo a forma de instalação e uso do software desenvolvido.

#### 1.1.1 Como instalar o software

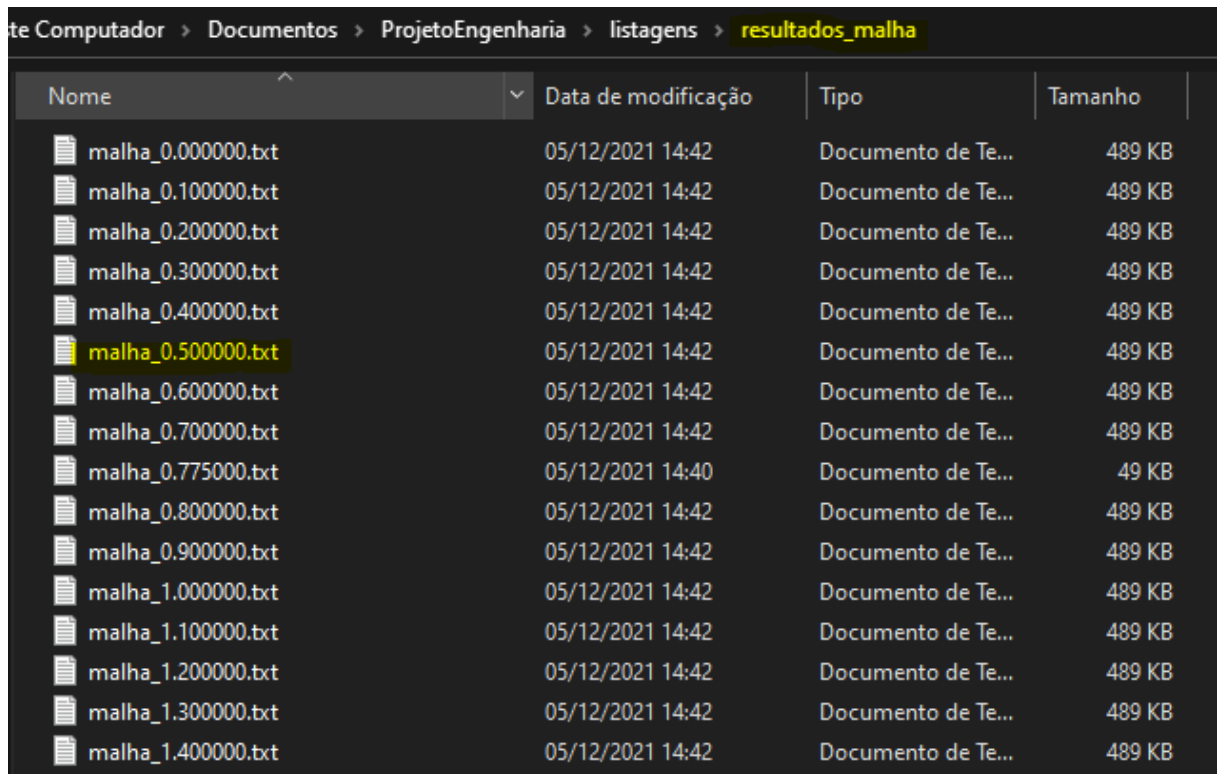
Para instalar o software execute o seguinte passo:

- Baixe e salve a pasta contendo os arquivos com os códigos do programa, cuja extensão dos arquivos são .hpp e .cpp. Também é necessário criar uma pasta com o nome “resultados\_malha”, onde será salvo os arquivos .txt referente as concentrações em função do tempo e do espaço.

#### 1.1.2 Como rodar o software

Para rodar o software compile o programa e, depois execute-o. Após a execução, será mostrado no console os valores dos parâmetros importados do arquivo ParticulaFluido.txt e do arquivo Rocha.txt. Também será mostrado no console os valores da malha do tempo e do tempo que inicia a formação de pontes(tb). Depois será obtido os resultados das concentrações em função do tempo e do espaço, que serão salvas em arquivo.txt na pasta “resultados\_malha”. Cada arquivo é referente a um tempo específico, dessa forma, os

arquivos são nomeados como “malha\_tempo[i]”. Por exemplo: no tempo=0,5s o arquivo é nomeado como “malha\_0.500000.txt”(Figura 8.1).



Nome	Data de modificação	Tipo	Tamanho
malha_0.000000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.100000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.200000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.300000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.400000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.500000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.600000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.700000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.775000.txt	05/12/2021 14:40	Documento de Te...	49 KB
malha_0.800000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_0.900000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_1.000000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_1.100000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_1.200000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_1.300000.txt	05/12/2021 14:42	Documento de Te...	489 KB
malha_1.400000.txt	05/12/2021 14:42	Documento de Te...	489 KB

Figura 1.1: Tela do programa mostrando xxx

Veja no Capítulo ?? - Teste, exemplos de uso do software.

## 1.2 Documentação para desenvolvedor

Apresenta-se nesta seção a documentação para o desenvolvedor, isto é, informações para usuários que queiram modificar, aperfeiçoar ou ampliar este software.

