

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE
LABORATÓRIO DE ENGENHARIA E EXPLORAÇÃO DE PETRÓLEO

PROJETO ENGENHARIA
DESENVOLVIMENTO DO SOFTWARE
ANÁLISE DA QUEDA DE PRESSÃO DEVIDO A MIGRAÇÃO DE
FINOS OCASIONADA
PELA INJEÇÃO DE ÁGUA DE BAIXA SALINIDADE
TRABALHO DA DISCIPLINA PROGRAMAÇÃO PRÁTICA

Versão 1:
Sayori Uto Borges
Prof. André Duarte Bueno

MACAÉ - RJ
Dezembro - 2021

Sumário

1	Introdução	1
1.1	Escopo do problema	1
1.2	Objetivos	2
2	Especificação	4
2.1	Nome do sistema/produto	4
2.2	Especificação	4
2.2.1	Requisitos funcionais	5
2.2.2	Requisitos não funcionais	6
2.3	Casos de uso	6
2.3.1	Diagrama de caso de uso geral	7
2.3.2	Diagrama de caso de uso específico	7
3	Elaboração	9
3.1	Análise de domínio	9
3.2	Formulação teórica	10
3.2.1	Premissas	11
3.2.2	Equações	11
3.2.3	Solução do sistema de equações	15
3.2.4	Método Numérico - Método de Simpsons	19
3.3	Identificação de pacotes – assuntos	21
3.4	Diagrama de pacotes – assuntos	21
4	AOO – Análise Orientada a Objeto	23
4.1	Diagramas de classes	23
4.1.1	Dicionário de classes	23
4.2	Diagrama de sequência – eventos e mensagens	26
4.2.1	Diagrama de sequência geral	26
4.2.2	Diagrama de sequência específico	26
4.3	Diagrama de comunicação – colaboração	29
4.4	Diagrama de máquina de estado	29
4.5	Diagrama de atividades	30

5	Projeto	31
5.1	Projeto do sistema	31
5.2	Projeto orientado a objeto – POO	33
5.3	Diagrama de componentes	35
5.4	Diagrama de implantação	36
6	Implementação	38
6.1	Código fonte	38
7	Teste	62
7.1	Teste 1: Entrada de dados, grid do tempo e cálculo do tb.	62
7.2	Teste 2: Gráficos	65
7.3	Teste 3: Integração numérica	65
8	Documentação	72
8.1	Documentação do usuário	72
8.1.1	Como instalar o software	72
8.1.2	Como rodar o software	72
8.2	Documentação para desenvolvedor	73
8.2.1	Dependências	73
8.2.2	Como gerar a documentação usando doxygen	74
	Referências Bibliográficas	78

Capítulo 1

Introdução

No presente trabalho desenvolve-se o software “Análise da queda de pressão devido a migração de finos ocasionada pela injeção de água de baixa salinidade”, um software aplicado a solução de um problema de Engenharia de Petróleo, por meio da linguagem orientada a objeto C++. A migração de finos provocada pela injeção de água de baixa salinidade causa declínio da permeabilidade, logo um declínio da produtividade do poço. Através do software será possível realizar a simulação desse processo, e assim realizar um estudo prévio para determinar qual o melhor método e consequentemente possibilitará uma elaboração mais específica e concreta do projeto a ser realizado.

1.1 Escopo do problema

- O transporte de emulsões e suspensões no meio poroso ocorre em várias situações e áreas da engenharia. Na área de engenharia de petróleo, esse processo é importante pois influencia na permeabilidade devido a captura de partículas pelo meio poroso. Esse processo é conhecido como dano à formação. A realização de um diagnóstico preciso e confiável sobre a causa do dano à formação é essencial para a tomada de decisão sobre qual método é o mais recomendado para remediação desse dano em poços. Em métodos de recuperação avançada, realizados para uma maior recuperação de óleo e gás, como a injeção de água de baixa salinidade/qualidade é observado esse fenômeno de captura e liberação de partículas. Sendo então interessante esse estudo para a área de engenharia de petróleo, visto que, para a produção de petróleo a mudança na permeabilidade influencia diretamente a injetividade e a produtividade do poço, ocasionando prejuízos as indústrias petrolíferas.
- Através da simulação da injeção de água de baixa salinidade, levando em consideração os dados de campo do reservatório será possível determinar qual a quantidade de finos, de partículas que ficam retidas no meio. O desenvolvimento do software é essencial, pois possibilita analisar e avaliar se seria recomendado a injeção de água de baixa salinidade, como método de recuperação avançada. Evitando prejuízos

e contribuindo para o desenvolvimento de um projeto mais específico, confiável e robusto.

- Os dados serão inseridos via arquivo .txt pelo usuário, uma vez que são dados obtidos da literatura ou por meio de experimentos laboratoriais com testemunhos. A equação do balanço de massa da cinética de captura, da cinética de liberação de partículas, do balanço dos íons e da lei de Darcy modificada formularam a modelagem matemática para a determinação da quantidade de partículas retidas. Para a resolução do problema foi selecionado o método numérico conhecido como “Método de Simpson” que será aplicado para resolução de integrais numéricas do problema para a obtenção do resultado da quantidade de partículas retidas no meio poroso por exclusão por tamanho, quando o tempo obedecer as condições estabelecidas. Caso contrário, o cálculo para obtenção da concentração de partículas retidas por exclusão por tamanho será efetuado sem necessidade de integração. A simulação tem o intuito de comparar os valores obtidos por meio do software e os casos reais, para avaliar a efetividade do mesmo. Tendo em vista, futuramente como intuito também desenvolver o cálculo da queda de pressão adimensional, ou seja, da impedância para avaliar o dano à formação do reservatório.

1.2 Objetivos

Os objetivos deste projeto de engenharia são:

- Objetivo geral:
 - Desenvolver um projeto de Engenharia de Software que calcule a quantidade de partículas depositadas no meio poroso devido a migração de finos, utilizando como dados/informações de referência os valores da literatura ou de experimentos laboratoriais, os quais serão inseridos pelo usuário por meio de arquivos .txt. Por meio do software será possível analisar o declínio de permeabilidade, ou seja o dano à formação, devido a quantidade de partículas retidas no meio poroso e assim evitar prejuízos econômicos as indústrias petrolíferas, pois será possível realizar uma análise e um estudo prévio do caso.
- Objetivos específicos:
 - Do ponto de vista da matemática foi desenvolvido um modelo matemático, onde é de suma importância para a teoria de fluxo dos meios porosos e para aplicações básicas na área de engenharia de reservatório. O software será baseado nas equações dessa modelagem matemática, que são a equação do balanço de massa, da cinética de liberação e captura de partículas, do balanço de íons, e da lei de Darcy modificada. Além disso, tem como intuito a solução de

uma integral numérica por meio de um método numérico que será aplicado por meio da programação orientada a objeto. Através do software poderá ser determinada as concentrações de partículas e conseqüentemente a quantidade de partículas depositadas.

- Do ponto de vista da engenharia através do software poderá ser explorado, a teoria de fluxo dos meios porosos e as práticas básicas da engenharia de reservatório. Além do mais, qual o melhor método a ser realizado para que ocorra uma melhoria na recuperação de óleo e gás.

- Etapas:

- Modelagem física e matemática do problema.
- Modelagem estática do software (diagramas de caso de uso, de pacotes, de classes).
- Modelagem dinâmica do software (desenvolver algoritmos e diagramas exemplificando os fluxos de processamento).
- Determinar as condições iniciais e de contorno do problema.
- Cálculo das concentrações que tem solução analítica e do tempo que inicia a formação de pontes(tb).
- Calcular a integral para obtenção da concentração retida por exclusão por tamanho ou formação de pontes, quando for necessário, de acordo com as condições estabelecidas.
- Calcular a queda de pressão em função do tempo.
- Salvar os resultados em arquivo.txt
- Gerar os gráficos com os resultados.
- Simular (realizar simulações para teste do software desenvolvido).

Capítulo 2

Especificação

Nesse capítulo será abordado as especificações do programa a ser modelado e desenvolvido para o projeto de engenharia de software. Será definido as características gerais do programa para que ele seja realizado. As necessidades que devem ser obedecidas e quais os serviços oferecidos e restrições do software.

2.1 Nome do sistema/produto

Nome	Simular o declínio de produtividade devido a migração de finos no reservatório, através do cálculo da quantidade de partículas depositadas.
Componentes principais	Sistema de equações, e de equações integrais numéricas para cálculo dos perfis de partículas depositadas dados os tempos dos perfis e as posições espaciais para amostragem.
Missão	Determinar a quantidade de partículas retidas por exclusão por tamanho, a fim de simular o declínio da produtividade dos poços, devido o declínio da permeabilidade causada pelas partículas retidas no meio poroso.

2.2 Especificação

Apresenta-se a seguir a especificação do software.

- O software que será desenvolvido, tem por objetivo simular a migração de nos ocasionada pela injeção de água de baixa salinidade a fim de determinar o declínio

de produtividade do poço. Esse programa irá determinar a quantidade de partículas depositadas/retidas, que são as responsáveis pela diminuição da permeabilidade. Essa análise é feita através do cálculo da Impendância/queda de pressão.

- A modelagem matemática do problema foi desenvolvida através das equações de balanço de massa de partículas, do balanço de massa de íons, da cinética de captura, da cinética de liberação e da lei de Darcy. A solução semi-analítica do problema será implementada no programa. As variáveis porosidade, velocidade, coeficiente de filtração por exclusão pelo tamanho, coeficiente de filtração por formação de pontes e coeficiente de dano à formação são conhecidas pelo usuário de acordo com os dados obtidos através de testes laboratoriais e através de dados da literatura, sendo de escolha do usuário qual valores de referência utilizar. A condição de contorno e a condição inicial também são conhecidas, e serão inseridas pelo usuário. Enquanto que a entrada dos valores das variáveis do fluido e da rocha/partícula para execução do software serão por meio da entrada de dados em um arquivo txt.
- O tempo será calculado por meio dos dados inseridos. Será implementado uma estrutura de seleção tendo como condição o tempo e n, para determinar qual método será executado para cálculo das concentrações. Determinado o método, os valores dos parâmetros será recebido e o software irá realizar o cálculo da concentração de partículas retidas por adesão e a concentração suspensa. Será implementado um método numérico para determinar a solução de integrais numéricas, e assim determinar a concentração de partículas retidas por exclusão por tamanho/formação de pontes. E assim, determinar os perfis de partículas depositadas de acordo com o tempo e a posição espacial.
- O presente código apresentará licença de software livre. O software apresentara interface em modo texto, visando simplificar a entrada e a saída de dados. O programa será realizado através da linguagem de programação orientada a objeto C++.

2.2.1 Requisitos funcionais

Apresenta-se a seguir os requisitos funcionais.

RF-01	O usuário deverá ter liberdade para selecionar com quais dados (experimentais ou da literatura), e qual material (tipo de rocha) deseja trabalhar.
RF-02	O usuário terá disponível os resultados obtidos em um arquivo de texto e gráfico.
RF-03	O usuário poderá plotar seus resultados em um gráfico. O gráfico poderá ser salvo como imagem ou ter seus dados exportados como texto.

RF-07	Nos casos em que o software for plotar gráficos o software externo <code>gnuplot</code> http://www.gnuplot.org deverá estar instalado no sistema.
--------------	--

2.2.2 Requisitos não funcionais

RNF-01	Os programa deverá ser multi-plataforma, podendo ser executado em <i>Windows</i> , <i>GNU/Linux</i> ou <i>Mac</i> .
---------------	---

2.3 Casos de uso

O caso de uso, descreve os cenários de uso do software, como ocorre a interação do usuário com as mesmas. Além disso, por meio do caso de uso é representado a sequência das tarefas que devem ser executadas, que são as etapas. E também, os cenários alternativos, que é representado pelas exceções, caso onde o usuário insere um dado errado, ou comete um erro.

Tabela 2.1: Caso de uso

Nome do caso de uso:	Cálculo da quantidade de partículas retidas no meio poroso.
Resumo/descrição:	Cálculo da quantidade de partículas, por meio da solução semi-analítica do problema e de parâmetros determinados experimentalmente ou de valores da literatura..
Etapas:	<ol style="list-style-type: none"> 1. Entrada/Leitura dos dados do fluido (velocidade e viscosidade). 2. Entrada/Leitura dados do reservatório (porosidade), do coeficiente de filtração por exclusão pelo tamanho, da formação de pontes e do dano à formação. 3. Entrada/Leitura constantes da equação constantes C e n. 4. Definir o tempo e criação de uma malha(grid) onde será armazenada os valores das concentrações de acordo com o tempo e espaço. 5. Analisar se o parâmetro tempo obedece a condição. 6. Determinar qual cálculo/equação realizar de acordo com os valores do tempo e do n. 7. Cálculo da concentração de partículas retidas e concentração de partículas suspensas. 8. Cálculo da integral numérica de concentração de partículas por exclusão pelo tamanho/formação de pontes, através do método numérico de Simpson. 9. Determinar os perfis de partículas depositadas de acordo com os tempos dos perfis e as posições espaciais. 10. Calcular a Impedância/queda de pressão para estimar a redução da permeabilidade. 11. Exportar resultados para disco 11. Gerar gráfico com os resultados obtidos.
Cenários alternativos:	Entrada errada de dados pelo usuário, como entrada de dados negativos quando são aceitos apenas valores positivos.

2.3.1 Diagrama de caso de uso geral

O diagrama de caso representa visualmente o caso de uso. Ele é utilizado para demonstrar as etapas do sistema pelo usuário, para que tenham uma visão geral do sistema. Pode ser utilizado antes e após a etapa de especificação. Assim, os diagramas mostram as interações do usuário com o programa.

2.3.2 Diagrama de caso de uso específico

No diagrama de caso de uso específico, o usuário com o sistema do programa. Mostrando a entrada de dados e a obtenção dos resultados obtidos que serão armazenados em

arquivos .txt na pasta “resultados malha”. E por fim, gerará gráficos com os resultados obtidos utilizando um sistema externo, o software gnuplot. Este diagrama de caso de uso ilustra as etapas a serem executadas pelo usuário ou sistema, a interação do usuário com o sistema.

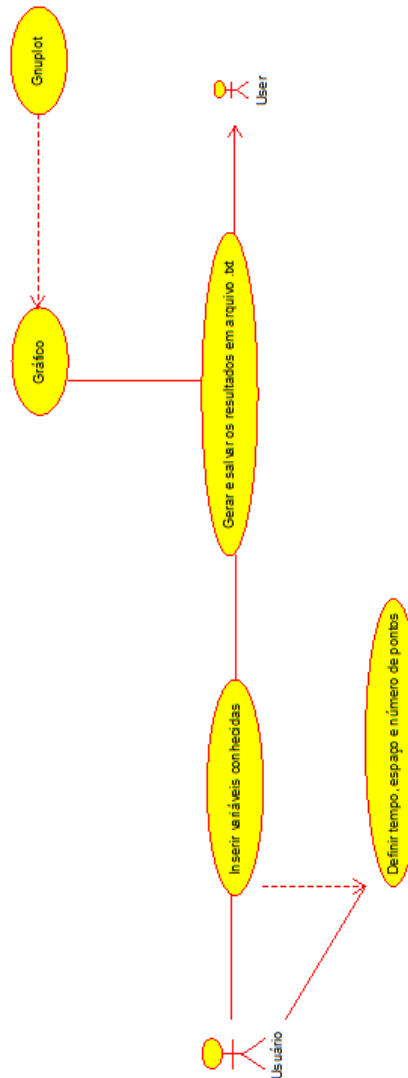


Figura 2.1: Diagrama de caso de uso específico – Título

Capítulo 3

Elaboração

Nesta etapa do projeto será realizado o estudo de conceitos do sistema que tenha relação com o sistema a ser desenvolvido, a análise de domínio e a identificação de pacotes, tendo como intuito uma melhor compreensão do software a ser desenvolvido. Além disso, será efetuada uma análise dos requisitos, para ajustar os requisitos iniciais e assim desenvolver um sistema útil, para que, futuramente, seja possível reutilizar e realizar a extensão do mesmo, como por exemplo incluir o cálculo da queda de pressão, conhecida como impedância, para determinar o declínio de injetividade e produtividade do poço.

Figura 3.1: Elaboração e análise de domínio (adaptado de [?])

3.1 Análise de domínio

O objetivo desta etapa é a compreensão do domínio, da área que abrange o software a ser desenvolvido, visto que o diagrama de pacotes representa as áreas abordadas. Além disso, visa determinar o que pode ser reaproveitado em outros sistemas.

- O desenvolvimento do software, aborda a migração de finos em meios porosos, um assunto presente em várias áreas e processos da engenharia, como a filtração industrial de líquidos e gases, a cromatografia de exclusão por tamanho, entre outros. No caso da engenharia de petróleo, especificamente, ocorre a captura e liberação de partículas pela rocha no meio poroso, ou seja, a migração de finos no reservatório causando o declínio na permeabilidade. A permeabilidade é uma propriedade da área petrolífera de suma importância, visto que, afeta diretamente a produtividade/injetividade do poço.

Disciplinas relacionadas ao desenvolvimento do sistema:

- Petrofísica: aborda as propriedades físicas de minerais e rochas, sendo necessária para entender os conceitos de permeabilidade, composição da matriz rochosa e do espaço poroso. As propriedades da rocha reservatório.

- Físico-química: para compreensão das reações que ocorrem do líquido com componentes presentes no meio poroso.
- Mecânica dos fluidos: estudo do comportamento dos fluidos. Essa área aborda as propriedades do fluido, que é necessária para entender o conceito de viscosidade e aborda algumas das forças que agem nas partículas como as forças de arraste e de elevação. Além da força gravitacional, no entanto o campo gravitacional é considerado agente externo na mecânica dos fluidos.
- Eletrostática: é a área da física que estuda as cargas elétricas em repouso, necessária para entender como atua as forças eletrostáticas sobre as partículas, que são causadas pelas interações entre as mesmas (colóides).
- Mecânica: área da física onde estuda os movimentos dos corpos, sendo necessária para entender o conceito de velocidade.
- Engenharia de reservatório: estudo da modelagem do comportamento dos fluidos de reservatório, e dos mecanismos de produção. Também estuda e propõe o método de recuperação secundária e/avançada de petróleo, estuda o comportamento do fluido no interior da rocha reservatório e visa estimar a reserva, entre outros. Fundamental para o entendimento e desenvolvimento da modelagem matemática e do sistema a ser desenvolvido.

3.2 Formulação teórica

Para avaliar a migração de finos é necessário analisar a captura de finos, descrita pela teoria fundamental de filtração profunda e a liberação de finos, descrita pelo modelo fenomenológico de filtração profunda. Esse processo é representado através da equação de balanço de massa, de cinética de captura e de liberação de partículas de acordo com Herzig et al. (1970). No presente estudo, as equações foram determinadas a partir do modelo modificado desenvolvido por Siqueira (2010), Bedrikovetsky et al. (2010) e Russel et al. (2018) que serão apresentadas nas subseções.(ARRUMAR REF)

O modelo fenomenológico de migração de finos devido a injeção de água de baixa salinidade honrando a formação de pontes e a demora na liberação de partículas é realizada através das equações do balanço de massa das partículas, do balanço de massa dos íons, da cinética de captura, da cinética de liberação de partículas e da lei de Darcy modificada, de acordo com as premissas mencionadas na seção 3.2.1. Por meio da modelagem foi obtida a equação para determinar a concentração de partículas depositadas. As propriedades do fluido e do meio poroso são obtidas por meio de experimentos laboratoriais utilizando amostras/testemunhos, através da literatura ou até mesmo por meio de dados, informações de casos de estudo e dados de campo.

3.2.1 Premissas

- Escoamento monofásico de salmoura em meio poroso de geometria linear (testes de laboratório, testemunhos).
- Fluidos e sólidos são considerados incompressíveis.
- Não ocorre dispersão/difusão significativa de íons e sólidos.
- Taxa de captura de partículas é influenciada pela concentração de partículas em suspensão, ocorre formação de pontes em alta concentração e captura de partículas por adesão em baixa concentração.
- Ocorre choque de salinidade instantâneo.
- Físico-química do processo (aumento do pH, difusão de íons nas camadas de argilominerais, etc.) encapsulada por cinética de liberação de partículas similar a dessorção iônica.
- Quantidade máxima de partículas presas por adesão depende do balanço de forças na parede do poro (salinidade/pH, velocidade do fluxo).

3.2.2 Equações

- Balanço de massa de partículas

A equação do balanço de massa descreve a variação de massa dentro do meio poroso devido ao fluxo do fluido, representada pela diferença da massa que entrou e a massa que saiu na direção do fluxo. Essa equação é muito utilizada na área de engenharia de reservatório para estudo do fluxo de líquido em meios porosos, para estudo do balanço de materiais que na realidade é o balanço de massa dos fluidos no interior dos poros da rocha-reservatório, entre outras áreas. Essa equação determina o acúmulo de partículas durante o fluxo de um fluido em um determinado período de tempo e espaço. No presente estudo o meio poroso é considerado unidimensional, com área de seção transversal variável $A(x)$. Além disso, é levado em consideração o fluxo linear em uma seção transversal Δx e área $A(x)$, durante um intervalo de tempo Δt . Assim, a equação do balanço de massa obtida para esse caso é apresentada como:

$$\frac{\partial}{\partial t}(\phi c + \sigma_a + \sigma_s) + \frac{\partial}{\partial x}(Uc) = 0, \quad (3.1)$$

onde ϕ é a porosidade do meio, t [unid] o tempo, x a posição, σ_a a concentração de partículas retidas nas paredes dos poros por adesão, σ_s a concentração de partículas retidas por exclusão por tamanho/pontes, c a concentração de partículas suspensas e U a velocidade de Darcy.

IMAGEM

- Balanço de massa de íons

A equação do balanço de massa de íons foi introduzida, pois a injeção de água de baixa salinidade acarreta a difusão de íons nas camadas de argilominerais. O inchamento de argilominerais, ocorre devido a injeção de uma solução salina em um meio onde não está em equilíbrio com ele.

Essa equação segue o mesmo conceito da equação do balanço de massa de partículas, no entanto é considerado a concentração molar χ ou massa por volume do íon. Assim a equação do balanço obtida:

$$\frac{\partial(\phi\chi)}{\partial t} + \frac{\partial(U\chi)}{\partial x} = 0, \quad (3.2)$$

onde χ [unidade] é a concentração molar ou massa por volume.

- Cinética de captura de partículas

A equação cinética de captura e liberação de partículas do modelo de Herzig et al. (1970) é empírica, e é representada como:

$$\frac{\partial\sigma_a}{\partial t} = \lambda'(\sigma)cU - k_{det}(U)\sigma_a, \quad (3.3)$$

onde a taxa de captura é proporcional ao fluxo advectivo de partículas suspensas (cU), e o termo de liberação é proporcional a concentração de partículas retidas. Nesse modelo c e σ_a representa a concentração suspensa e a concentração retida, respectivamente. A função de filtração é representada por $\lambda'(\sigma)$, quando essa função é constante é conhecida como coeficiente de filtração, isso ocorre quando a deposição é pequena. O coeficiente de liberação de partículas é representado por $k_{det}(U)$, e o módulo da velocidade de Darcy é representado por U . O coeficiente de liberação é normalmente determinado experimentalmente, sendo uma limitação do modelo. O valor do coeficiente de liberação é considerado como zero na equação (acima), para melhor exposição do modelo fenomenológico de filtração profunda.

O modelo clássico de filtração profunda apresenta algumas limitações, uma delas é o equilíbrio mecânico da partícula. A cinética de liberação desta equação (3.3) não reflete a condição do equilíbrio mecânico das partículas. A condição de equilíbrio é responsável pela permanência ou liberação da partícula, de acordo com as forças que atuam na partícula que são as forças eletrostáticas, força de arraste, força de elevação e força gravitacional. Vários estudos foram realizados e foi desenvolvido o modelo de liberação de partícula modificado, onde condiz com a condição do equilíbrio mecânico. No modelo proposto por Bedrikovetsky et al. (2011a), substitui-se a cinética de liberação pela função de retenção máxima (crítica). Essa função determina que ocorrerá captura de partículas até o momento em que a quantidade de partículas retida seja igual ou inferior a função de retenção máxima. A equação do modelo modificado é representada como:

$$\frac{\partial \sigma_s}{\partial t} = \alpha \lambda c, \quad (3.4)$$

$$\sigma_a = \sigma_{cr}(\gamma, U). \quad (3.5)$$

No presente estudo, a equação cinética de captura de partículas é baseada na equação do modelo clássico de filtração profunda MODIFICADA(CONFERIR), porém considerando a concentração mínima para a formação de pontes (c_b). Quando a concentração de partículas (c) é maior ou igual a concentração mínima para formação de pontes o coeficiente de filtração é a soma do coeficiente de filtração por exclusão pelo tamanho e por formação de pontes, pois considera-se que ocorre os dois fenômenos. No entanto, quando $c < c_b$ ocorre apenas captura de partícula por exclusão pelo tamanho. A cinética de captura é representada pela equação abaixo:

$$\frac{\partial \sigma_s}{\partial t} = \begin{cases} (\lambda_b + \lambda_s) U c & ; c \geq c_b, \\ \lambda_s U c & ; c < c_b, \end{cases} \quad (3.6)$$

onde o λ_s [unidade] representa o coeficiente de filtração por exclusão pelo tamanho, λ_b [unidade] representa o coeficiente de filtração por formação de pontes e c_b [v/v] a concentração mínima para formação de pontes. Uma análise dimensional da equação precedente será realizada para encontrar dimensões de lambda. Para efeito de análise dimensional, será considerado vol. Part./vol. Rocha como equivalente à vol. part./vol. fluido.

- Cinética de liberação de partículas

Neste modelo matemático é considerada a influência da salinidade da água injetada, ou seja a concentração molar de íons da solução, no caso a água de baixa salinidade. Devido ao fato de que a salinidade causa a liberação de partículas. Como é conhecido a quantidade máxima de partículas presas por adesão depende do balanço de forças na parede do poro, ou seja, depende da da velocidade de fluxo, da salinidade/pH, temperatura entre outros fatores. A equação da cinética de liberação é representada como:

$$\frac{\partial \sigma_a}{\partial t} = -C |\sigma_a - \sigma_{a,M}(\chi, U)|^n, \quad (3.7)$$

onde C e n são variáveis constantes do modelo, σ_a é a concentração de partículas retidas nas paredes dos poros por adesão, e $\sigma_{a,M}$ é a concentração de equilíbrio para as partículas aderidas na parede do poro, em função de χ e U .

- Lei de Darcy modificada

A lei de Darcy, descreve o fluxo de um fluido em um meio poroso. Esta lei relaciona a velocidade aparente do fluido com os gradientes de pressão, uma variável importante para avaliar o declínio da permeabilidade, que normalmente é expressa em termos de pressão. No

entanto, no presente estudo será utilizada a lei de Darcy modificada (BARENBLATT et al., 1990). A equação do problema unidimensional na direção x é expressa como:

$$U = -\frac{k_0 k(\sigma)}{\mu} \frac{\partial p}{\partial x}, \quad (3.8)$$

onde k_0 é a permeabilidade inicial, ΔP o gradiente de pressão, μ é a viscosidade do fluido e $k(\sigma)$ representa a redução percentual da permeabilidade em função da deposição.

A função $k(\sigma)$ é adotada como:

$$k(\sigma) = \frac{1}{1 + \beta\sigma}, \quad (3.9)$$

onde β é conhecido como o coeficiente de dano à formação, que é considerado constante. Esse coeficiente, é uma constante empírica que depende das características das partículas retidas e do meio poroso.

Através da lei de Darcy modificada é possível calcular a queda de pressão dimensional:

$$-\Delta p(t) = \frac{\mu U L}{k_0} + \frac{\mu U L}{k_0} \beta \int_0^L \sigma_s(x, t) dx. \quad (3.10)$$

- Condições iniciais do problema

$$c(x, 0) = \sigma_s(x, 0) = 0; \quad (3.11)$$

$$\sigma_a(x, 0) = \sigma_{a0}; \quad (3.12)$$

onde σ_{a0} representa a concentração de partículas retidas nas paredes dos poros por adesão inicial. O cálculo da concentração é obtida através da equação (), tomando como exemplo a caulinita como principal mineral presente:

$\%_a$ Caulinita

$$V_a = V(1 - \phi)\%_a. \quad (3.13)$$

$$\sigma = \frac{V_{\text{sólido}}}{V} \Rightarrow \sigma_{a0} = \frac{V(1 - \phi)\%_a}{V} = (1 - \phi)\%_a. \quad (3.14)$$

$$\chi(x, 0) = \chi_H; \quad (3.15)$$

- Condições de contorno

$$c(0, t) = 0; \quad (3.16)$$

$$\chi(0, t) = \chi_L. \quad (3.17)$$

nas condições de contorno, quando $x = 0$, a concentração em suspensão é nula, visto que, à montante do testemunho a injeção de água é ausente de partículas.

3.2.3 Solução do sistema de equações

Nesta seção será apresentada a solução do sistema de equações da subseção 3.2.3. O sistema de equações admite solução semi-analítica e parte do sistema solução numérica. Sabendo que a velocidade de Darcy U é determinada pela vazão de injeção do teste, que é conhecida e constante é possível observar que a equação do balanço de massa dos íons está desacoplada do resto do sistema. Por meio do método das características a equação do balanço de massa de íons é escrita como:

$$\frac{d\chi}{d\eta} = \frac{\partial\chi}{\partial t} \frac{dt}{d\eta} + \frac{\partial\chi}{\partial x} \frac{dx}{d\eta} = \frac{\partial\chi}{\partial t} \phi + \frac{\partial\chi}{\partial x} U = 0. \quad (3.18)$$

Assim:

$$\phi = \frac{dt}{d\eta} \Rightarrow t - t_0 = \phi\eta; \quad (3.19)$$

$$U = \frac{dx}{d\eta} \Rightarrow x - x_0 = U\eta. \quad (3.20)$$

As condições possibilitam o cálculo da salinidade da solução em qualquer seção transversal e a qualquer momento, como:

$$\chi(x, t) = \begin{cases} \chi_H & , \quad x > \frac{Ut}{\phi}, \\ \chi_L & , \quad ; \quad x < \frac{Ut}{\phi}. \end{cases} \quad (3.21)$$

Determinada a salinidade, conforme equação (21), é possível resolver a equação da cinética de liberação de partícula(7), que pode ser reescrita como:

$$\frac{\partial\sigma_a}{\partial t} = -C(\sigma_a(x, t) - \sigma_{a,M}(\chi(x, t), U))^n. \quad (3.22)$$

Para obter a concentração de partículas retidas nas paredes dos poros por adesão (σ_a), é realizada a integração da equação precedente. Primeiro, é percebido que as curvas características desta equação são todas verticais. Segundo, essas curvas características são divididas em dois segmentos: em um dos segmentos o $t \in [0, \frac{\phi x}{U})$ e no outro semi-infinito $t \in (\frac{\phi x}{U}, \infty)$.

Resolvendo a integração e considerando os limites de integração, é obtida a equação para σ_a para $n=1$ e quando $n \neq 1$, conforme abaixo:

para $n = 1$

$$\sigma_a(x, t) = \begin{cases} \sigma_{a0}, & t < \frac{x\phi}{U}, \\ \sigma_{a,M} + (\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}, & t > \frac{x\phi}{U}. \end{cases} \quad (3.23)$$

$$\frac{\partial \sigma_a}{\partial t} = \begin{cases} 0, & t < \frac{x\phi}{U} \\ -C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}, & t > \frac{x\phi}{U}. \end{cases} \quad (3.24)$$

para $n \neq 1$

$$\sigma_a(x, t) = \begin{cases} \sigma_{a0}, & t < \frac{x\phi}{U}, \\ [\sigma_{a,M} + (\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - \frac{x\phi}{U})]^{\frac{n}{1-n}}, & t > \frac{x\phi}{U}. \end{cases} \quad (3.25)$$

$$\frac{\partial \sigma_a}{\partial t} = \begin{cases} 0, & t < \frac{x\phi}{U}, \\ -C [(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - \frac{x\phi}{U})]^{\frac{n}{1-n}}, & t > \frac{x\phi}{U}. \end{cases} \quad (3.26)$$

Através da equação do balanço de massa (1), da equação de cinética de captura (6) e da equação (24) para $n=1$ e da equação (26) para $n \neq 1$, é obtida a equação de concentração de partículas suspensas (c) para $n=1$ e para $n \neq 1$. Foi levado em consideração as condições para $t < \frac{\phi x}{U}$ e para $t \geq \frac{\phi x}{U}$, e em cada condição do tempo é levado em consideração a condição de $c < cb$ e $c \geq cb$, onde cb é a concentração mínima para formação de pontes. As equações e condições estão representadas abaixo.

Obtendo assim:

para $n = 1$, temos dois casos onde

$$1. \ t < \frac{x\phi}{U}$$

$$\bullet \ c < cb$$

$$c(x, t) = 0. \quad (3.27)$$

$$\bullet \ c \geq cb$$

$$c(x, t) = 0 \Rightarrow \sigma_s(x, t) = 0. \quad (3.28)$$

$$2. \ t \geq \frac{x\phi}{U}$$

$$\bullet \ c < cb$$

$$c(x, t) = \frac{C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}}{U\lambda_s}(1 - e^{-\lambda_s x}) \quad (3.29)$$

$$\bullet \ c \geq cb$$

$$c(x, t) = \frac{C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}}{U(\lambda_b + \lambda_s)} \left[1 - \frac{1 - \frac{U(\lambda_b + \lambda_s)cb}{C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}}}{\left(1 - \frac{U\lambda_s cb}{C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}}\right)^{1 + \frac{\lambda_b}{\lambda_s}}} e^{(\lambda_b + \lambda_s)x} \right]. \quad (3.30)$$

para $n \neq 1$, temos dois casos onde

$$1. \ t < \frac{x\phi}{U}$$

- $c < c_b$

$$c(x, t) = 0. \quad (3.31)$$

- $c \geq cb$

$$c(x, t) = 0 \Rightarrow \sigma_s(x, t) = 0. \quad (3.32)$$

$$2. \ t \geq \frac{x\phi}{U}$$

- $c < c_b$

$$c(x, t) = \frac{C [(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}} (1 - e^{-x\lambda_s})}{\lambda_s U}, \quad (3.33)$$

onde $t_0 = t - \frac{\phi x}{U}$ e $\eta = \frac{x}{U}$.

- $c \geq cb$

$$c(x, t) = \frac{C [(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}}}{U(\lambda_s + \lambda_b)}$$

$$\left[1 - \frac{[1 - \frac{U(\lambda_s + \lambda_b)c_b}{C[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}}}]e^{-x(\lambda_b + \lambda_s)}}{\left(1 - \frac{\lambda_s U c_b}{C[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}}}\right)^{1 + \frac{\lambda_b}{\lambda_s}}} \right], \quad (3.34)$$

onde $t_0 = t - \frac{\phi x}{U}$ e $\eta = \frac{x}{U}$.

Encontrada as equações da concentração de partículas suspensas(c) é possível determinar o momento em que inicia a formação de pontes(tb). A zona de formação de pontes é quando a concentração de partículas suspensas é igual a concentração mínima para formação de pontes. Sendo assim, ao igualar a equação de $c(x, t)$ a cb em cada caso é encontrado o tb . Obtendo :

para $n = 1$

$$tb = \frac{x\phi}{U} + \frac{1}{C} \ln\left(\frac{C(\sigma_{a0} - \sigma_{a,M})(1 - e^{-\lambda_s x})}{c_b U \lambda_s}\right), \quad (3.35)$$

para $n \neq 1$

$$t_b = 2\frac{\phi x}{U} + \frac{(\sigma_{a0} - \sigma_{a,M})^{1-n}}{C(1-n)} - \frac{1}{C(1-n)} \left(\frac{\lambda_s U c_b}{C(1 - e^{-x\lambda_s})} \right)^{\frac{1-n}{n}}. \quad (3.36)$$

Para encontrar a concentração de partículas retidas por exclusão por tamanho/pontes, utiliza-se a equação de cinética de captura e $c(x, t)$. É levado em consideração tres casos, onde $t < \frac{\phi x}{U}$, $t \geq \frac{\phi x}{U}$ e $t < tb$, e por fim o caso onde $t < \frac{\phi x}{U}$ ou $t > tb$ para $n = 1$ e para $n \neq 1$. As equações obtidas foram:

para $n = 1$, temos três casos:

1. $t < \frac{x\phi}{U}$

$$\sigma_s(x_0, t) = 0 \quad (3.37)$$

2. $t \geq \frac{x\phi}{U}$ e $t < t_b$

$$\frac{\partial \sigma_s}{\partial t} = (\lambda_s + \lambda_b)U \frac{C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}}{U(\lambda_s + \lambda_b)} \left[1 - \frac{1 - \frac{U(\lambda_s + \lambda_b)}{C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}}}{\left(1 - \frac{U\lambda_s c_b}{C(\sigma_{a0} - \sigma_{a,M})e^{-C(t - \frac{x\phi}{U})}}\right)} \right], \quad (3.38)$$

$$\sigma_s(x_0, t) = C(\sigma_{a0} - \sigma_{a,M})$$

$$* \left[\frac{1 - e^{-C(t - \frac{\phi x_0}{U})}}{C} + e^{-(\lambda_s + \lambda_b - C(\frac{x\phi}{U})x_0)} \int_{\frac{\phi x}{U}}^t \frac{\frac{U(\lambda_s + \lambda_b)c_b}{C(\sigma_{a0} - \sigma_{a,M})e^{-C(\frac{\phi x_0}{U})}} - e^{-Ct}}{\left(1 - \frac{U\lambda_s c_b}{C(\sigma_{a0} - \sigma_{a,M})e^{-C(\frac{\phi x_0}{U})}}\right)} dt \right], \quad (3.39)$$

cuja integral só pode ser expressa em termos de funções elementares.

3. $t \geq \frac{x\phi}{U}$, $t > t_b$ ou $t < \frac{x\phi}{U}$

$$\frac{\partial \sigma_s}{\partial t}(x_0, t) = C(\sigma_{a0} - \sigma_{a,M})(1 - e^{\lambda_s x_0})e^{-C(t - \frac{x\phi}{U})}, \quad (3.40)$$

$$\sigma_s(x_0, t) = \sigma_s \left(x_0, \max(t_b(x_0), \frac{\phi x_0}{U}) \right) +$$

$$+ (\sigma_{a0} - \sigma_{a,M})(1 - e^{-\lambda_s x_0})e^{C\frac{\phi x_0}{U}} \left[e^{-C\max(tb, \frac{\phi x}{U})} - e^{-ct} \right]. \quad (3.41)$$

Para $n \neq 1$, temos três casos, onde

1. $t < \frac{x\phi}{U}$

$$\sigma_s(x_0, t) = 0 \quad (3.42)$$

2. $t \geq \frac{x\phi}{U}$ e $t < t_b$

$$\frac{\partial \sigma_s(x_0, t)}{\partial t} = C \left[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U}) \right]^{\frac{n}{1-n}}$$

$$* \left[1 - \frac{\left[1 - \frac{U(\lambda_s + \lambda_b)c_b}{C[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}}} \right] e^{x(\lambda_b + \lambda_s)}}{\left(1 - \frac{\lambda_s U c_b}{C[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}}} \right)^{1 + \frac{\lambda_b}{\lambda_s}}} \right], \quad (3.43)$$

$$\sigma_s(x_0, t) - \sigma_s(x_0, \frac{\phi x}{U}) = C \int_{\frac{\phi x}{U}}^t \left[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U}) \right]^{\frac{n}{1-n}}$$

$$* \left[1 - \frac{\left[1 - \frac{U(\lambda_s + \lambda_b)c_b}{C[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}}} \right] e^{x(\lambda_b + \lambda_s)}}{\left(1 - \frac{\lambda_s U c_b}{C[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{n}{1-n}}} \right)^{1 + \frac{\lambda_b}{\lambda_s}}} \right] dt, \quad (3.44)$$

cuja integral só pode ser expressa em termos de funções elementares.

3. $t \geq \frac{x\phi}{U}$, $t > t_b$ ou $t < \frac{x\phi}{U}$

$$\frac{\partial \sigma_s(x_0, t)}{\partial t} = C \left[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U}) \right]^{\frac{n}{1-n}} (1 - e^{-x\lambda_s}), \quad (3.45)$$

$$\sigma_s(x_0, t) - \sigma_s(x_0, \max(t_b(x_0), \frac{\phi x_0}{U})) = (1 - e^{-x\lambda_s}) \frac{(n-1)}{(1-n)}$$

$$\left[[(\sigma_{a0} - \sigma_{a,M})^{1-n} - C(1-n)(t - 2\frac{\phi x}{U})]^{\frac{1}{1-n}} \right]. \quad (3.46)$$

3.2.4 Método Numérico - Método de Simpsons

O Cálculo Numérico engloba um conjunto de métodos que são utilizados para se obter a solução de problemas matemáticos de forma aproximada. Esses métodos se aplicam para problemas em que não tem uma solução exata, que não tem uma solução analítica,

sendo necessário resolver por um método numérico. Um exemplo é a integral da equação (44) que não possui solução analítica.

O método numérico selecionado foi o Método de Simpsons, que consiste na aproximação das funções por arcos de parábola, com o objetivo de encontrar o valor aproximado da integral da função. Esse método, diferente dos métodos anteriores como o do trapézio, precisa de três pontos do intervalo e não apenas dos pontos dos extremos.

Primeiro, considera - se os pontos igualmente espaçados entre si, e a origem o ponto médio. Assim, os pontos são $x_0 = -h, x_1 = 0, x_2 = h$ e as respectivas imagens são y_0, y_1, y_2 . O polinômio interpolador é escrito como $p(x) = ax^2 + bx + c$, visto que é aproximado a arcos de parábola, ou seja, uma função de segundo grau.

Segundo, integrando a função obtemos:

$$\begin{aligned}\int_{x_0}^{x_2} p(x)dx &= \int_{-h}^h (ax^2 + bx + c)dx = \\ &= \left[\frac{ax^3}{3} + \frac{bx^2}{2} + cx \right]_{x=-h}^{x=h} = \\ &= \frac{2ah^3}{3} + 2ch^2 = \frac{h}{3}(2ah^2 + 6c).\end{aligned}$$

Entretanto, considerando que a parábola passe pelos pontos $(-h, y_0)$, $(0, y_1)$ e (h, y_2) , obtém-se:

$$y_0 = a(-h)^2 + b(-h) + c = ah^2 - bh + c;$$

$$y_1 = a * 0^2 + b * 0 + c = c$$

$$y_2 = ah^2 + bh + c.$$

Dessa forma, é obtida a equação $y_0 + 4y_1 + y_2 = 2ah^2 + 6c$, assim é possível escrever a integral de $p(x)$ como:

$$\int_{x_0}^{x_2} p(x)dx = \frac{h}{3}(2ah^2 + 6c) = \frac{h}{3}(y_0 + 4y_1 + y_2).$$

A proposição da regra de simpsons enuncia que:

Se $f(x)$ é uma função contínua, e os pontos (x_0, y_0) , (x_1, y_1) e (x_2, y_2) do gráfico de $f(x)$ estão igualmente espaçados horizontalmente, ou seja, se $x_2 - x_1 = x_1 - x_0 = h$, então podemos aproximar:

$$\int_{x_0}^{x_2} f(x)dx \cong \frac{h}{3}(y_0 + 4y + y_2) \quad (3.47)$$

3.3 Identificação de pacotes – assuntos

Pacote é um conjunto de classes que se relacionam por meio de conceitos, assuntos em comum, ou seja, que fazem parte de uma mesma área.