### 1.2.1 Dependências

Para compilar o software é necessário atender as seguintes dependências:

- Instalar o compilador g++ da GNU disponível em <http://gcc.gnu.org>. Para instalar no GNU/Linux use o comando `yum install gcc`.
- No sistema operacional do Windows, recomenda-se o Dev C++ disponível em <http://devc.softonic.com.br/>, visto que foi onde o código foi testado.
- No sistema operacional do MAC, recomenda-se o Xcode disponível em <https://developer.apple.com>.
- Instalar o software Gnuplot, disponível no endereço <http://www.gnuplot.info/>.

- É possível que haja necessidade de setar o caminho para execução do `gnuplot`.
- Biblioteca CGnuplot; os arquivos para acesso a biblioteca CGnuplot devem estar no diretório com os códigos do software
- O programa depende da entrada de dados por meio de arquivos de dados no formato `.txt` para preencher os parâmetros de entrada.

### 1.2.2 Como gerar a documentação usando doxygen

A documentação do código do software deve ser feita usando o padrão JAVADOC, conforme apresentada no Capítulo - Documentação, do livro texto da disciplina. Depois de documentar o código, use o software `doxygen` para gerar a documentação do desenvolvedor no formato html. O software `doxygen` lê os arquivos com os códigos (`*.hpp` e `*.cpp`) e gera uma documentação muito útil e de fácil navegação no formato html.

A listagem das classes no `doxygen` é apresentada na Figura 8.2, onde é possível acessar a documentação de cada classe. Na Figura 8.3 é apresentada a tela do `doxygen` do `main.cpp`, mostrando a documentação. E na Figura 8.4 é apresentado o mesmo, no entanto para a classe `CSimuladorParticulas.hpp`.

**listagens**

## Referência do diretório listagens

### Ficheiros

ficheiro	<b>CGnuplot.cpp</b>
ficheiro	<b>CGnuplot.hpp</b> [código]
ficheiro	<b>CGrid.cpp</b>
ficheiro	<b>CGrid.hpp</b> [código]
ficheiro	<b>CParticulaFluido.cpp</b>
ficheiro	<b>CParticulaFluido.hpp</b> [código]
ficheiro	<b>CRocha.cpp</b>
ficheiro	<b>CRocha.hpp</b> [código]
ficheiro	<b>CSimuladorParticulas.cpp</b>
ficheiro	<b>CSimuladorParticulas.hpp</b> [código]
ficheiro	<b>funcao.hpp</b> [código]
ficheiro	<b>main.cpp</b>
ficheiro	<b>metodointegracaonumerica.hpp</b> [código]
ficheiro	<b>metodosimpson.cpp</b>
ficheiro	<b>metodosimpson.hpp</b> [código]

Figura 1.2: Lista das classes no Doxygen

listagens

## Referência ao ficheiro main.cpp

```
#include <iostream>
#include <string>
#include "CSimuladorParticulas.cpp"
```

## Funções

```
int main ()
```

## Documentação das funções

### ◆ main()

```
int main ( )
```

```
9      {
10
12      CSimuladorParticulas simulacao("particulaFluido.txt", "rocha.txt");
13      simulacao.printCSimuladorParticulas();
14      simulacao.run();
16      /*delete x_par_ptr;
17      delete y_par_ptr;
18      delete objSeno2Dptr;
19      delete metodo;*/
20  }
```

Figura 1.3: Documentação do main.cpp

listagens

## CSimuladorParticulas.hpp

Ir para a documentação deste ficheiro.

```
1  #ifndef CSIMULADORPARTICULAS_HPP
2  #define CSIMULADORPARTICULAS_HPP
3
4  #include "CGrid.cpp"
5  #include "CRocha.cpp"
6  #include "funcao.hpp"
7  #include "CGnuplot.cpp"
8  #include "metodosimpson.cpp"
9  #include "CParticulaFluido.cpp"
10
11 #include<vector>
12 #include<string>
13 #include<iostream>
14
15 class CSimuladorParticulas : public CParticulaFluido, CRocha {
16 private:
17     size_t indiceTempoAtual = 0;
18     size_t size_tempo = 2001;
19     size_t size_malha = 10001;
20     size_t numPontosIntegral = 1001;
21     double start_x = 0.0, end_x = 0.10, start_t = 0, end_t = 200.0;
22
23     std::vector<double> tempo;
24     std::vector<double> malha;
25
26     std::vector<CGrid*> resultados_ao_longo_do_tempo;
27
28 public:
29     CSimuladorParticulas(std::string pathParticulaFluido, std::string pathRocha);
30
31     void run();
32
33 private:
34     double CalculoSigma_a(double x, double t);
35     double CalculoDiffSigma_a(double x, double t);
36     double CalculoLinhaZona(double x);
37     double CalculoTb(double x);
38 }
```

Figura 1.4: Documentação da classe CSimuladorParticulas.hpp.