- Pacote Dados: Pacote de dados onde são obtidos os parâmetros do programa via arquivo txt, e que foram obtidos através da literatura, ou através de experimentos laboratoriais.
- Pacote MétodoNumerico: Pacote onde apresenta o método numérico para cálculo das integrais que não tem solução analítica, apenas numérica.
- Pacote Simulação: Composto de diversas equações para realização do cálculo da concentração retida nas paredes dos poros por adesão, da concentração retida por exclusão por tamanho/pontes e do cálculo da zona de formação de pontes para determinar, do tempo que inicia a formação de pontes (t_b). Nesse pacote é criada a malha do espaço e do tempo, é salvo os resultados obtidos dos cálculos.
- Pacote Resultados: Pacote onde contém os métodos para gerar os gráficos e para criar o grid dos resultados e salvar nos arquivos.

3.4 Diagrama de pacotes – assuntos

Através do diagrama de pacotes é possível observar as dependências entre as diferentes partes do sistemas. Pode ser composto por sistemas, subsistemas, hierarquias de classes, classes, interfaces, componentes, nós, colaborações e casos de uso.

No presente projeto, o diagrama de pacotes é composto por três pacotes, o pacote de dados, de método numérico e de simulação.

coloque aqui texto falando do diagrama de pacotes, referencie a figura. Veja Figura 3.2.

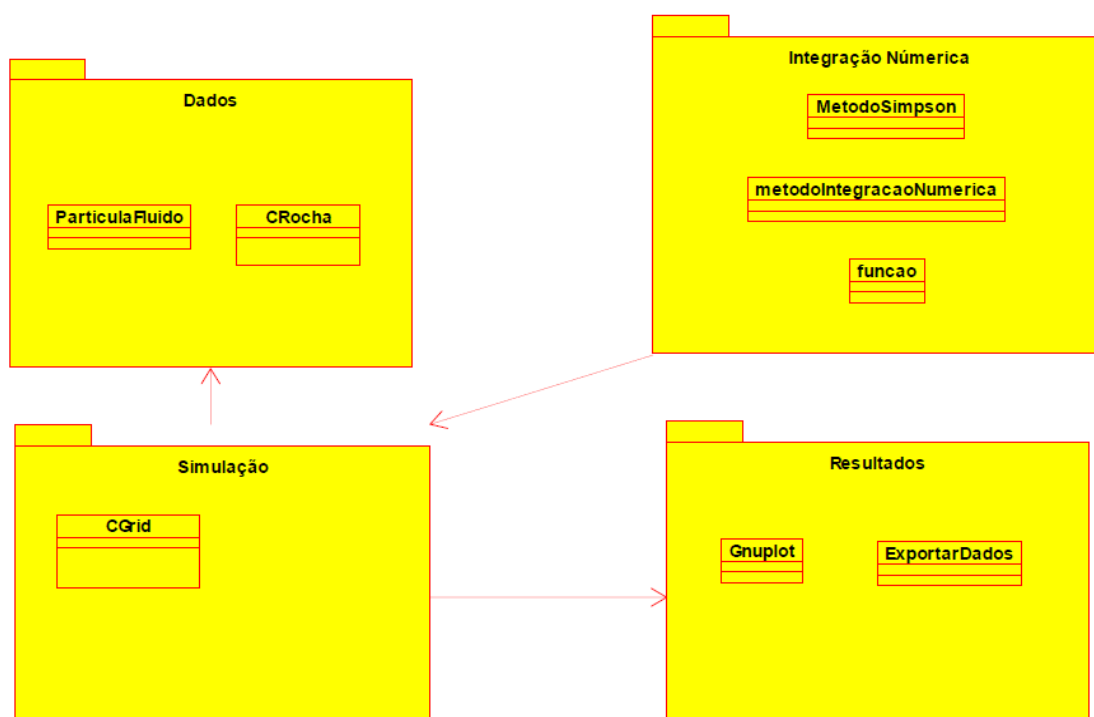


Figura 3.2: Diagrama de Pacotes

Capítulo 4

AOO – Análise Orientada a Objeto

Nesse capítulo será apresentado o desenvolvimento do projeto de engenharia sobre migração de finos devido a injeção de água de baixa salinidade, no nosso caso um software aplicado a engenharia de petróleo, utilizando a linguagem a AOO – Análise Orientada a Objeto. A AOO utiliza algumas regras para identificar os objetos de interesse, as relações entre os pacotes, as classes, os atributos, os métodos, as heranças, as associações, as agregações, as composições e as dependências. A análise consiste em apresentar os modelos estruturais dos objetos e gerar um conjunto de diagramas.

O modelo de análise deve ser conciso, simplificado e deve mostrar o que deve ser feito, não se preocupando como isso será realizado.

4.1 Diagramas de classes

O diagrama de classes é apresentado na Figura 4.1 e na figura 4.2.

4.1.1 Dicionário de classes

- Classe CParticulaFluido: representa a leitura das variáveis referente as propriedades das partículas e do fluido via arquivo .txt.
- Classe CRocha: representa a leitura das variáveis referente as propriedades da rocha via arquivo .txt.
- Classe CGrid: classe que possui o método para criação de uma malha/grid do tempo e do espaço e salva os resultados das concentrações obtidas em função do espaço e do tempo em arquivo .txt.
- Classe CSimuladorParticulas: representa o cálculo da concentração de partículas em suspensão, da concentração de partículas retidas por adesão, da concentração de partículas retidas por formação de pontes/exclusão, do tempo em que inicia a

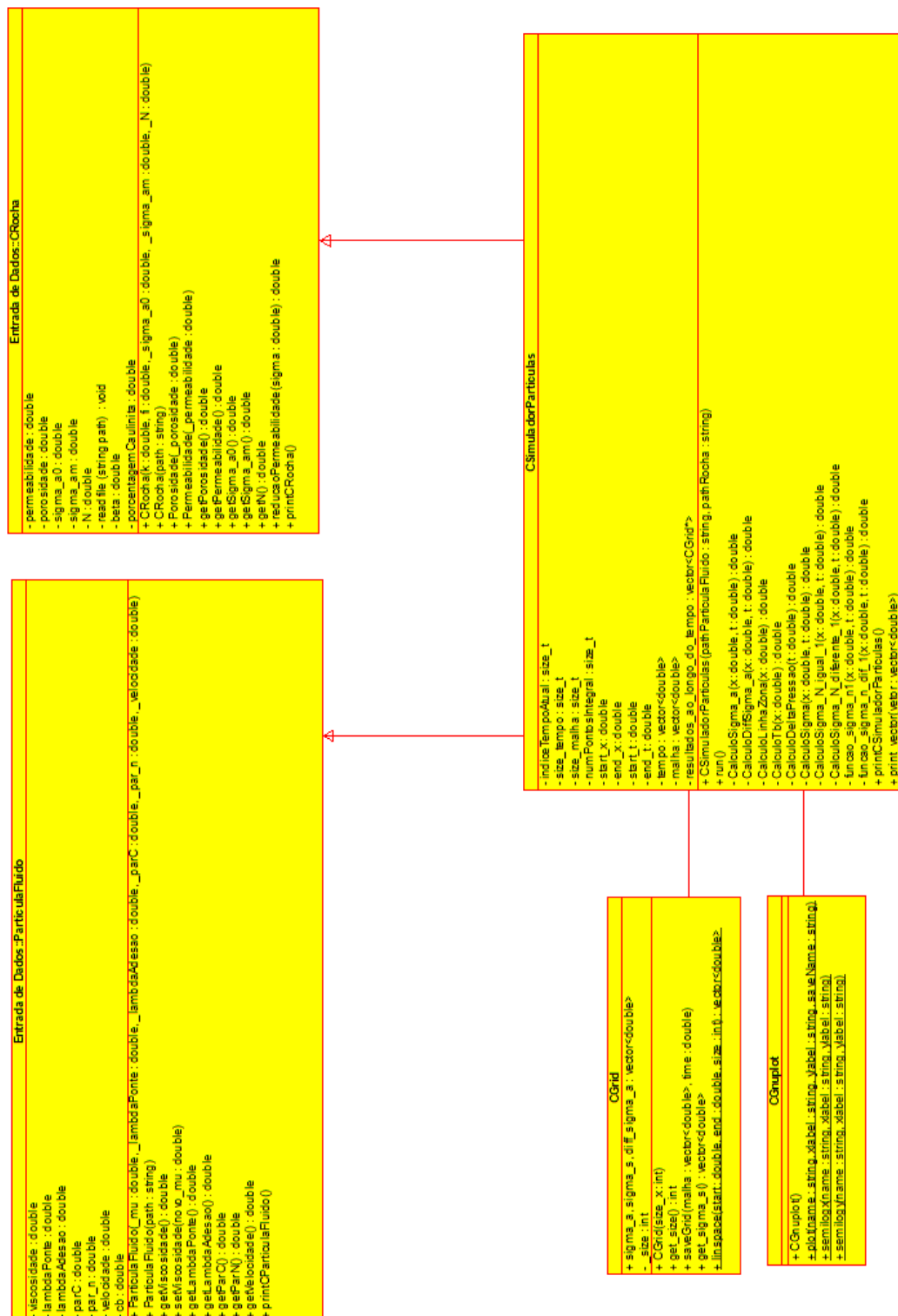


Figura 4.1: Diagrama de classes

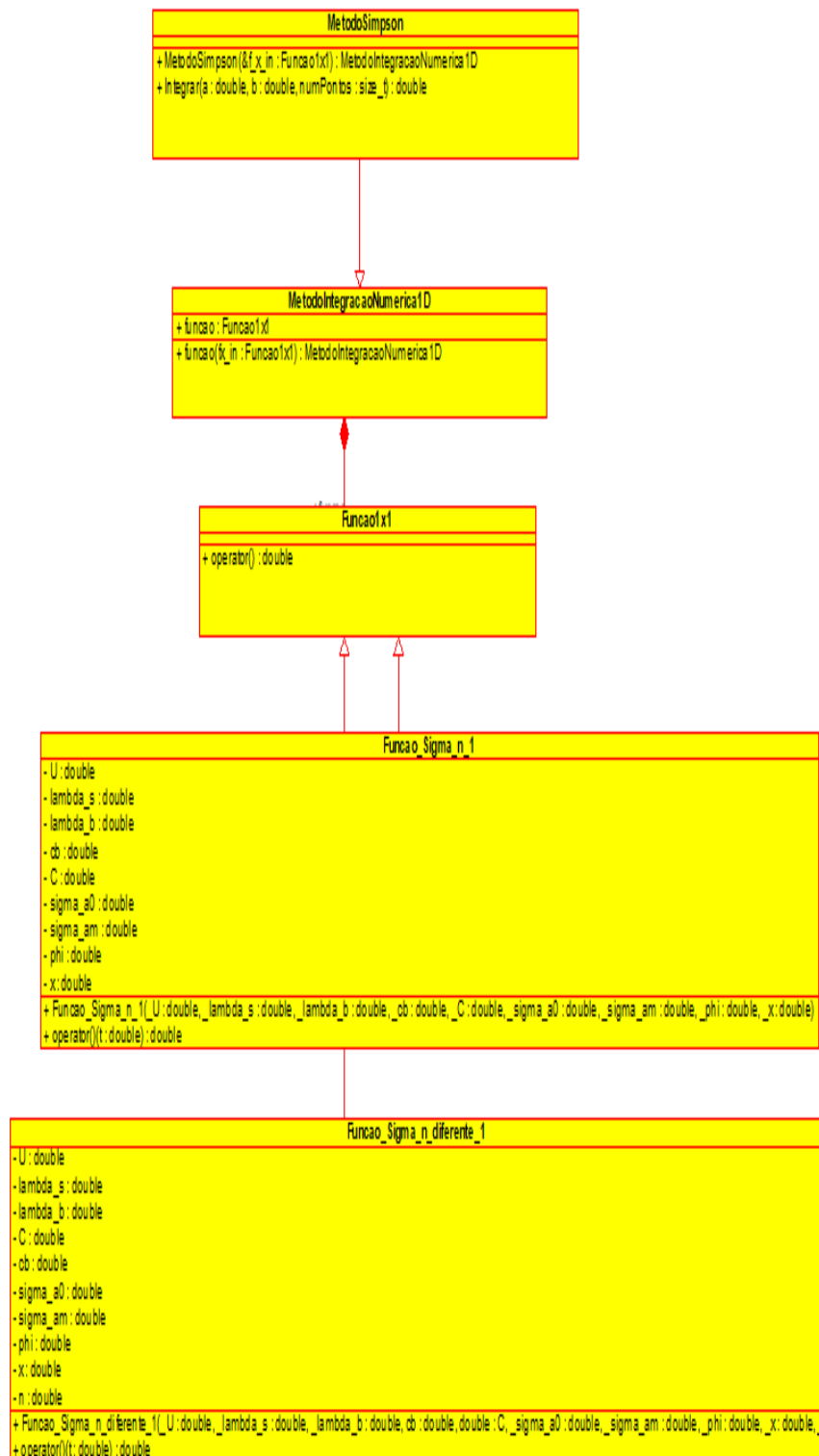


Figura 4.2: Diagrama de classes

formação de pontes e da impedância/queda de pressão. Salva os resultados nos arquivos .txt e .dat, além disso mostra os gráficos.

- Classe MetodoSimpson: herdeira da classe MetodoIntegracaoNumerica1D, apresenta o método numérico que será utilizado para resolver a integral da equação do σ_s , apenas quando for necessário de acordo com a condição do tempo e do n(número de pontos).
- Classe MetodoIntegracaoNumerica1D: representa o método de integração.
- Classe Funcao1x1:método virtual, passa a função que será integrada na classe MetodoIntegracaoNumerica1D.
- Classe Funcao_Sigma_n_1: representa as fórmulas que serão integradas pelo método de Simson para $n=1$;
- Classe Funcao_Sigma_n_diferente_1: representa as fórmulas que serão integradas pelo método de Simson para $n \neq 1$.
- Classe CGnuplot: plota o gráfico da queda de pressão em função do tempo e da concentração de partículas retidas por exclusão por tamanho(σ_s) pelo tempo.

4.2 Diagrama de sequência – eventos e mensagens

O diagrama de sequência enfatiza a troca de eventos e mensagens e sua ordem temporal. Contém informações sobre o fluxo de controle do software. Costuma ser montado a partir de um diagrama de caso de uso e estabelece o relacionamento dos atores (usuários e sistemas externos) com alguns objetos do sistema.

4.2.1 Diagrama de sequência geral

O diagrama da figura 4.3 detalha a sequência do fluxo de forma geral, mostrando apenas os principais eventos que ocorrem para realizar os devidos cálculos e posteriormente salvar os resultados obtidos em arquivos .txt e plotar os gráficos.

4.2.2 Diagrama de sequência específico

O diagrama da figura 4.4 detalha como ocorre a sequência do fluxo de controle do software de forma específica a fim de calcular a queda de pressão em função do tempo e salvar os resultados das concentrações obtidas em função do tempo e do espaço, além de plotar os gráficos. Veja o diagrama de sequência específico na Figura 4.4.

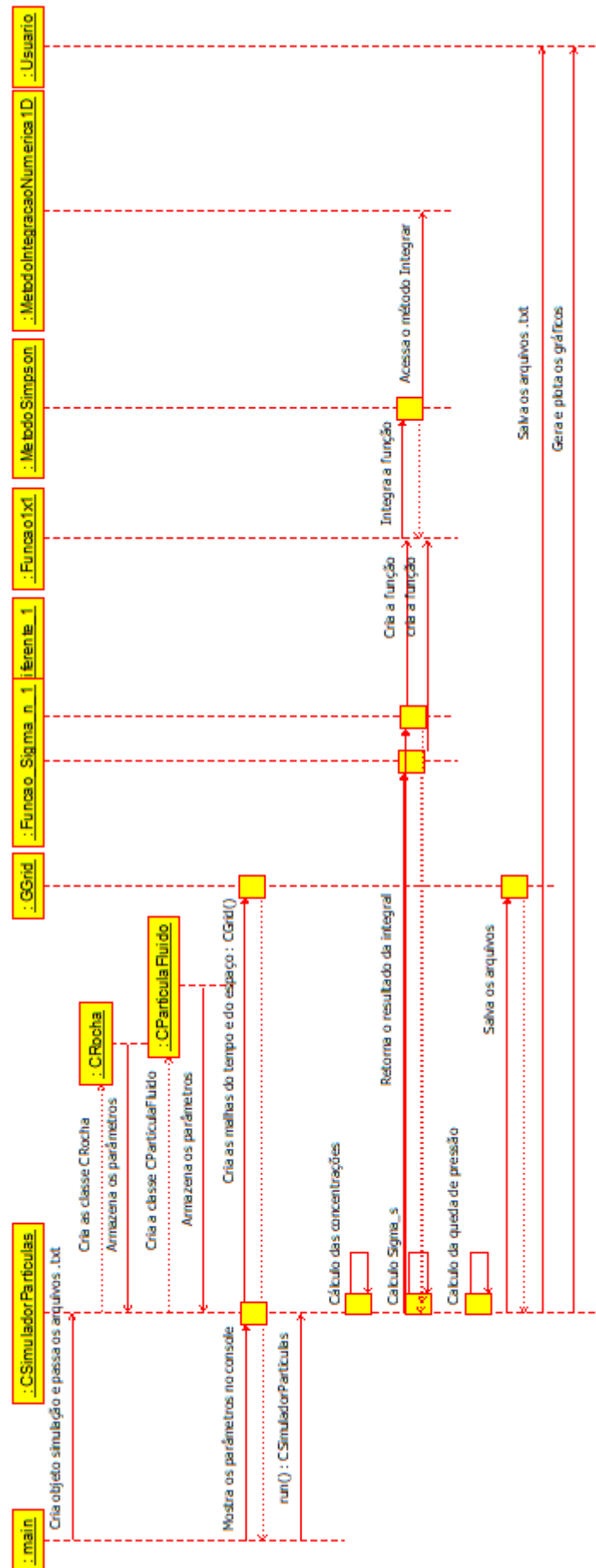
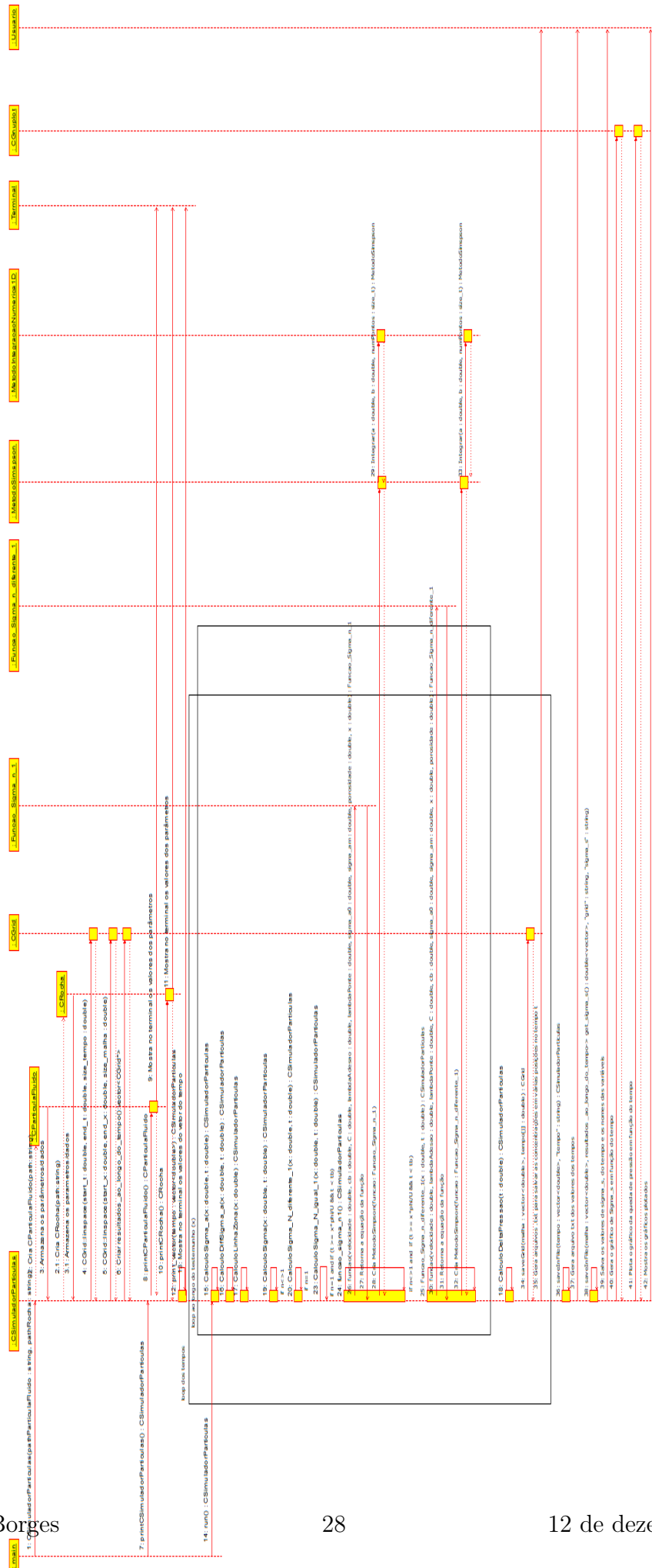


Figura 4.3: Diagrama de sequência geral.



4.3 Diagrama de comunicação – colaboração

No diagrama de comunicação o foco é a interação e a troca de mensagens e dados entre os objetos.

Veja na Figura 4.5 o diagrama de comunicação mostrando a sequência de passos executadas pelo software desde a entrada de dados feita pelo usuário até a obtenção dos gráficos.

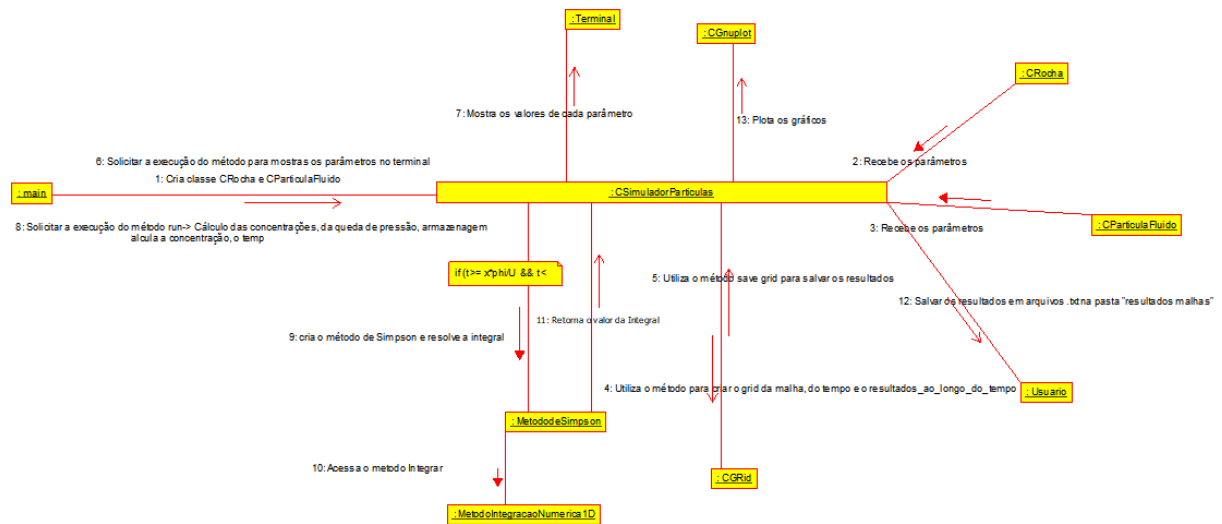


Figura 4.5: Diagrama de comunicação

4.4 Diagrama de máquina de estado

Um diagrama de máquina de estado representa os diversos estados que o objeto assume e os eventos que ocorrem ao longo de sua vida ou mesmo ao longo de um processo (histórico do objeto). É usado para modelar aspectos dinâmicos do objeto.

Veja na Figura 4.6 o diagrama de máquina de estado para os objetos da classe CSimuladoParticulas.

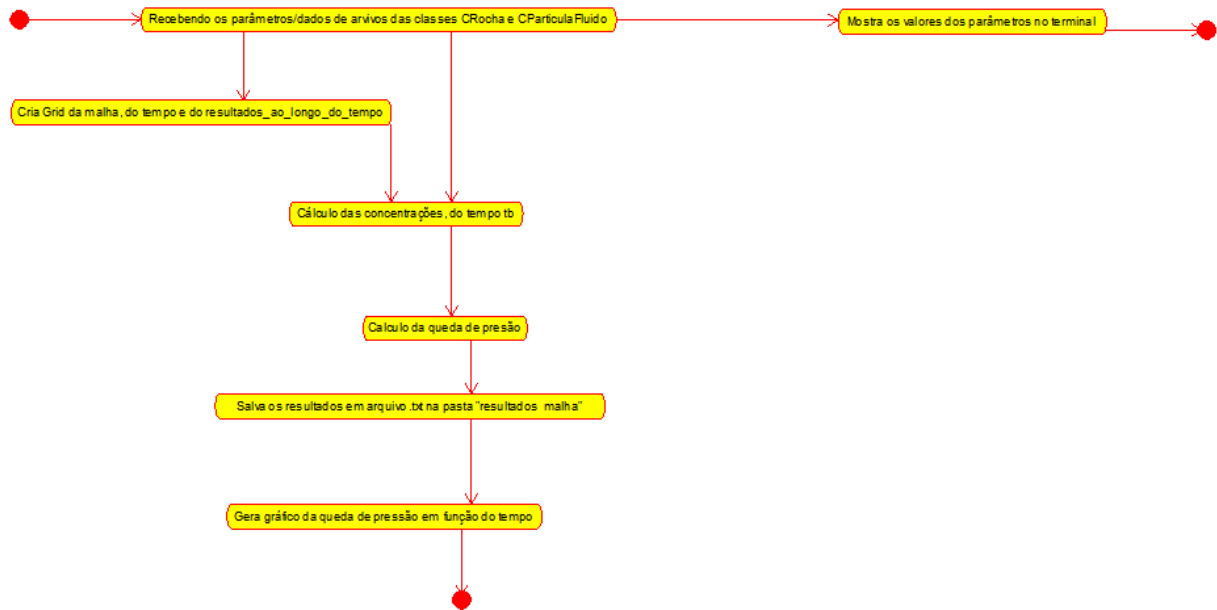


Figura 4.6: Diagrama de máquina de estado para a classe CSimuladorParticulas.

4.5 Diagrama de atividades

Veja na Figura 4.7 o diagrama de atividades correspondente a uma atividade específica do diagrama de máquina de estado. Observe que a partir da entrada de dados é criado a malha do tempo e do espaço, após a criação é calculado a concentração de partículas retidas por exclusão por tamanho. De acordo com a condição do tempo é definido qual método deve ser realizado, e se é necessário utilizar o método numérico de Simpson para solução da integral que não possui solução analítica. Calculada a concentração, levando em consideração todas as condições, os valores obtidos são salvos em arquivos .txt.

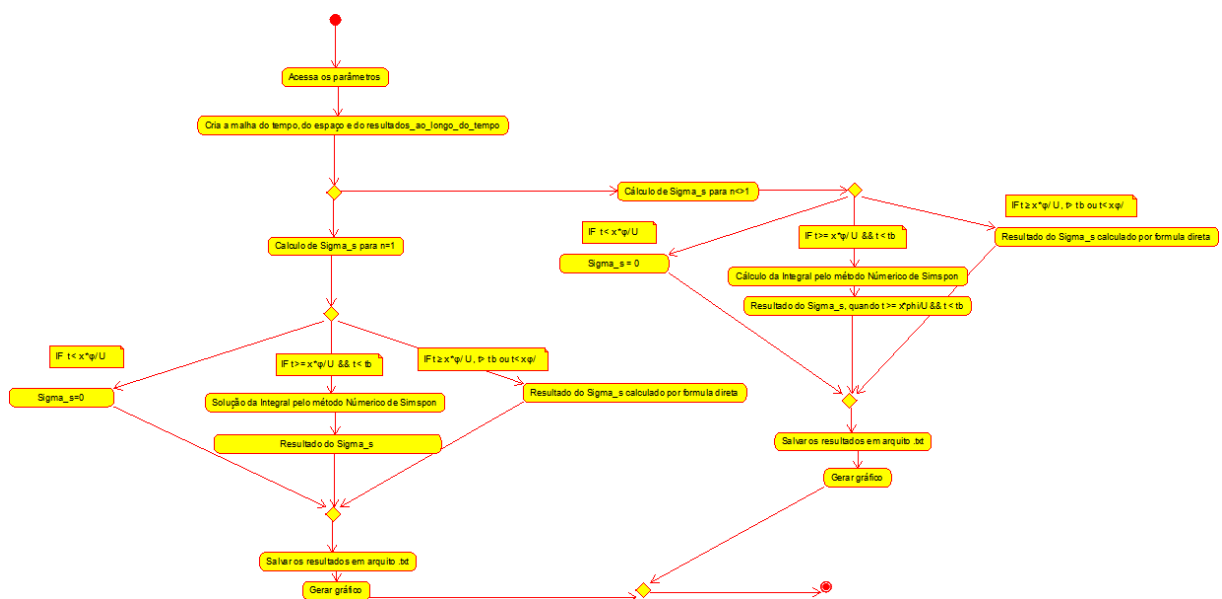


Figura 4.7: Diagrama de atividades para o cálculo da concentração retida por exclusão por tamanho(σ_s).

Capítulo 5

Projeto

Neste capítulo do projeto de engenharia veremos questões associadas ao projeto do sistema, incluindo protocolos, recursos, plataformas suportadas, implicações nos diagramas feitos anteriormente, diagramas de componentes e implantação. Na segunda parte revisamos os diagramas levando em conta as decisões do projeto do sistema.

5.1 Projeto do sistema

Depois da análise orientada a objeto desenvolve-se o projeto do sistema, qual envolve etapas como a definição dos protocolos, da interface API, o uso de recursos, a subdivisão do sistema em subsistemas, a alocação dos subsistemas ao hardware e a seleção das estruturas de controle, a seleção das plataformas do sistema, das bibliotecas externas, dos padrões de projeto, além da tomada de decisões conceituais e políticas que formam a infraestrutura do projeto.

Deve-se definir padrões de documentação, padrões para o nome das classes, padrões de retorno e de parâmetros em métodos, características da interface do usuário e características de desempenho.

O projeto do sistema é a estratégia de alto nível para resolver o problema e elaborar uma solução. O presente projeto foi desenvolvido para apresentar soluções gráficas e via arquivo .txt.

1. Protocolos

- Definição dos protocolos de comunicação entre os diversos elementos externos
 - Neste projeto o software irá se comunicar com o componente externo GNU-PLOT, que tem como finalidade plotar um gráfico da queda de pressão em função do tempo para analisar o comportamento da permeabilidade a medida que ocorre a injeção de água de baixa salinidade.
 - A entrada de dados referente as propriedades da rocha, da partícula e do fluido será efetuada no formato ascii, utilizando a extensão .txt. Enquanto

que outros dados a entrada será através do teclado.

- Definição do formato dos arquivos gerados pelo software.
 - Neste projeto será criado uma pasta com os resultados ao final de sua execução. Nessa pasta será gerado vários arquivos .txt contento as concentrações de acordo com o espaço(x) e o tempo, cada arquivo .txt é referente a um tempo específico, variando apenas o x.
 - O programa salva as imagens dos gráficos no disco no formato .png.

2. Recursos

- Identificação e alocação dos recursos globais, como os recursos do sistema serão alocados, utilizados, compartilhados e liberados. Implicam modificações no diagrama de componentes.
 - Neste projeto será utilizado o HD, o processador, o teclado, a memória, a tela e os demais componentes internos do computador.
 - Neste projeto será utilizado um banco de dados no formato .txt.
- Identificação da necessidade do uso de banco de dados. Implicam em modificações nos diagramas de atividades e de componentes.
 - Neste projeto será necessário o uso do GNUPLOT para plotar o gráfico.

3. Controle

- Identificação da necessidade de otimização. Por exemplo: prefira sistemas com grande capacidade de memória; prefira vários hds pequenos a um grande.
 - Não há necessidade otimização. Porém, dependendo da quantidade do número de pontos o programa pode ser otimizado.
- Identificação e definição de *loops* de controle e das escalas de tempo.
 - O tempo e o espaço é dividido de forma igualmente espaçada.

4. Plataformas

- Identificação e definição das plataformas a serem suportadas: hardware, sistema operacional e linguagem de software.
 - A linguagem utilizada neste projeto será C++, desse modo o programa será multiplataforma. Pode ser executado no Mac OS X, GNU/Linux e Windows.
- Seleção das bibliotecas externas a serem utilizadas.

- Neste projeto será utilizada a biblioteca padrão da linguagem de C++, incluindo `vector`, `iostream`, `string`, `math.h`, `cstdint`, `fstream`, `iomanip`. A partir da classe `CGNUPLOT` contendo a biblioteca externa `CGNUPLOT` será permitido ter acesso ao programa `GNUPLOT`.
- Seleção do ambiente de desenvolvimento para montar a interface de desenvolvimento – IDE.
 - O software DEV C++ será o ambiente de desenvolvimento utilizado para montar a interface de desenvolvimento do programa. O programa será utilizado no sistema operacional do Windows 10 de 64 bits. O compilador será o `gcc/g++`.

5.2 Projeto orientado a objeto – POO

O projeto orientado a objeto é a etapa posterior ao projeto do sistema. Baseia-se na análise, mas considera as decisões do projeto do sistema. Acrescenta a análise desenvolvida e as características da plataforma escolhida (hardware, sistema operacional e linguagem de softwareção). Passa pelo maior detalhamento do funcionamento do software, acrescentando atributos e métodos que envolvem a solução de problemas específicos não identificados durante a análise.

Envolve a otimização da estrutura de dados e dos algoritmos, a minimização do tempo de execução, de memória e de custos. Existe um desvio de ênfase para os conceitos da plataforma selecionada.

Por exemplo: na análise você define que existe um método para salvar um arquivo em disco, define um atributo `nomeDoArquivo`, mas não se preocupa com detalhes específicos da linguagem. Já no projeto, você inclui as bibliotecas necessárias para acesso ao disco, cria um objeto específico para acessar o disco, podendo, portanto, acrescentar novas classes àquelas desenvolvidas na análise.

Efeitos do projeto no modelo estrutural

- Adicionar nos diagramas de pacotes as bibliotecas e subsistemas selecionados no projeto do sistema (exemplo: a biblioteca gráfica selecionada).
 - Neste projeto a biblioteca gráfica selecionada foi a `CGnuplot`, sendo necessária a instalação do software `Gnuplot` para plotar os gráficos.
 - `vector`: armazenamento de dados por meio de vetores.
 - `iostream`: entrada e saída de dados, pelo teclado e pela tela respectivamente.
 - `string`: utilizada para conversão de um valor número em uma sequência de caracteres.

- `cmath.h`: possui funções matemáticas visando a realização de cálculos.
- `fstream`: estabelece um canal de comunicação entre o arquivo e o programa. Utilizado para gravar e ler arquivos de disco.
- `iomanip`: fornece recursos para manipular a formatação de saída.
- Novas classes e associações oriundas das bibliotecas selecionadas e da linguagem escolhida devem ser acrescentadas ao modelo.
 - Neste projeto o usuário acessa o arquivo dos dados de entrada por meio do nome dos arquivos `.txt`, utilizando as bibliotecas `fstream`, `string` etc.
- Estabelecer as dependências e restrições associadas à plataforma escolhida.
 - O software pode ser executado por meio das plataformas: GNU/Linux, MAC OS ou Windows.

Efeitos do projeto nos atributos

- Atributos novos podem ser adicionados a uma classe, como, por exemplo, atributos específicos de uma determinada linguagem de softwareção (acesso a disco, ponteiros, constantes e informações correlacionadas).
 - Neste projeto foi necessário adicionar alguns atributos e inserir via teclado para facilitar a alteração e execução do programa.

Efeitos do projeto nos métodos

- Em função da plataforma escolhida, verifique as possíveis alterações nos métodos. O projeto do sistema costuma afetar os métodos de acesso aos diversos dispositivos (exemplo: `hd`, `rede`).
 - Neste projeto foi necessário adicionar um método para salvar em disco o resultado das integrais para teste dos valores obtidos com os valores reais, a fim de validar o método de numérico de integração.
 - Métodos para salvar em arquivo `.dat` foram adicionados, a fim de obter todos os valores calculados e obtidos permitindo comparação e cálculo para verificação.

Efeitos do projeto nas heranças

- Reorganização das classes e dos métodos (criar métodos genéricos com parâmetros que nem sempre são necessários e englobam métodos existentes).

- Foi realizado uma reorganização das classes para facilitar o acesso ao banco de dados, dessa forma foi criada uma herança para que a Classe CSimuladorParticulas consiga herdar todos os parâmetros da Classe CRocha e CParticulaFluido, visando acelerar o acesso às variáveis.
- Revise as heranças no diagrama de classes.
 - A classe CSimuladorParticulas é herdeira das classes CRocha e CParticulaFluido, a fim de acessar os valores das variáveis para cálculo das concentrações.
 - A classe MetodoSimpson é herdeira da classe MetodoIntegracaoNumerica1D. E as classes Funcao_Sigma_n_diferente_1, Funcao_Sigma_n_1 são herdeiras da classe Funcao1x1.

Efeitos do projeto nas associações

- Não houve alteração nas associações do projeto.

Efeitos do projeto nas otimizações

- Não foi necessário rever nesta etapa do projeto.

5.3 Diagrama de componentes

O diagrama de componentes mostra a forma como os componentes do software se relacionam, suas dependências. Inclui itens como: componentes, subsistemas, executáveis, nós, associações, dependências, generalizações, restrições e notas. Exemplos de componentes são bibliotecas estáticas, bibliotecas dinâmicas, dlls, componentes Java, executáveis, arquivos de disco, código-fonte.

Veja na Figura 5.1 um exemplo de diagrama de componentes. Observe que este inclui muitas dependências, ilustrando as relações entre os arquivos. Por exemplo: o subsistema biblioteca inclui os arquivos das classes A e B, e a geração dos objetos A.obj e B.obj depende dos arquivos A.h, A.cpp, B.h e B.cpp. A geração da biblioteca depende dos arquivos A.obj e B.obj. O subsistema biblioteca Qt, um subsistema externo, inclui os arquivos de código da biblioteca Qt e a biblioteca em si. O subsistema banco de dados representa o banco de dados utilizado pelo sistema e tem uma interface de acesso que é utilizada pelo software para acesso aos dados armazenados no banco de dados. O software executável a ser gerado depende da biblioteca gerada, dos arquivos da biblioteca Qt, do módulo de arquivos MinhaJanela e do banco de dados.

Algumas observações úteis para o diagrama de componentes:

- De posse do diagrama de componentes, temos a lista de todos os arquivos necessários para compilar e rodar o software.

- Observe que um assunto/pacote pode se transformar em uma biblioteca e será incluído no diagrama de componentes.
- A ligação entre componentes pode incluir um estereótipo indicando o tipo de relacionamento ou algum protocolo utilizado.

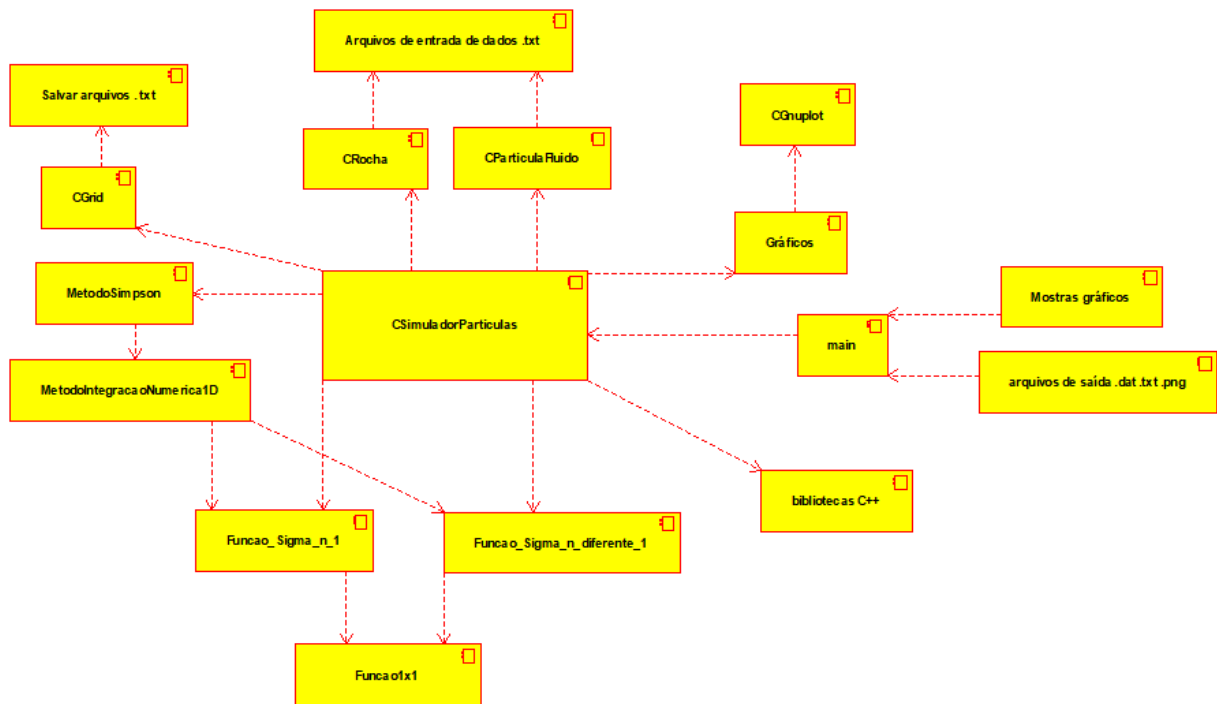


Figura 5.1: Diagrama de componentes

5.4 Diagrama de implantação

O diagrama de implantação é um diagrama de alto nível que inclui relações entre o sistema e o hardware e que se preocupa com os aspectos da arquitetura computacional escolhida. Seu enfoque é o hardware, a configuração dos nós em tempo de execução.

O diagrama de implantação deve incluir os elementos necessários para que o sistema seja colocado em funcionamento: computador, periféricos, processadores, dispositivos, nós, relacionamentos de dependência, associação, componentes, subsistemas, restrições e notas.

A Figura 5.2 mostra o diagrama de implantação utilizado. Os dados foram obtidos da literatura e foram armazenados em arquivos .txt no computador. O programa importa os arquivos para acessar os dados e utiliza o monitor para comunicar com o usuário. Os gráficos e arquivos obtidos são salvos no disco rígido.

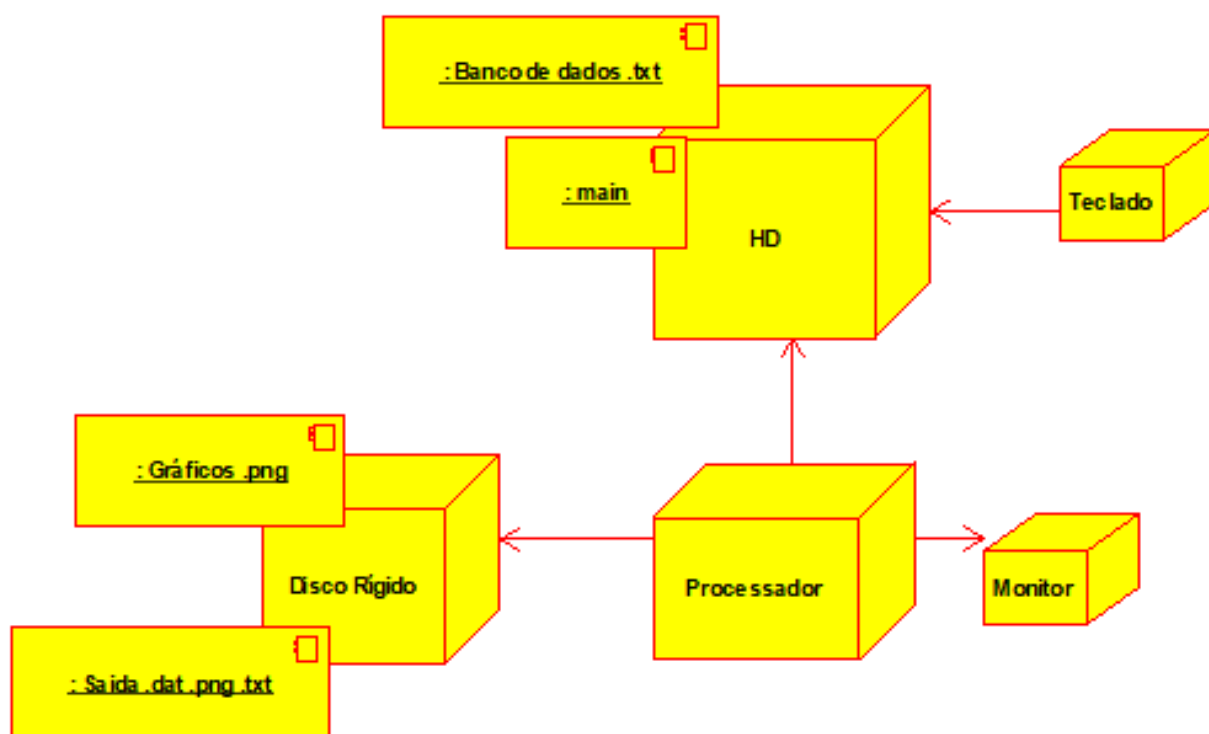


Figura 5.2: Diagrama de implantação

Capítulo 6

Implementação

Neste capítulo do projeto de engenharia apresentamos os códigos fonte que foram desenvolvidos.

6.1 Código fonte

Apresenta-se a seguir um conjunto de classes (arquivos .h e .cpp) além do programa main.

Apresenta-se na listagem 6.1 o arquivo com código da classe CParticulaFluido.hpp.

Listing 6.1: Arquivo de cabeçalho da classe CParticulaFluido.hpp

```
1 #ifndef PARTICULAFLUIDO_HPP
2 #define PARTICULAFLUIDO_HPP
3
4 #include <fstream>
5 #include <iostream>
6 #include <string>
7
8 class CParticulaFluido {
9 protected:
10     double viscosidade;
11     double lambdaPonte;
12     double lambdaAdesao;
13     double parC;
14     double par_n;
15     double velocidade;
16     double C = 2.0;
17     double cb = 0.0004;
18
19     void readFile(std::string path);
20
```

```

21 public:
22     // CParticulaFluido() {}
23     // construtores por variáveis e por nome do arquivo txt
24     CParticulaFluido(double _mu, double _lambdaPonte, double
        _lambdaAdesao, double _parC, double _par_n, double
        _velocidade):
25         viscosidade{ _mu }, lambdaPonte{ _lambdaPonte },
        lambdaAdesao{ _lambdaAdesao }, parC{ _parC },
        par_n{ _par_n }, velocidade{ _velocidade } {}
26
27     CParticulaFluido(std::string path) { readFile(path); }
28
29     // funções get-set
30     double getViscosidade() { return viscosidade; };
31     void setViscosidade(double novo_mu) { viscosidade = novo_mu
        ; }
32
33     double getLambdaPonte() { return lambdaPonte; }
34     double getLambdaAdesao() { return lambdaAdesao; }
35     double getParC() { return parC; }
36     double getParN() { return par_n; }
37     double getVelocidade() { return velocidade; }
38
39     // função para mostrar os parâmetros da classe
40     void printCParticulaFluido();
41 };
42
43 #endif

```

Apresenta-se na listagem 6.2 o arquivo de implementação da classe CParticulaFluido.cpp.

Listing 6.2: Arquivo de implementação da classe CParticulaFluido.cpp

```

1 #include "CParticulaFluido.hpp"
2
3 void CParticulaFluido::readFile(std::string path) {
4     std::ifstream infile;
5     infile.open(path);
6     std::string temp;
7     std::getline(infile, temp); // pular linha com texto
8     std::getline(infile, temp); // viscosidade
9     viscosidade = atof(temp.c_str());
10
11     std::getline(infile, temp); // pular linha com texto

```

```

12         std::getline(infile, temp); // lambdaPonte
13         lambdaPonte = atof(temp.c_str());
14
15         std::getline(infile, temp); // pular linha com texto
16         std::getline(infile, temp); // lambdaAdesao
17         lambdaAdesao = atof(temp.c_str());
18
19         std::getline(infile, temp); // pular linha com texto
20         std::getline(infile, temp); // parC
21         parC = atof(temp.c_str());
22
23         std::getline(infile, temp); // pular linha com texto
24         std::getline(infile, temp); // par_n
25         par_n = atof(temp.c_str());
26
27         std::getline(infile, temp); // pular linha com texto
28         std::getline(infile, temp); // velocidade
29         velocidade = atof(temp.c_str());
30
31     }
32
33 void CParticulaFluido::printCParticulaFluido() {
34     std::cout << "\n-----" << std::endl;
35     std::cout << "Valores de ParticulaFluido" << std::endl;
36     std::cout << "-----" << std::endl;
37     std::cout << "viscosidade:" << viscosidade << std::endl;
38     std::cout << "lambdaPonte:" << lambdaPonte << std::endl;
39     std::cout << "lambdaAdesao:" << lambdaAdesao << std::endl;
40     std::cout << "parC:" << parC << std::endl;
41     std::cout << "par_n:" << par_n << std::endl;
42     std::cout << "velocidade:" << velocidade << std::endl;
43 }

```

Apresenta-se na listagem 6.3 o arquivo de implementação da classe CRocha.hpp.

Listing 6.3: Arquivo de implementação da classe CRocha.hpp

```

1 #ifndef CROCHA_HPP
2 #define CROCHA_HPP
3
4 #include <fstream>
5 #include <iostream>
6 #include <string>
7

```

```

8 class CRocha{
9 protected:
10     double permeabilidade;
11     double porosidade;
12     double sigma_am;
13     double N;
14     double beta = 0.5;
15     double porcentagemCaulinita = 0.2;
16
17     double sigma_a0;
18
19     void readFile(std::string path);
20 public:
21     //CRocha() {}
22     // construtores por variáveis e por nome do arquivo txt
23     CRocha(double k, double fi, double _sigma_a0, double
        _sigma_am, double _N); // : permeabilidade{ k },
        porosidade{ fi }, sigma_a0{ _sigma_a0 }, sigma_am{
        _sigma_am }, N{ _N } {}
24     CRocha(std::string path) {
25         readFile(path); }
26
27     double reducaoPermeabilidade(double sigma) { return 1 / (1
        + beta * sigma); }
28
29     // função para mostrar os parâmetros da classe
30     void printCRocha();
31 };
32
33 #endif

```

Apresenta-se na listagem 6.4 o arquivo de implementação da classe CRocha.cpp.

Listing 6.4: Arquivo de implementação da classe CRocha.cpp

```

1 #include "CRocha.hpp"
2
3 CRocha::CRocha(double k, double fi, double _sigma_a0, double
    _sigma_am, double _N) :
4     permeabilidade{ k }, porosidade{ fi }, sigma_a0{
        _sigma_a0 }, sigma_am{ _sigma_am }, N{ _N } {
5     sigma_a0 = (1 - porosidade) * porcentagemCaulinita;
6 }
7

```

```

8 void CRocha::readFile(std::string path) {
9     std::ifstream infile;
10    infile.open(path);
11    std::string temp;
12    std::getline(infile, temp); // pular linha com texto
13    std::getline(infile, temp); // permeabilidade
14    permeabilidade = atof(temp.c_str());
15
16    std::getline(infile, temp); // pular linha com texto
17    std::getline(infile, temp); // porosidade
18    porosidade = atof(temp.c_str());
19
20    std::getline(infile, temp); // pular linha com texto
21    std::getline(infile, temp); // sigma_a0
22    sigma_a0 = atof(temp.c_str());
23
24    std::getline(infile, temp); // pular linha com texto
25    std::getline(infile, temp); // sigma_am
26    sigma_am = atof(temp.c_str());
27
28    std::getline(infile, temp); // pular linha com texto
29    std::getline(infile, temp); // N
30    N = atof(temp.c_str());
31 }
32
33 void CRocha::printCRocha() {
34     std::cout << "\n-----" << std::endl;
35     std::cout << "Valores de CRocha" << std::endl;
36     std::cout << "-----" << std::endl;
37     std::cout << "permeabilidade:" << permeabilidade << std::endl;
38     std::cout << "porosidade:" << porosidade << std::endl;
39     std::cout << "sigma_a0:" << sigma_a0 << std::endl;
40     std::cout << "sigma_am:" << sigma_am << std::endl;
41     std::cout << "N:" << N << std::endl;
42 }

```

Apresenta-se na listagem 6.5 o arquivo de implementação da classe CGrid.hpp.

Listing 6.5: Arquivo de implementação da classe CGrid.cpp

```

1 #ifndef CGRID_HPP
2 #define CGRID_HPP
3

```

```

4#include <string>
5#include <vector>
6#include <fstream> /// escrever em disco
7#include <iomanip>    /// setw
8
9class CGrid {
10private:
11    int _size;
12public:
13    std::vector<double> c_susp, sigma_a, sigma_s, diff_sigma_a,
        concentracao;
14
15public:
16    CGrid(int size_x) {
17        _size = size_x;
18        c_susp.resize(_size, 0.0);
19        sigma_a.resize(_size, 0.0);
20        sigma_s.resize(_size, 0.0);
21        diff_sigma_a.resize(_size, 0.0);
22        concentracao.resize(_size, 0.0);
23    }
24
25    int get_size() { return _size;}
26
27    void saveGrid(std::vector<double> malha, double time);
28    std::vector<double> get_sigma_s() { return sigma_s; }
29    /// metodo para criar malhas, eh static para criar as
        malhas do espaco e do tempo
30    std::vector<double> static linspace(double start, double
        end, int size);
31};
32#endif

```

Apresenta-se na listagem 6.6 o arquivo de implementação da classe CGrid.cpp.

Listing 6.6: Arquivo de implementação da classe CGrid.cpp

```

1#include "CGrid.hpp"
2
3void CGrid::saveGrid(std::vector<double> malha, double time) {
4    std::string path = "resultados_malha//malha_" + std::
        to_string(time) + ".txt";
5
6    std::ofstream file(path);

```

```

7      file << "uuuuu malhauu-uu sigma_auu-uu sigma_su- diff_sigma_a\
          n";
8      for (unsigned int i = 0; i < _size; i++) {
9          file << std::setw(9) << malha[i]
10             //<< std::setw(12) << c_susp[i]
11             << std::setw(12) << sigma_a[i]
12             << std::setw(12) << sigma_s[i]
13             << std::setw(15) << diff_sigma_a[i]
14             //<< std::setw(15) << concentracao[i] << "\
                n";
15      }
16      file.close();
17 }
18
19 std::vector<double> CGrid::linspace(double start, double end, int
    size) {
20     double delta = (end - start) / (1.0 * size - 1);
21     std::vector<double> grid;
22
23     for (int i = 0; i < size; i++)
24         grid.push_back(start + delta * i);
25
26     return grid;
27 }

```

Apresenta-se na listagem 6.7 o arquivo de implementação da classe CSimuladorParticulas.hpp.

Listing 6.7: Arquivo de implementação da classe CSimuladorParticulas.cpp

```

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;
19     size_t size_malha;
20     size_t numPontosIntegral=11;
21     double start_x, end_x, start_t, end_t;
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     /// CONSTRUTORES
30     CSimuladorParticulas(std::string pathParticulaFluido, std::
        string pathRocha);
31
32     /// 'MAIN' metodo, ele que esta executando o objeto
33     void run();
34
35     /// CALCULOS
36 private:
37     void readFile(std::string pathSimulador);
38     double CalculoSigma_a(double x, double t);
39     double CalculoDiffSigma_a(double x, double t);
40     double CalculoLinhaZona(double x);
41     double CalculoTb(double x);
42     //double CalculoConcentracoes(double x, double t);
43     //double CalculoConcentracoes_N_igual_1(double x, double t)
        ;
44     //double CalculoConcentracoes_N_diferente_1(double x,
        double t);
45     double CalculoDeltaPressao(double t);
46
47     double CalculoSigma(double x, double t);
48     double CalculoSigma_N_igual_1(double x, double t);
49     double CalculoSigma_N_diferente_1(double x, double t);
50     double funcao_sigma_n1(double x, double t);
51     double funcao_sigma_n_dif_1(double x, double t);
52
53 public:

```



```

54      /// metodos para SALVAR e APRESENTAR os resultados
55      void printCSimuladorParticulas();
56      void print_vector(std::vector<double> vetor);
57
58      void saveInFile(std::vector<double> vector1, std::string
          name_vector1);
59      void saveInFile(std::vector<double> vector1, std::vector<
          double> vector2, std::string name_vector1, std::string
          name_vector2);
60
61      void plot(std::vector<double> vector1, std::vector<double>
          vector2, std::string name_vector1, std::string
          name_vector2);
62};
63#endif

```

Apresenta-se na listagem 6.8 o arquivo de implementação da classe CSimuladorParticulas.cpp.

Listing 6.8: Arquivo de implementação da classe CSimuladorParticulas.cpp

```

1#include "CSimuladorParticulas.hpp"
2
3CSimuladorParticulas::CSimuladorParticulas(std::string
    pathParticulaFluido, std::string pathRocha):CParticulaFluido(
    pathParticulaFluido), CRocha(pathRocha) {
4    std::cout << "-----" << std::
        endl;
5    std::cout << "----SIMULADOR_DE_PARTÍCULAS----" << std::
        endl;
6    std::cout << "-----" << std::
        endl;
7    std::cout << "Deseja_inserir_os_valores_da_simulacao_
        manualmente?(s/n)" << std::endl;
8    std::cout << "-->";
9    std::string opt;
10   std::cin >> opt;
11   if (opt == "s") {
12       std::cout << "Numero_de_tempos:";
13       std::cin >> size_tempo;
14       std::cout << "Numero_de_espaco:";
15       std::cin >> size_malha;
16       std::cout << "Posicao_inicial_testemunho:";
17       std::cin >> start_x;

```

```

18         std::cout << "Posicao_final_testemunho: ";
19         std::cin >> end_x;
20         std::cout << "Tempo_inicial: ";
21         std::cin >> start_t;
22         std::cout << "Tempo_final: ";
23         std::cin >> end_t;
24     }
25     else {
26         std::cout << "Informe o nome do arquivo: "
27             "ParticulaFluido ";
28         std::string pathSimulador;
29         std::cin >> pathSimulador;
30         readFile(pathSimulador);
31     }
32
33     tempo = CGrid::linspace(start_t, end_t, size_tempo); ///
34     nao eh objeto CGrid, eh vetor - metodo static
35     malha = CGrid::linspace(start_x, end_x, size_malha);
36     /// inicio as malhas
37     resultados_ao_longo_do_tempo.resize(size_tempo);
38     for (unsigned int i = 0; i < size_tempo; i++)
39         resultados_ao_longo_do_tempo[i] = new CGrid(
40             size_malha);
41
42 void CSimuladorParticulas::run() {
43     int nx = resultados_ao_longo_do_tempo[0]->get_size();
44     std::vector<double> linhazona(nx);
45     std::vector<double> tb(nx);
46     std::vector<double> delta_p(size_tempo);
47
48     for (unsigned int j = 0; j < size_tempo; j++) { /// loop
49         dos tempos
50         for (unsigned int i = 0; i < nx; i++) { /// loop ao
51             longo do testemunho
52             resultados_ao_longo_do_tempo[j]->sigma_a[i]
53                 = CalculoSigma_a(malha[i], tempo[j]);
54             resultados_ao_longo_do_tempo[j]->
55                 diff_sigma_a[i] = CalculoDiffSigma_a(
56                     malha[i], tempo[j]);
57             //resultados_ao_longo_do_tempo[j]->
58                 concentracao[i] = CalculoConcentracoes(

```

```

        malha[i], tempo[j]);
51      resultados_ao_longo_do_tempo[j]->sigma_s[i]
        = CalculoSigma(malha[i], tempo[j]);
52      linhazona[i] = CalculoLinhaZona(malha[i]);
53      tb[i] = CalculoTb(malha[i]);
54    }
55    delta_p[j] = CalculoDeltaPressao(tempo[j]);
56    resultados_ao_longo_do_tempo[j]->saveGrid(malha,
        tempo[j]); /// os resultados são salvos em
        arquivo .txt
57  }
58  std::cout << "Vetor de tb:" << std::endl;
59  print_vector(tb);
60  saveInFile(tempo, "tempo");
61  saveInFile(malha, "grid");
62  plot(malha, resultados_ao_longo_do_tempo[size_tempo - 1]->
        get_sigma_s(), "grid", "sigma_s");
63  plot(tempo, delta_p, "tempo", "dp");
64}
65
66/// abaixo estao os calculos do simulador
67
68double CSimuladorParticulas::CalculoSigma_a(double x, double t){
69    if (N == 1) {
70        if (t < (x * porosidade / velocidade))
71            return sigma_a0;
72        else {
73            return sigma_am + (sigma_a0 - sigma_am) *
                exp(-parC * (t - x * porosidade /
                velocidade));
74        }
75    }
76    else {
77        if (t < (x * porosidade / velocidade))
78            return sigma_a0;
79        else {
80            return pow(sigma_am+pow(sigma_a0-sigma_am,
                1- N) - parC *(1- N)*(t-x* porosidade /
                velocidade), N / (1- N));
81        }
82    }
83}

```

```

84
85 double CSimuladorParticulas::CalculoDiffSigma_a(double x, double t)
    {
86     if (N == 1) {
87         if (t < (x * porosidade / velocidade))
88             return 0.0;
89         else {
90             return -parC *(sigma_a0 - sigma_am) * exp(-
                parC * (t - x * porosidade / velocidade)
            );
91         }
92     }
93     else {
94         if (t < (x * porosidade / velocidade))
95             return 0.0;
96         else {
97             return -parC *pow( pow(sigma_a0 - sigma_am,
                1 - N) - parC * (1 - N) * (t - x *
                porosidade / velocidade), N / (1 - N));
98         }
99     }
100 }
101
102 double CSimuladorParticulas::CalculoLinhaZona(double x) {
103     return porosidade * x / velocidade;
104 }
105
106 double CSimuladorParticulas::CalculoTb(double x) {
107     double cb = 0.0004; /// rocha
108     double linhaZona = CalculoLinhaZona(x);
109
110     double tb;
111     if (N == 1)
112         tb = linhaZona + (1/ parC) * log(parC *(sigma_a0 -
            sigma_am)*(1-exp(-lambdaPonte *x))/(cb*
            velocidade * lambdaPonte));
113     else
114         tb = 2 * linhaZona + pow(sigma_a0 - sigma_am, 1 - N
            ) / (parC * (1 - N)) - (1 / (parC * (1 - N))) *
            pow(lambdaPonte * velocidade * cb / (parC * (1 -
            exp(-x * lambdaPonte))), (1 - N) / N);
115

```

```

116         return tb;
117     }
118
119 //double CSimuladorParticulas::CalculoConcentracoes(double x,
120 //    double t) {
121 //    if (N == 1)
122 //        return CalculoConcentracoes_N_igual_1(x, t);
123 //    else
124 //        return CalculoConcentracoes_N_diferente_1(x, t);
125 //}
126
127 //double CSimuladorParticulas::CalculoConcentracoes_N_igual_1(
128 //    double x, double t) {
129 //    double concentracao;
130 //    if (t < CalculoTb(x))
131 //        concentracao = 0.0;
132 //    else {
133 //        if (C < cb)
134 //            concentracao = C * (sigma_a0 - sigma_am) *
135 //                exp(-C * (t - CalculoTb(x))) * (1 - exp(-lambdaAdesao * x)) / (
136 //                    velocidade * lambdaAdesao);
137 //        else
138 //            concentracao = C * (sigma_a0 - sigma_am) *
139 //                exp(-C * t - CalculoTb(x)) / (velocidade * (lambdaAdesao +
140 //                    lambdaPonte)) * (1 - (1 - velocidade * (lambdaAdesao +
141 //                        lambdaPonte) * cb / (C * (sigma_a0 - sigma_am) * exp(-C * (t -
142 //                            CalculoTb(x))))) / pow(1 - velocidade * lambdaAdesao * cb / (C *
143 //                                (sigma_a0 - sigma_am) * exp(-C * (t - CalculoTb(x)))), 1 +
144 //                                    lambdaPonte / lambdaAdesao) * exp((lambdaAdesao + lambdaPonte) *
145 //                                        x));
146 //    }
147 //    return concentracao;
148 //}
149
150 //double CSimuladorParticulas::CalculoConcentracoes_N_diferente_1(
151 //    double x, double t) {
152 //    double concentracao;
153 //    if (t < CalculoTb(x))
154 //        concentracao = 0.0;
155 //    else {
156 //        if (C < cb)

```

```

146 //                                concentracao = C * pow(pow(sigma_a0 -
sigma_am, 1 - N) - C * (1 - N) * (t - 2 * CalculoTb(x)), N / (N
- 1)) * (1 - exp(-x * lambdaAdesao)) / (lambdaAdesao *
velocidade);
147 //                                else {
148 //                                double c1 = C * pow(pow(sigma_a0 - sigma_am
, 1 - N) - C * (1 - N) * (t - 2 * CalculoTb(x)), N / (N - 1));
149 //                                double esquerda = c1 / (velocidade * (
lambdaAdesao + lambdaPonte));
150 //                                double direita = 1-(1 - (velocidade*(
lambdaAdesao+lambdaPonte))/c1)*exp(-x*(lambdaAdesao+lambdaPonte)
)/ pow(1-lambdaAdesao*velocidade*cb/c1,1+lambdaPonte/
lambdaAdesao);
151 //                                concentracao = esquerda * direita;
152 //                                }
153 //                                }
154 //                                return concentracao;
155 //}

156
157 double CSimuladorParticulas::CalculoDeltaPressao(double t) {
158     double integral = 0.0;
159     double const1 = viscosidade * velocidade * end_x /
permeabilidade;
160     for (int i = 0; i < size_malha; i++) {
161         integral += CalculoSigma(malha[i], t)*(malha[1]-
malha[0]);
162     }
163     return -1*(const1 + const1 * beta * integral);
164 }

165
166 double CSimuladorParticulas::CalculoSigma(double x, double t) {
167     if (N == 1)
168         return CalculoSigma_N_igual_1(x, t);
169     else
170         return CalculoSigma_N_diferente_1(x, t);
171 }

172
173 double CSimuladorParticulas::CalculoSigma_N_igual_1(double x,
double t) {
174     double sigma = 0.0;
175     double tb = CalculoTb(x);
176     double C0 = x * porosidade / velocidade;

```

```

177         if (t < C0)
178             sigma = 0.0;
179         else if (t >= C0 && t < tb) {
180             sigma = funcao_sigma_n1(x, t);
181         }
182         else {
183             double maxT = tb > C0 ? tb : C0;
184             sigma = funcao_sigma_n1(x, maxT) + (sigma_a0 -
                sigma_am) * (1 - exp(-lambdaAdesao * x)) * exp(C
                * porosidade * x / velocidade) * (exp(-C * maxT
                ) - exp(-C * t));
185         }
186         return sigma;
187     }
188
189     double CSimuladorParticulas::CalculoSigma_N_diferente_1(double x,
        double t) {
190         double sigma = 0.0;
191         double tb = CalculoTb(x);
192         double C0 = x * porosidade / velocidade;
193         if (t < C0)
194             sigma = 0.0;
195         else if (t >= C0 && t < tb) {
196             sigma = funcao_sigma_n1(x, t);
197         }
198         else {
199             double maxT = tb > C0 ? tb : C0;
200             sigma = funcao_sigma_n1(x, maxT) +(1-exp(-x*
                lambdaAdesao))*(N-1)/(1-N)*pow(pow(sigma_a0 -
                sigma_am, 1-N) - C*(1-N)*(t-2*porosidade*x/
                velocidade), 1/(1-N));
201         }
202         return sigma;
203     }
204
205     double CSimuladorParticulas::funcao_sigma_n1(double x, double t) {
206         Funcao_Sigma_n_1 funcao(velocidade, lambdaAdesao,
            lambdaPonte, cb, C, sigma_a0, sigma_am, porosidade, x);
207         MetodoIntegracaoNumerica1D* metodo = new MetodoSimpson(
            funcao);
208         double integral = metodo->Integrar(porosidade * x /
            velocidade, t, numPontosIntegral);

```

```

209         return C * (sigma_a0 - sigma_am) * ((1 - exp(-C * (t -
            porosidade * x / velocidade))) / C) + exp(-(lambdaAdesao
            + lambdaPonte - C * (x * porosidade / velocidade) * x))
            * integral);
210     }
211
212 double CSimuladorParticulas::funcao_sigma_n_dif_1(double x, double
    t) {
213     /// na linha abaixo, é criado a funcao relacionado a funcao
        sigma
214     Funcao_Sigma_n_diferente_1 funcao(velocidade, lambdaAdesao,
        lambdaPonte, cb, C, sigma_a0, sigma_am, porosidade, x,
        N);
215
216     /// na linha abaixo, é criado o método de simpson, e é
        enviada a função criada acima
217     MetodoIntegracaoNumericalD* metodo = new MetodoSimpson(
        funcao);
218     // na linha abaixo, é executado o método para integrar
219     double integral = metodo->Integrar(porosidade * x /
        velocidade, t, numPontosIntegral);
220     return C * integral;
221 }
222
223 void CSimuladorParticulas::printCSimuladorParticulas() {
224     printCParticulaFluido();
225     printCRocha();
226     std::cout << "\n-----" << std::endl;
227     std::cout << "Classe da simulacao:" << std::endl;
228     std::cout << "-----" << std::endl;
229
230     std::cout << "Grid dos tempos:" << std::endl;
231     print_vector(tempo);
232 }
233
234 void CSimuladorParticulas::saveInFile(std::vector<double> vector1,
    std::string name_vector1) {
235     std::ofstream outdata; //save data
236     outdata.open((name_vector1 + ".dat").c_str());
237     outdata << "# " << name_vector1 << std::endl;
238     for (unsigned int i = 0; i < vector1.size(); i++)
239         outdata << vector1[i] << std::endl;

```



```

240         outdata.close();
241     }
242
243 void CSimuladorParticulas::saveInFile(std::vector<double> vector1,
    std::vector<double> vector2, std::string name_vector1, std::
    string name_vector2){
244     if (vector1.size() != vector2.size()) {
245         std::cout << "Nao_foi_possivel_salvar_os_vetores,
                por_terem_tamanhos_distintos!"<<std::endl;
246         return;
247     }
248
249     std::ofstream outdata; //save data
250     outdata.open((name_vector1+"_" + name_vector2 + ".dat").
        c_str());
251     outdata << "#_" << name_vector1 << "_" << name_vector2 <<
        std::endl;
252     for (unsigned int i = 0; i < vector1.size(); i++)
253         outdata << vector1[i] << "_" << vector2[i] << std::
            endl;
254     outdata.close();
255 }
256
257 void CSimuladorParticulas::print_vector(std::vector<double> vetor)
    {
258     std::cout << vetor[0]; /// este primeiro nao fica dentro do
        loop por causa do ' - '
259     for (unsigned int i = 1; i < vetor.size(); i++)
260         std::cout << "__" << vetor[i];
261     std::cout << std::endl;
262 }
263
264 void CSimuladorParticulas::plot(std::vector<double> vector1, std::
    vector<double> vector2, std::string name_vector1, std::string
    name_vector2) {
265     saveInFile(vector1, vector2, name_vector1, name_vector2);
266     CGnuplot::plot((name_vector1 + "_" + name_vector2 + ".dat")
        .c_str(), name_vector1, name_vector2, (name_vector1 + "_"
        + name_vector2 + ".png").c_str());
267 }
268
269 void CSimuladorParticulas::readFile(std::string pathSimulador) {

```

```

270
271     std::ifstream infile;
272     infile.open(pathSimulador);
273     std::string temp;
274     std::getline(infile, temp); // pular linha com texto
275     std::getline(infile, temp); // size tempo
276     size_tempo = atoi(temp.c_str());
277
278     std::getline(infile, temp); // pular linha com texto
279     std::getline(infile, temp); // size malha
280     size_malha = atoi(temp.c_str());
281
282     std::getline(infile, temp); // pular linha com texto
283     std::getline(infile, temp); // num pontos
284     integral
285     numPontosIntegral = atoi(temp.c_str());
286
287     std::getline(infile, temp); // pular linha com texto
288     std::getline(infile, temp); // start_x
289     start_x = atof(temp.c_str());
290
291     std::getline(infile, temp); // pular linha com texto
292     std::getline(infile, temp); // end_x
293     end_x = atof(temp.c_str());
294
295     std::getline(infile, temp); // pular linha com texto
296     std::getline(infile, temp); // start_t
297     start_t = atof(temp.c_str());
298
299     std::getline(infile, temp); // pular linha com texto
300     std::getline(infile, temp); // end_t
301     end_t = atof(temp.c_str());
302 }

```

Apresenta-se na listagem 6.9 o arquivo de implementação da classe Função1x1, da classe Funcao_Sigma_n_1 e da classe Funcao_Sigma_n_diferente_1.

Listing 6.9: Arquivo de implementação da classe funcao.cpp

```

1 #ifndef FUNCAO_HPP_
2 #define FUNCAO_HPP_
3
4 #include <math.h>
5 class Funcao1x1 {

```

```

6 public:
7     Funcao1x1() {}
8     Funcao1x1(double, double, double, double, double, double,
9         double, double, double) {}
10    Funcao1x1(double, double, double, double, double, double,
11        double, double, double, double) {}
12    virtual double operator()(double) = 0; // Funcao virtual pura
13};
14
15 class Funcao_Sigma_n_1 : public Funcao1x1 {
16 public:
17     Funcao_Sigma_n_1(double _U, double _lambda_s, double _lambda_b,
18         double _cb, double _C, double _sigma_a0, double _sigma_am,
19         double _phi, double _x):
20         U{ _U }, lambda_s{ _lambda_s }, lambda_b{ _lambda_b }, cb{
21             _cb }, C{ _C }, sigma_a0{ _sigma_a0 }, sigma_am{
22                 _sigma_am }, phi{ _phi }, x{ _x }{}
23
24     virtual double operator()(double t) {
25         return (U * (lambda_s + lambda_b) * cb / (C * (sigma_a0 -
26             sigma_am) * exp(C * phi * x / U)) - exp(-C * t)) / (1 -
27             U * lambda_s * cb / (C * (sigma_a0 * sigma_am) * exp(C *
28                 phi * x / U)));
29     }
30 private:
31     double U, lambda_s, lambda_b, cb, C, sigma_a0, sigma_am, phi, x
32     ;
33 };
34
35 class Funcao_Sigma_n_diferente_1 : public Funcao1x1 {
36 public:
37     Funcao_Sigma_n_diferente_1(double _U, double _lambda_s, double
38         _lambda_b, double _cb, double _C, double _sigma_a0, double
39         _sigma_am, double _phi, double _x, double _n) :
40         U{ _U }, lambda_s{ _lambda_s }, lambda_b{ _lambda_b }, cb{
41             _cb }, C{ _C }, sigma_a0{ _sigma_a0 }, sigma_am{
42                 _sigma_am }, phi{ _phi }, x{ _x }, n{ _n }{}
43
44     virtual double operator()(double t) {
45         double esquerda = pow(pow(sigma_a0 - sigma_am, 1 - n) - C *
46             (1 - n) * (t - 2 * phi * x / U), n / (1 - n));
47         double dir_sup = exp(x*(lambda_b+lambda_s)) * (1 - U*(

```

```

        lambda_s+lambda_b)*cb/ esquerda);
33     double dir_inf = pow(1-lambda_s*U*cb/ esquerda, 1+lambda_b/
        lambda_s);
34     return esquerda * (1 - dir_sup / dir_inf);
35 }
36 private:
37     double U, lambda_s, lambda_b, cb, C, sigma_a0, sigma_am, phi, x
        , n;
38 };
39 #endif

```

Apresenta-se na listagem 6.10 o arquivo de implementação da classe MetodoSimpson.hpp.

Listing 6.10: Arquivo de implementação da classe MetodoSimpson.hpp

```

1 #ifndef METODOTRAPEZIO_HPP_
2 #define METODOTRAPEZIO_HPP_
3
4 #include "metodointegracaonumerica.hpp"
5 #include <iostream>
6
7 class MetodoSimpson : public MetodoIntegracaoNumerica1D {
8 public:
9     // construtor
10     MetodoSimpson(Funcao1x1& f_x_in) : MetodoIntegracaoNumerica1D(
        f_x_in) {}
11
12     // a função abaixo resolve a integral pelo método de trapezio
        ... a função é usada aqui dentro
13     virtual double Integrar(double a, double b, size_t numPontos);
14 };
15 #endif

```

Apresenta-se na listagem 6.11 o arquivo de implementação da classe MetodoSimpson.cpp.

Listing 6.11: Arquivo de implementação da classe MetodoSimpson.cpp

```

1 #include "metodosimpson.hpp"
2
3 double MetodoSimpson::Integrar(double a, double b, size_t numPontos
    ) {
4     // se numPontos for par, somo 1
5     if (numPontos % 2 == 0) {
6         std::cout << "\nWarning: numPontos passado eh par, somando
            1 para integrar pelo metodo de Simpson!" << std::endl;

```

```

7         numPontos += 1;
8     }
9
10    double h = (b-a)/(numPontos-1);
11
12    // método de Simpson
13    double dx = (b - a) / (numPontos - 1);
14
15    double resultado = (funcao(a) + funcao(b)) * h / 3;
16
17    for (size_t i = 1; i < numPontos - 1; i++) {
18        if (i % 2 == 0) // se i for par, multiplico por 2
19            resultado += funcao(a + i * dx) * 2 * h / 3;
20        else // se i for impar, multiplico por 4
21            resultado += funcao(a + i * dx) * 4 * h / 3;
22    }
23
24    return resultado;
25}

```

Apresenta-se na listagem 6.12 o arquivo de implementação da classe MetodoIntegracaoNumericalD.hpp.

Listing 6.12: Arquivo de implementação da classe metodointegracaonumerica.hpp

```

1 #ifndef METODOINTEGRACAONUMERICA_HPP_
2 #define METODOINTEGRACAONUMERICA_HPP_
3
4 #include <cstdlib>
5
6 #include "funcao.hpp"
7
8 class MetodoIntegracaoNumericalD {
9 public:
10     MetodoIntegracaoNumericalD(Funcao1x1& fx_in) : funcao(fx_in) {}
11     virtual double Integrar(double a, double b, size_t numPontos) =
12         0; // Virtual pura
13
14     Funcao1x1& funcao; // Funcao 1D a ser integrada
15 };
16 #endif

```

Apresenta-se na listagem 6.13 o arquivo de implementação da classe CGnuplot.hpp.

Listing 6.13: Arquivo de implementação da classe CGnuplot.cpp

```

1 #ifndef CGNUPLOT_HPP
2 #define CGNUPLOT_HPP
3
4 #include <vector>
5 #include <string>
6 #include <iostream>
7 #include <stdio.h>
8 #include <stdlib.h>
9
10 #ifdef _WIN32
11 #define GNUPLOT_NAME "C:\\Program\\"_ "\\Files\\gnuplot\\bin\\gnuplot_
    -p"
12 #else
13 #define GNUPLOT_NAME "gnuplot"
14 #endif
15
16 class CGnuplot {
17 public:
18     CGnuplot() {}
19
20     static void plot(std::string name, std::string xlabel, std
        ::string ylabel, std::string saveName);
21     static void semilogx(std::string name, std::string xlabel,
        std::string ylabel, std::string saveName);
22     static void semilogy(std::string name, std::string xlabel,
        std::string ylabel, std::string saveName);
23 };
24 #endif

```

Apresenta-se na listagem 6.14 o arquivo de implementação da classe CGnuplot.cpp.

Listing 6.14: Arquivo de implementação da classe CGnuplot.cpp

```

1 #include "CGnuplot.hpp"
2
3 using namespace std;
4
5 void CGnuplot::plot(string name, string xlabel, string ylabel,
    string saveName) {
6 #ifdef _WIN32
7     FILE* pipe = _popen(GNUPLOT_NAME, "w");
8 #else
9     FILE* pipe = popen(GNUPLOT_NAME, "w");
10 #endif

```

```

11         fprintf(pipe, ("set_xlabel_" + xlabel + "'\n").c_str());
12         fprintf(pipe, ("set_ylabel_" + ylabel + "'\n").c_str());
13         fprintf(pipe, "unset_key\n");
14         fprintf(pipe, ("plot_" + name + "'_with_linespoints_"
15             linestyle_1\n").c_str());
16         fprintf(pipe, "set_term_pngcairo\n");
17         fprintf(pipe, ("set_output_" + saveName + "'\n").c_str());
18         fprintf(pipe, "replot\n");
19         fprintf(pipe, "set_term_win\n");
20         fflush(pipe);
21     }
22 void CGnuplot::semilogy(string name, string xlabel, string ylabel,
23     string saveName) {
24     #ifdef _WIN32
25         FILE* pipe = _popen(GNUPLOT_NAME, "w");
26     #else
27         FILE* pipe = popen(GNUPLOT_NAME, "w");
28     #endif
29     fprintf(pipe, ("set_xlabel_" + xlabel + "'\n").c_str());
30     fprintf(pipe, ("set_ylabel_" + ylabel + "'\n").c_str());
31     fprintf(pipe, ("set_logscale_y\n"));
32     fprintf(pipe, "unset_key\n");
33     fprintf(pipe, ("plot_" + name + "'_with_linespoints_"
34         linestyle_1\n").c_str());
35     fprintf(pipe, "set_term_pngcairo\n");
36     fprintf(pipe, ("set_output_" + saveName + "'\n").c_str());
37     fprintf(pipe, "replot\n");
38     fprintf(pipe, "set_term_win\n");
39     fflush(pipe);
40 }
41 void CGnuplot::semilogx(string name, string xlabel, string ylabel,
42     string saveName) {
43     #ifdef _WIN32
44         FILE* pipe = _popen(GNUPLOT_NAME, "w");
45     #else
46         FILE* pipe = popen(GNUPLOT_NAME, "w");
47     #endif
48     fprintf(pipe, ("set_xlabel_" + xlabel + "'\n").c_str());
49     fprintf(pipe, ("set_ylabel_" + ylabel + "'\n").c_str());
50     fprintf(pipe, ("set_logscale_x\n"));

```

```

49     fprintf(pipe, "unset_key\n");
50     fprintf(pipe, ("plot_" + name + "'_with_linespoints_
        linestyle_1\n").c_str());
51     fprintf(pipe, "set_term_pngcairo\n");
52     fprintf(pipe, ("set_output_" + saveName + "'\n").c_str());
53     fprintf(pipe, "replot\n");
54     fprintf(pipe, "set_term_win\n");
55     fflush(pipe);
56 }

```

Apresenta-se na listagem 6.15 o arquivo de implementação da classe main.cpp.

Listing 6.15: Arquivo de implementação da função main().

```

1 #include <iostream>
2 #include <string>
3
4 #include "CSimuladorParticulas.cpp"
5
6 using namespace std;
7
8 int main() {
9
10     //////////////////////////////////////
11     CSimuladorParticulas simulacao("particulaFluido.txt", "
        rocha.txt");
12     simulacao.printCSimuladorParticulas();
13     simulacao.run();
14     //////////////////////////////////////
15     /*delete x_par_ptr;
16     delete y_par_ptr;
17     delete objSeno2Dptr;
18     delete metodo;*/
19 }

```

Capítulo 7

Teste

Todo projeto de engenharia passa por uma etapa de testes. Neste capítulo apresentamos alguns testes do software desenvolvido. Estes testes devem dar resposta aos diagramas de caso de uso inicialmente apresentados (diagramas de caso de uso geral e específicos).

Posteriormente, os resultados gerados serão comparados com os resultados obtidos de experimento laboratoriais a fim de validar a modelagem matemática desenvolvida.

7.1 Teste 1: Entrada de dados, grid do tempo e cálculo do tb.

Nessa etapa serão testadas as entradas de dados via arquivo .txt, a criação do grid do tempo e do espaço e do cálculo do tempo em que incia a formação de pontes(tb) . O teste será executado utilizando valores da literatura. Os resultados obtidos das equações da modelagem matemática serão comparados com os resultados calculados manualmente.

Posteriormente, a modelagem será validada a partir da conferência dos resultados obtidos e dos gráficos gerados com os resultados obtidos e dos gráficos gerados em laboratório.

Nesse primeiro teste, os parâmetros devem ser mostrados no terminal (Figura 7.2) e devem ser os mesmo que estão no arquivo ParticulaFluido.txt e Rocha.txt (Figura 7.1) . Foi testada a criação da malha do tempo e do espaço bem como o valor do cálculo do tb comparando com os valores calculados manualmente. Pode-se observar que os valores obtidos na Figura 7.2 concorda com os valores dos parâmetros inseridos conforme Figura 7.1. Além disso, a malha do tempo foi criada de forma correta e o tempo tb também, conforme figura 7.3.

A seguir apresente texto explicando a sequência do teste e imagens do programa (captura de tela).

coloque aqui texto falando do diagrama de pacotes, referencie a figura. Veja Figura 7.1.

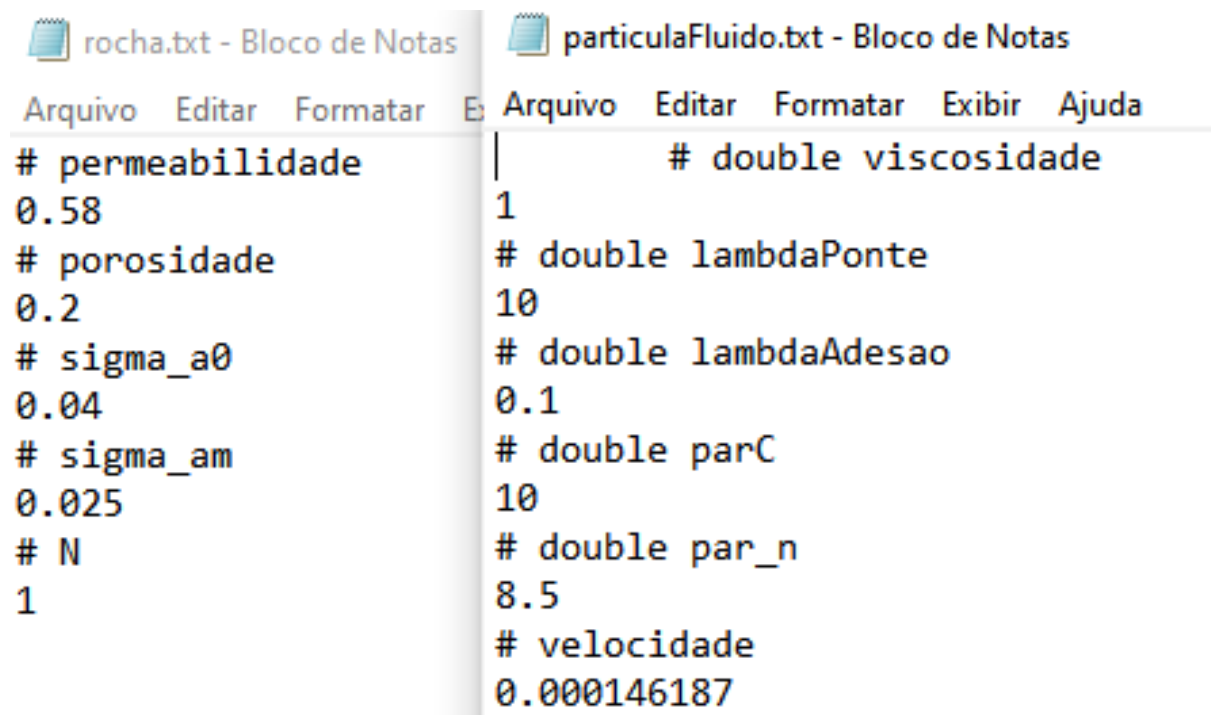


Figura 7.1: Valores dos parâmetros inseridos no arquivo .txt.

```
-----
Valores de ParticulaFluido
-----
viscosidade: 1
lambdaPonte: 10
lambdaAdesao: 0.1
parC: 10
par_n: 8.5
velocidade: 0.000146187

-----
Valores de CRocha
-----
permeabilidade: 0.58
porosidade: 0.2
sigma_a0: 0.04
sigma_am: 0.025
N: 1

-----
Classe da simulacao:
-----
Grid dos tempos:
0 - 0.1 - 0.2 - 0.3 - 0.4 - 0.5 - 0.6 - 0.7 - 0.8 - 0.9 - 1 - 1.1 - 1.2 - 1.3 - 1.4 - 1.5 - 1.6 - 1.7 - 1.8 - 1.9 - 2 - 2.1 - 2.2 - 2.3 - 2.4 - 2.5 - 2.6 - 2.7 - 2.8 - 2.9 - 3 - 3.1 - 3.2 - 3.3 - 3.4 - 3.5 - 3.6 - 3.7 - 3.8 - 3.9 - 4 - 4.1 - 4.2 - 4.3 - 4.4 - 4.5 - 4.6 - 4.7 - 4.8 - 4.9 - 5 - 5.1 - 5.2 - 5.3 - 5.4 - 5.5 - 5.6 - 5.7 - 5.8 - 5.9 - 6 - 6.1 - 6.2 - 6.3 - 6.4 - 6.5 - 6.6 - 6.7 - 6.8 - 6.9 - 7 - 7.1 - 7.2 - 7.3 - 7.4 - 7.5 - 7.6 - 7.7 - 7.8 - 7.9 - 8 - 8.1 - 8.2 - 8.3 - 8.4 - 8.5 - 8.6 - 8.7 - 8.8 - 8.9 - 9 - 9.1 - 9.2 - 9.3 - 9.4 - 9.5 - 9.6 - 9.7 - 9.8 - 9.9 - 10
Vetor de tb:
-inf - 0.691482 - 0.897558 - 1.07487 - 1.24039 - 1.39947 - 1.55446 - 1.70664 - 1.85675 - 2.00529 - 2.15259 - 2.30017 - 2.44782 - 2.59547 - 2.74312 - 2.89077 - 3.03842 - 3.18607 - 3.33372 - 3.48137 - 3.62902 - 3.77667 - 3.92432 - 4.07197 - 4.21962 - 4.36727 - 4.51492 - 4.66257 - 4.81022 - 4.95787 - 5.10552 - 5.25317 - 5.40082 - 5.54847 - 5.69612 - 5.84377 - 5.99142 - 6.13907 - 6.28672 - 6.43437 - 6.58202 - 6.72967 - 6.87732 - 7.02497 - 7.17262 - 7.32027 - 7.46792 - 7.61557 - 7.76322 - 7.91087 - 8.05852 - 8.20617 - 8.35382 - 8.50147 - 8.64912 - 8.79677 - 8.94442 - 9.09207 - 9.23972 - 9.38737 - 9.53502 - 9.68267 - 9.83032 - 9.97797 - 10.12562 - 10.27327 - 10.42092 - 10.56857 - 10.71622 - 10.86387 - 11.01152 - 11.15917 - 11.30682 - 11.45447 - 11.60212 - 11.74977 - 11.89742 - 12.04507 - 12.19272 - 12.34037 - 12.48802 - 12.63567 - 12.78332 - 12.93097 - 13.07862 - 13.22627 - 13.37392 - 13.52157 - 13.66922 - 13.81687 - 13.96452 - 14.11217 - 14.25982 - 14.40747 - 14.55512 - 14.70277 - 14.85042 - 14.99807 - 15.14572 - 15.29337 - 15.44102 - 15.58867 - 15.73632 - 15.88397 - 16.03162 - 16.17927 - 16.32692 - 16.47457 - 16.62222 - 16.76987 - 16.91752 - 17.06517 - 17.21282 - 17.36047 - 17.50812 - 17.65577 - 17.80342 - 17.95107 - 18.09872 - 18.24637 - 18.39402 - 18.54167 - 18.68932 - 18.83697 - 18.98462 - 19.13227 - 19.27992 - 19.42757 - 19.57522 - 19.72287 - 19.87052 - 20.01817 - 20.16582 - 20.31347 - 20.46112 - 20.60877 - 20.75642 - 20.90407 - 21.05172 - 21.19937 - 21.34702 - 21.49467 - 21.64232 - 21.78997 - 21.93762 - 22.08527 - 22.23292 - 22.38057 - 22.52822 - 22.67587 - 22.82352 - 22.97117 - 23.11882 - 23.26647 - 23.41412 - 23.56177 - 23.70942 - 23.85707 - 24.00472 - 24.15237 - 24.29992 - 24.44757 - 24.59522 - 24.74287 - 24.89052 - 25.03817 - 25.18582 - 25.33347 - 25.48112 - 25.62877 - 25.77642 - 25.92407 - 26.07172 - 26.21937 - 26.36702 - 26.51467 - 26.66232 - 26.80997 - 26.95762 - 27.10527 - 27.25292 - 27.40057 - 27.54822 - 27.69587 - 27.84352 - 27.99117 - 28.13882 - 28.28647 - 28.43412 - 28.58177 - 28.72942 - 28.87707 - 29.02472 - 29.17237 - 29.31992 - 29.46757 - 29.61522 - 29.76287 - 29.91052 - 30.05817 - 30.20582 - 30.35347 - 30.50112 - 30.64877 - 30.79642 - 30.94407 - 31.09172 - 31.23937 - 31.38702 - 31.53467 - 31.68232 - 31.82997 - 31.97762 - 32.12527 - 32.27292 - 32.42057 - 32.56822 - 32.71587 - 32.86352 - 33.01117 - 33.15882 - 33.30647 - 33.45412 - 33.60177 - 33.74942 - 33.89707 - 34.04472 - 34.19237 - 34.33992 - 34.48757 - 34.63522 - 34.78287 - 34.93052 - 35.07817 - 35.22582 - 35.37347 - 35.52112 - 35.66877 - 35.81642 - 35.96407 - 36.11172 - 36.25937 - 36.40702 - 36.55467 - 36.70232 - 36.84997 - 36.99762 - 37.14527 - 37.29292 - 37.44057 - 37.58822 - 37.73587 - 37.88352 - 38.03117 - 38.17882 - 38.32647 - 38.47412 - 38.62177 - 38.76942 - 38.91707 - 39.06472 - 39.21237 - 39.35992 - 39.50757 - 39.65522 - 39.80287 - 39.95052 - 40.09817 - 40.24582 - 40.39347 - 40.54112 - 40.68877 - 40.83642 - 40.98407 - 41.13172 - 41.27937 - 41.42702 - 41.57467 - 41.72232 - 41.86997 - 42.01762 - 42.16527 - 42.31292 - 42.46057 - 42.60822 - 42.75587 - 42.90352 - 43.05117 - 43.19882 - 43.34647 - 43.49412 - 43.64177 - 43.78942 - 43.93707 - 44.08472 - 44.23237 - 44.37992 - 44.52757 - 44.67522 - 44.82287 - 44.97052 - 45.11817 - 45.26582 - 45.41347 - 45.56112 - 45.70877 - 45.85642 - 46.00407 - 46.15172 - 46.29937 - 46.44702 - 46.59467 - 46.74232 - 46.88997 - 47.03762 - 47.18527 - 47.33292 - 47.48057 - 47.62822 - 47.77587 - 47.92352 - 48.07117 - 48.21882 - 48.36647 - 48.51412 - 48.66177 - 48.80942 - 48.95707 - 49.10472 - 49.25237 - 49.39992 - 49.54757 - 49.69522 - 49.84287 - 49.99052 - 50.13817 - 50.28582 - 50.43347 - 50.58112 - 50.72877 - 50.87642 - 51.02407 - 51.17172 - 51.31937 - 51.46702 - 51.61467 - 51.76232 - 51.90997 - 52.05762 - 52.20527 - 52.35292 - 52.50057 - 52.64822 - 52.79587 - 52.94352 - 53.09117 - 53.23882 - 53.38647 - 53.53412 - 53.68177 - 53.82942 - 53.97707 - 54.12472 - 54.27237 - 54.41992 - 54.56757 - 54.71522 - 54.86287 - 55.01052 - 55.15817 - 55.30582 - 55.45347 - 55.60112 - 55.74877 - 55.89642 - 56.04407 - 56.19172 - 56.33937 - 56.48702 - 56.63467 - 56.78232 - 56.92997 - 57.07762 - 57.22527 - 57.37292 - 57.52057 - 57.66822 - 57.81587 - 57.96352 - 58.11117 - 58.25882 - 58.40647 - 58.55412 - 58.70177 - 58.84942 - 58.99707 - 59.14472 - 59.29237 - 59.43992 - 59.58757 - 59.73522 - 59.88287 - 60.03052 - 60.17817 - 60.32582 - 60.47347 - 60.62112 - 60.76877 - 60.91642 - 61.06407 - 61.21172 - 61.35937 - 61.50702 - 61.65467 - 61.80232 - 61.94997 - 62.09762 - 62.24527 - 62.39292 - 62.54057 - 62.68822 - 62.83587 - 62.98352 - 63.13117 - 63.27882 - 63.42647 - 63.57412 - 63.72177 - 63.86942 - 64.01707 - 64.16472 - 64.31237 - 64.45992 - 64.60757 - 64.75522 - 64.90287 - 65.05052 - 65.19817 - 65.34582 - 65.49347 - 65.64112 - 65.78877 - 65.93642 - 66.08407 - 66.23172 - 66.37937 - 66.52702 - 66.67467 - 66.82232 - 66.96997 - 67.11762 - 67.26527 - 67.41292 - 67.56057 - 67.70822 - 67.85587 - 68.00352 - 68.15117 - 68.29882 - 68.44647 - 68.59412 - 68.74177 - 68.88942 - 69.03707 - 69.18472 - 69.33237 - 69.47992 - 69.62757 - 69.77522 - 69.92287 - 70.07052 - 70.21817 - 70.36582 - 70.51347 - 70.66112 - 70.80877 - 70.95642 - 71.10407 - 71.25172 - 71.39937 - 71.54702 - 71.69467 - 71.84232 - 71.98997 - 72.13762 - 72.28527 - 72.43292 - 72.58057 - 72.72822 - 72.87587 - 73.02352 - 73.17117 - 73.31882 - 73.46647 - 73.61412 - 73.76177 - 73.90942 - 74.05707 - 74.20472 - 74.35237 - 74.49992 - 74.64757 - 74.79522 - 74.94287 - 75.09052 - 75.23817 - 75.38582 - 75.53347 - 75.68112 - 75.82877 - 75.97642 - 76.12407 - 76.27172 - 76.41937 - 76.56702 - 76.71467 - 76.86232 - 77.00997 - 77.15762 - 77.30527 - 77.45292 - 77.60057 - 77.74822 - 77.89587 - 78.04352 - 78.19117 - 78.33882 - 78.48647 - 78.63412 - 78.78177 - 78.92942 - 79.07707 - 79.22472 - 79.37237 - 79.51992 - 79.66757 - 79.81522 - 79.96287 - 80.11052 - 80.25817 - 80.40582 - 80.55347 - 80.70112 - 80.84877 - 80.99642 - 81.14407 - 81.29172 - 81.43937 - 81.58702 - 81.73467 - 81.88232 - 82.02997 - 82.17762 - 82.32527 - 82.47292 - 82.62057 - 82.76822 - 82.91587 - 83.06352 - 83.21117 - 83.35882 - 83.50647 - 83.65412 - 83.80177 - 83.94942 - 84.09707 - 84.24472 - 84.39237 - 84.53992 - 84.68757 - 84.83522 - 84.98287 - 85.13052 - 85.27817 - 85.42582 - 85.57347 - 85.72112 - 85.86877 - 86.01642 - 86.16407 - 86.31172 - 86.45937 - 86.60702 - 86.75467 - 86.90232 - 87.04997 - 87.19762 - 87.34527 - 87.49292 - 87.64057 - 87.78822 - 87.93587 - 88.08352 - 88.23117 - 88.37882 - 88.52647 - 88.67412 - 88.82177 - 88.96942 - 89.11707 - 89.26472 - 89.41237 - 89.55992 - 89.70757 - 89.85522 - 90.00287 - 90.15052 - 90.29817 - 90.44582 - 90.59347 - 90.74112 - 90.88877 - 91.03642 - 91.18407 - 91.33172 - 91.47937 - 91.62702 - 91.77467 - 91.92232 - 92.06997 - 92.21762 - 92.36527 - 92.51292 - 92.66057 - 92.80822 - 92.95587 - 93.10352 - 93.25117 - 93.39882 - 93.54647 - 93.69412 - 93.84177 - 93.98942 - 94.13707 - 94.28472 - 94.43237 - 94.57992 - 94.72757 - 94.87522 - 95.02287 - 95.17052 - 95.31817 - 95.46582 - 95.61347 - 95.76112 - 95.90877 - 96.05642 - 96.20407 - 96.35172 - 96.49937 - 96.64702 - 96.79467 - 96.94232 - 97.08997 - 97.23762 - 97.38527 - 97.53292 - 97.68057 - 97.82822 - 97.97587 - 98.12352 - 98.27117 - 98.41882 - 98.56647 - 98.71412 - 98.86177 - 99.00942 - 99.15707 - 99.30472 - 99.45237 - 99.59992 - 99.74757 - 99.89522 - 100.04287 - 100.19052 - 100.33817 - 100.48582 - 100.63347 - 100.78112 - 100.92877 - 101.07642 - 101.22407 - 101.37172 - 101.51937 - 101.66702 - 101.81467 - 101.96232 - 102.10997 - 102.25762 - 102.40527 - 102.55292 - 102.70057 - 102.84822 - 102.99587 - 103.14352 - 103.29117 - 103.43882 - 103.58647 - 103.73412 - 103.88177 - 104.02942 - 104.17707 - 104.32472 - 104.47237 - 104.61992 - 104.76757 - 104.91522 - 105.06287 - 105.21052 - 105.35817 - 105.50582 - 105.65347 - 105.80112 - 105.94877 - 106.09642 - 106.24407 - 106.39172 - 106.53937 - 106.68702 - 106.83467 - 106.98232 - 107.12997 - 107.27762 - 107.42527 - 107.57292 - 107.72057 - 107.86822 - 108.01587 - 108.16352 - 108.31117 - 108.45882 - 108.60647 - 108.75412 - 108.90177 - 109.04942 - 109.19707 - 109.34472 - 109.49237 - 109.63992 - 109.78757 - 109.93522 - 110.08287 - 110.23052 - 110.37817 - 110.52582 - 110.67347 - 110.82112 - 110.96877 - 111.11642 - 111.26407 - 111.41172 - 111.55937 - 111.70702 - 111.85467 - 112.00232 - 112.14997 - 112.29762 - 112.44527 - 112.59292 - 112.74057 - 112.88822 - 113.03587 - 113.18352 - 113.33117 - 113.47882 - 113.62647 - 113.77412 - 113.92177 - 114.06942 - 114.21707 - 114.36472 - 114.51237 - 114.65992 - 114.80757 - 114.95522 - 115.10287 - 115.25052 - 115.39817 - 115.54582 - 115.69347 - 115.84112 - 115.98877 - 116.13642 - 116.28407 - 116.43172 - 116.57937 - 116.72702 - 116.87467 - 117.02232 - 117.16997 - 117.31762 - 117.46527 - 117.61292 - 117.76057 - 117.90822 - 118.05587 - 118.20352 - 118.35117 - 118.49882 - 118.64647 - 118.79412 - 118.94177 - 119.08942 - 119.23707 - 119.38472 - 119.53237 - 119.67992 - 119.82757 - 119.97522 - 120.12287 - 120.27052 - 120.41817 - 120.56582 - 120.71347 - 120.86112 - 121.00877 - 121.15642 - 121.30407 - 121.45172 - 121.59937 - 121.74702 - 121.89467 - 122.04232 - 122.18997 - 122.33762 - 122.48527 - 122.63292 - 122.78057 - 122.92822 - 123.07587 - 123.22352 - 123.37117 - 123.51882 - 123.66647 - 123.81412 - 123.96177 - 124.10942 - 124.25707 - 124.40472 - 124.55237 - 124.69992 - 124.84757 - 124.99522 - 125.14287 - 125.29052 - 125.43817 - 125.58582 - 125.73347 - 125.88112 - 126.02877 - 126.17642 - 126.32407 - 126.47172 - 126.61937 - 126.76702 - 126.91467 - 127.06232 - 127.20997 - 127.35762 - 127.50527 - 127.65292 - 127.80057 - 127.94822 - 128.09587 - 128.24352 - 128.39117 - 128.53882 - 128.68647 - 128.83412 - 128.98177 - 129.12942 - 129.27707 - 129.42472 - 129.57237 - 129.71992 - 129.86757 - 129.91522 - 130.06287 - 130.21052 - 130.35817 - 130.50582 - 130.65347 - 130.80112 - 130.94877 - 131.09642 - 131.24407 - 131.39172 - 131.53937 - 131.68702 - 131.83467 - 131.98232 - 132.12997 - 132.27762 - 132.42527 - 132.57292 - 132.72057 - 132.86822 - 133.01587 - 133.16352 - 133.31117 - 133.45882 - 133.60647 - 133.75412 - 133.90177 - 134.04942 - 134.19707 - 134.34472 - 134.49237 - 134.63992 - 134.78757 - 134.93522 - 135.08287 - 135.23052 - 135.37817 - 135.52582 - 135.67347 - 135.82112 - 135.96877 - 136.11642 - 136.26407 - 136.41172 - 136.55937 - 136.70702 - 136.85467 - 136.99992 - 137.14757 - 137.29522 - 137.44287 - 137.59052 - 137.73817 - 137.88582 - 138.03347 - 138.18112 - 138.32877 - 138.47642 - 138.62407 - 138.77172 - 138.91937 - 139.06702 - 139.21467 - 139.36232 - 139.50997 - 139.65762 - 139.80527 - 139.95292 - 140.10057 - 140.24822 - 140.39587 - 140.54352 - 140.69117 - 140.83882 - 140.98647 - 141.13412 - 141.28177 - 141.42942 - 141.57707 - 141.72472 - 141.87237 - 142.01992 - 142.16757 - 142.31522 - 142.46287 - 142.61052 - 142.75817 - 142.90582 - 143.05347 - 143.20112 - 143.34877 - 143.49642 - 143.64407 - 143.79172 - 143.93937 - 144.08702 - 144.23467 - 144.38232 - 144.52997 - 144.67762 - 144.82527 - 144.97292 - 145.12057 - 145.26822 - 145.41587 - 145.56352 - 145.71117 - 145.85882 - 146.00647 - 146.15412 - 146.30177 - 146.44942 - 146.59707 - 146.74472 - 146.89237 - 147.03992 - 147.18757 - 147.33522 - 147.48287 - 147.63052 - 147.77817 - 147.92582 - 148.07347 - 148.22112 - 148.36877 - 148.51642 - 148.66407 - 148.81172 - 148.95937 - 149.10702 - 149.25467 - 149.40232 - 149.54997 - 149.69762 - 149.84527 - 149.99292 - 150.14057 - 150.28822 - 150.43587 - 150.58352 - 150.73117 - 150.87882 - 151.02647 - 151.17412 - 151.32177 - 151.46942 - 151.61707 - 151.76472 - 151.91237 - 152.05992 - 152.20757 - 152.35522 - 152.50287 - 152.65052 - 152.79817 - 152.94582 - 153.09347 - 153.24112 - 153.38877 - 153.53642 - 153.68407 - 153.83172 - 153.97937 - 154.12702 - 154.27467 - 154.42232 - 154.56997 - 154.71762 - 154.86527 - 155.01292 - 155.16057 - 155.30822 - 155.45587 - 155.60352 - 155.75117 - 155.89882 - 156.04647 - 156.19412 - 156.34177 - 156.48942 - 156.63707 - 156.78472 - 156.93237 - 157.07992 - 157.22757 - 157.37522 - 157.52287 - 157.67052 - 157.81817 - 157.96582 - 158.11347 - 158.26112 - 158.40877 - 158.55642 - 158.70407 - 158.85172 - 158.99937 - 159.14702 - 159.29467 - 159.44232 - 159.58997 - 159.73762 - 159.88527 - 160.03292 - 160.18057 - 160.32822 - 160.47587 - 160.62352 - 160.77117 - 160.91882 - 161.06647 - 161.21412 - 161.36177 - 161.50942 - 161.65707 - 161.80472 - 161.95237 - 162.10002 - 162.24767 - 162.39532 - 162.54297 - 162.69062 - 162.83827 - 162.
```

PARÂMETROS			
fi =	0,2		
q =	10	ml/min =>	
D =	1,5	in =>	
U =	0,000146187	m /s	
%Clay =	5	%	
sigmaa_0 =	0,04	m3 / m3 rocha	
sigmaM =	0,025	m3 / m3 rocha	
C =	10	1/s	
lambda_s =	0,1	1/m	
L =	1	m	
n_pontos	10000		
Dx =	0,0001		
cb =	0,0004		
n	1		
lambda_ponte =	0,2		
x	Linha Zonas (s)		Tb(s) n=1
0,0001	0,136811019		0,691531441
0,0002	0,273622039		0,897656678
0,0003	0,410433058		1,075013708
0,0004	0,547244077		1,240592435
0,0005	0,684055097		1,399717309
0,0006	0,820866116		1,554759984

Figura 7.3: Tabela criada no Excel para cálculo dos valores do tempo no início da formação de pontes(tb).

7.2 Teste 2: Gráficos

Nessa etapa os gráficos serão testados e posteriormente serão validados por meio da comparação com os gráficos obtidos a partir dos resultados de laboratório.

Nas figuras 7.4 e 7.5 e nas figuras 7.8 e 7.9 são apresentados os parâmetros que foram utilizados para gerar os gráficos para o primeiro teste e para o segundo respectivamente. É gerado o gráfico da queda de pressão adimensional pelo tempo (Figura 7.6 e Figura 7.10) e da concentração de partículas retidas por exclusão por tamanho pelo espaço (x) (Figura 7.7 e 7.11). Vários testes foram realizados alterando apenas os parâmetros do tempo e do espaço para análise. Pode-se observar que houve um aumento da queda da pressão e um aumento da concentração de partículas, conforme deveria ocorrer. Desse modo, os gráficos foram gerados corretamente, no entanto é necessário a validação com os valores obtidos em experimentos laboratoriais para validação da modelagem.

7.3 Teste 3: Integração numérica

O método de integração numérica pelo método de Simpson será testado utilizando equações de integrais simples e conhecidas com solução analítica para conferência e validação do método numérico para resolução da equação da integral da modelagem matemática. O programa será validado com base nos valores obtidos, onde devem ser iguais aos valores obtidos manualmente.

No Excel foi criada uma tabela (Figura 7.12) para cálculo da integral de $t * \sin(t)$ para conferência com os valores obtidos por meio do método de integração numérica (método de Simpson). No programa, foi alterada a equação para a equação $t * \sin(t)$ a fim de obter os valores e gráficos obtidos através dessa Integral. Os parâmetros e a malha do tempo e do espaço utilizada apresenta-se na Figura 7.13. O resultado da integral calculado pelo Excel e o resultado obtido pelo programa (Figura 7.14) foram exatamente iguais, confirmando que o método numérico está executando de forma correta a integração das equações. A equação da concentração de partículas retidas por exclusão por tamanho ou formação de pontes foi substituída pelo cálculo da Integral, e foi possível perceber que o gráfico (Figura 7.15) gerado concorda com o gráfico da função.

```

-----
Valores de ParticulaFluido
-----
viscosidade: 1
lambdaPonte: 10
lambdaAdesao: 0.1
parC: 10
par_n: 8.5
velocidade: 0.000146187

-----
Valores de CRocha
-----
permeabilidade: 0.58
porosidade: 0.2
sigma_a0: 0.04
sigma_am: 0.025
N: 1

-----
Classe da simulacao:
-----
Grid dos tempos:
0 - 0.1 - 0.2 - 0.3 - 0.4 - 0.5 - 0.6 - 0.7 - 0.8 - 0.9 - 1 - 1.1 - 1.2 - 1.3 - 1.4
2.9 - 3 - 3.1 - 3.2 - 3.3 - 3.4 - 3.5 - 3.6 - 3.7 - 3.8 - 3.9 - 4 - 4.1 - 4.2 - 4.3
5.8 - 5.9 - 6 - 6.1 - 6.2 - 6.3 - 6.4 - 6.5 - 6.6 - 6.7 - 6.8 - 6.9 - 7 - 7.1 - 7.2
8.7 - 8.8 - 8.9 - 9 - 9.1 - 9.2 - 9.3 - 9.4 - 9.5 - 9.6 - 9.7 - 9.8 - 9.9 - 10
Vetor de tb:
-inf - 0.691482 - 0.897558 - 1.07487 - 1.24039 - 1.39947 - 1.55446 - 1.70664 - 1.856
3.16298 - 3.30546 - 3.44763 - 3.58952 - 3.73116 - 3.87257 - 4.01378 - 4.15479 - 4.2
- 5.55724 - 5.6969 - 5.83648 - 5.97598 - 6.11541 - 6.25477 - 6.39406 - 6.53329 - 6.
- 7.92273 - 8.06143 - 8.2001 - 8.33873 - 8.47733 - 8.61589 - 8.75442 - 8.89292 - 9.
- 10.2764 - 10.4147 - 10.5529 - 10.691 - 10.8292 - 10.9673 - 11.1055 - 11.2436 - 11.
- 12.6237 - 12.7616 - 12.8996 - 13.0375 - 13.1754 - 13.3132 - 13.4511 - 13.589 - 13
1 - 14.9669 - 15.1046 - 15.2424 - 15.3801 - 15.5178 - 15.6555 - 15.7932 - 15.9309 -
97 - 17.3073 - 17.4449 - 17.5825 - 17.72 - 17.8576 - 17.9952 - 18.1328 - 18.2703 - 1
1 - 19.6456 - 19.7831 - 19.9206 - 20.058 - 20.1955 - 20.333 - 20.4705 - 20.6079 - 20

```

Figura 7.4: Tela do programa mostrando os parâmetros, a malha do tempo e o tb.

```

#ifndef CSIMULADORPARTICULAS_HPP
#define CSIMULADORPARTICULAS_HPP

#include "CGrid.cpp"
#include "CRocha.cpp"
#include "funcao.hpp"
#include "CGnuplot.cpp"
#include "metodosimpson.cpp"
#include "CParticulaFluido.cpp"

#include<vector>
#include<string>
#include<iostream>

class CSimuladorParticulas : public CParticulaFluido, CRocha {
private:
    size_t indiceTempoAtual = 0;
    size_t size_tempo = 101;
    size_t size_malha = 1001;
    size_t numPontosIntegral = 1001;
    double start_x = 0.0, end_x = 0.10, start_t = 0, end_t = 10.0;

    std::vector<double> tempo;

```

Figura 7.5: Tela do programa mostrando os parâmetros do tempo, do espaço, e o número de pontos para integração.

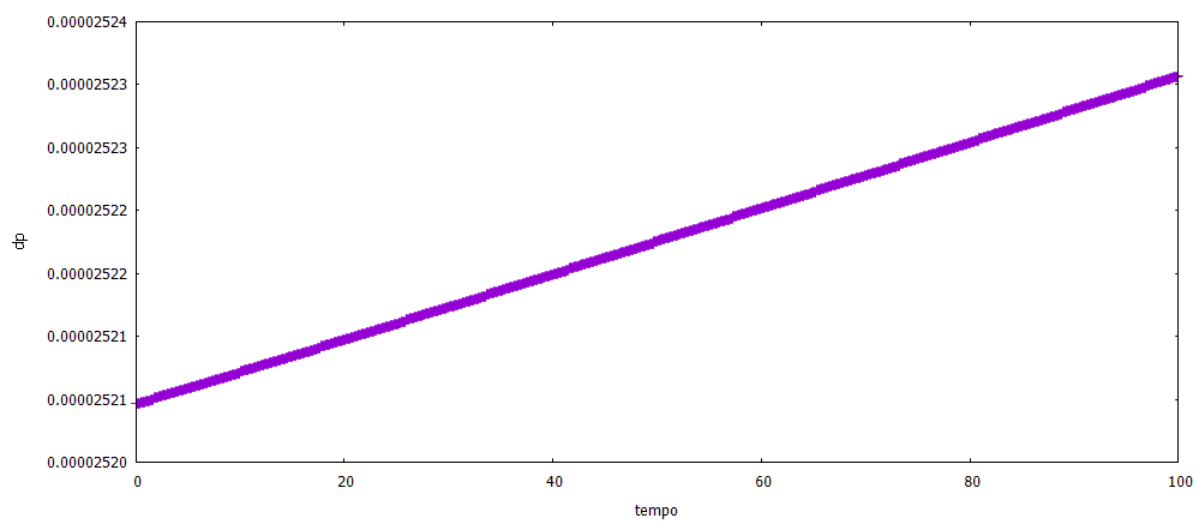
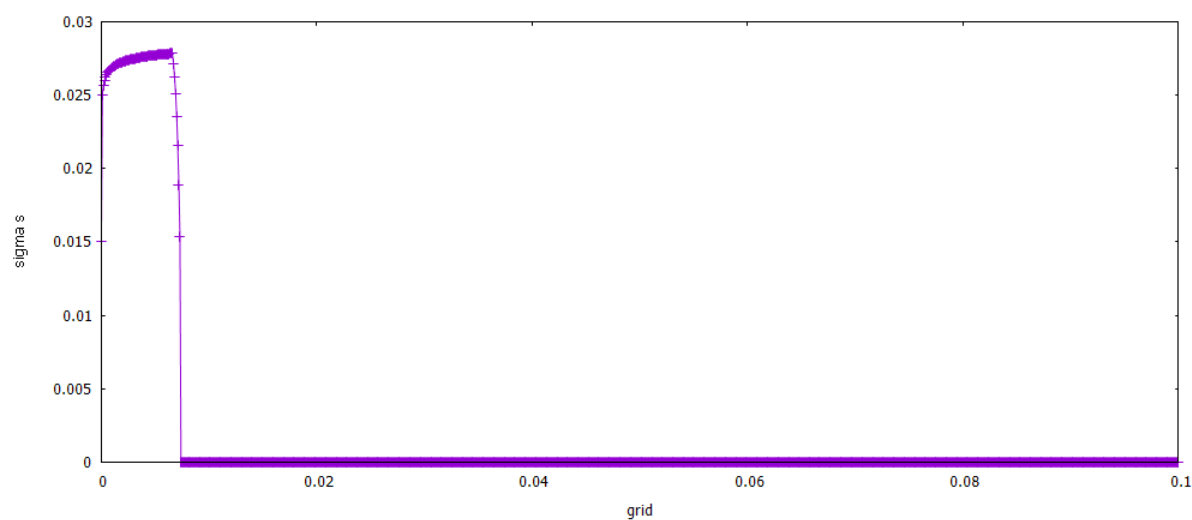


Figura 7.6: Gráfico gerado da queda de pressão pelo tempo.

Figura 7.7: Gráfico do Σ_s pelo espaço.

```

-----
Valores de ParticulaFluido
-----
viscosidade: 1
lambdaPonte: 10
lambdaAdesao: 0.1
parC: 10
par_n: 8.5
velocidade: 0.000146187

-----
Valores de CRocha
-----
permeabilidade: 0.58
porosidade: 0.2
sigma_a0: 0.04
sigma_am: 0.025
N: 1

-----
Classe da simulacao:
-----
Grid dos tempos:
0 - 2 - 4 - 6 - 8 - 10 - 12 - 14 - 16 - 18 - 20 - 22 - 24 - 26 - 28 - 30 - 32 -
- 70 - 72 - 74 - 76 - 78 - 80 - 82 - 84 - 86 - 88 - 90 - 92 - 94 - 96 - 98 - 100 -
30 - 132 - 134 - 136 - 138 - 140 - 142 - 144 - 146 - 148 - 150 - 152 - 154 - 156 -
86 - 188 - 190 - 192 - 194 - 196 - 198 - 200
Vetor de tb:
-inf - 0.691482 - 0.897558 - 1.07487 - 1.24039 - 1.39947 - 1.55446 - 1.70664 -
3.16298 - 3.30546 - 3.44763 - 3.58952 - 3.73116 - 3.87257 - 4.01378 - 4.15479
- 5.55724 - 5.6969 - 5.83648 - 5.97598 - 6.11541 - 6.25477 - 6.39406 - 6.53325
- 7.92273 - 8.06143 - 8.2001 - 8.33873 - 8.47733 - 8.61589 - 8.75442 - 8.89295
- 10.2764 - 10.4147 - 10.5529 - 10.691 - 10.8292 - 10.9673 - 11.1055 - 11.2436
- 12.6237 - 12.7616 - 12.8996 - 13.0375 - 13.1754 - 13.3132 - 13.4511 - 13.5892

```

Figura 7.8: Tela do programa mostrando os parâmetros, a malha do tempo e o tb.

```

#include "CGrid.cpp"
#include "CRocha.cpp"
#include "funcao.hpp"
#include "CGnuplot.cpp"
#include "metodosimpson.cpp"
#include "CParticulaFluido.cpp"

#include<vector>
#include<string>
#include<iostream>

class CSimuladorParticulas : public CParticulaFluido, CRocha {
private:
    size_t indiceTempoAtual = 0;
    size_t size_tempo = 101;
    size_t size_malha = 1001;
    size_t numPontosIntegral = 1001;
    double start_x = 0.0, end_x = 0.10, start_t = 0, end_t = 200.0;

    std::vector<double> tempo;
    std::vector<double> malha;

    std::vector<CGrid*> resultados_ao_longo_do_tempo;

public:
    .....

```

Figura 7.9: Tela do programa mostrando os parâmetros do tempo, do espaço, e o número de pontos para integração

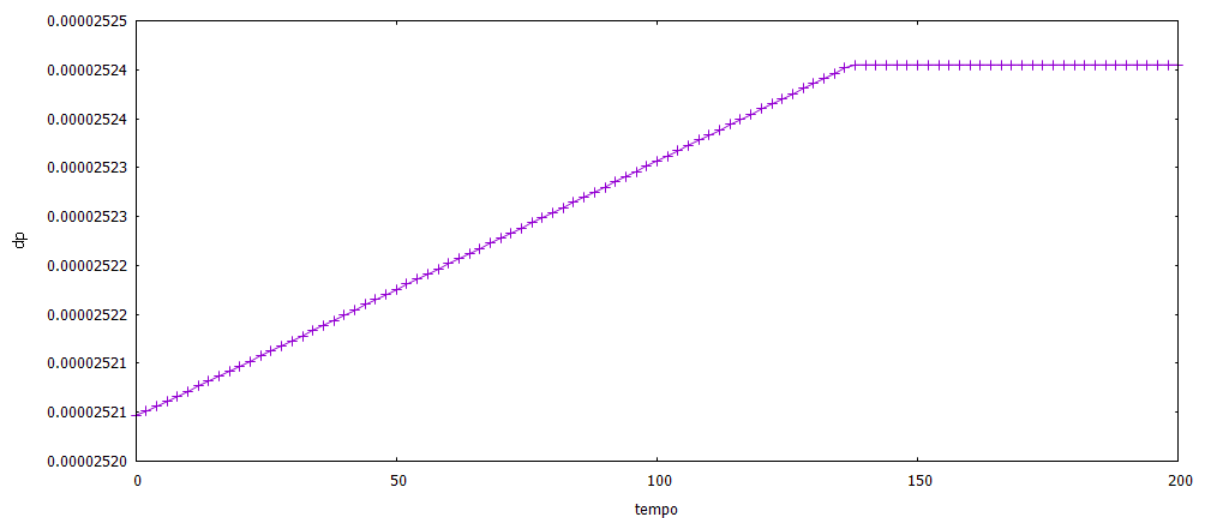


Figura 7.10: Gráfico gerado da queda de pressão pelo tempo.

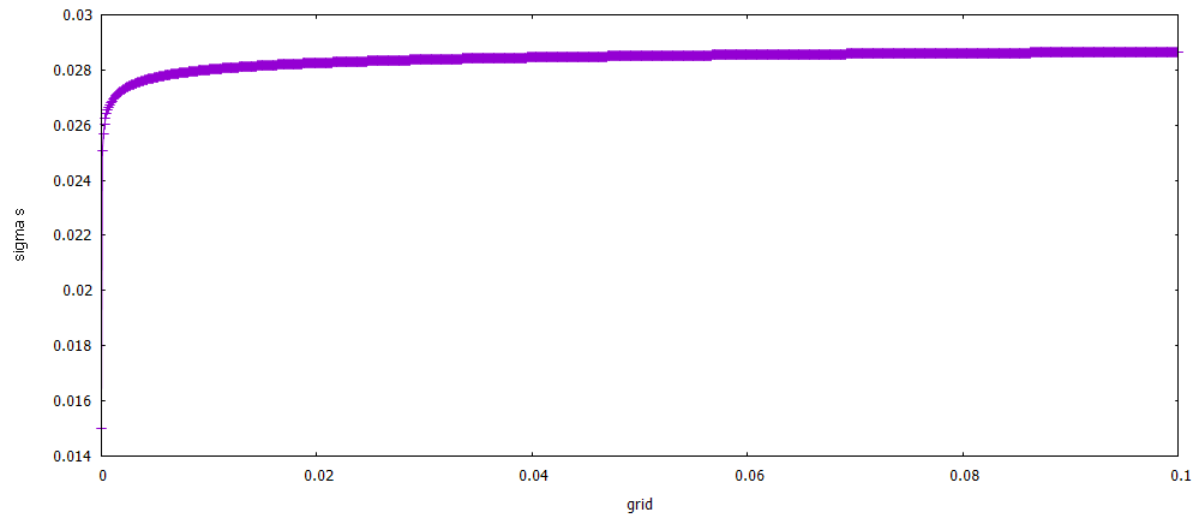


Figura 7.11: Gráfico do Sigma_s pelo espaço.

$$\int t * \sin x = -t \cos t + \sin t + C$$

Cálculo da integral de "t* SIN(t) " de t= 0,1368 a t=0,5 para conferência do método de simpsons						
considerando t=0,5	t	x	cos t	sin t	cos t- 0,13	sin t (0,13)
t> fi*x/U	0,5	0,0001	0,877583	0,479426	0,990655961	0,136384631
t<tb=0,69	0,13681102					
-t*cos t + sin t	0,04063426					
-t*cos t + sin t	0,00085198					
resultado da integral	0,03978228					
calculo da parte const de sigma_s	0,12574095	(sem multiplicar pela integral)				
calculo do sigma S	0,01902433					
resultado no programa	0,0227449					

Figura 7.12: Tabela gerada no Excel da resolução da Integral de t*sin(x).

```

-----
viscosidade: 1
lambdaPonte: 10
lambdaAdesao: 0.1
parC: 10
par_n: 8.5
velocidade: 0.000146187

-----
Valores de CRocha
-----
permeabilidade: 0.58
porosidade: 0.2
sigma_a0: 0.04
sigma_am: 0.025
N: 1

-----
Classe da simulacao:
-----
Grid dos tempos:
0 - 0.1 - 0.2 - 0.3 - 0.4 - 0.5 - 0.6 - 0.7 - 0.8 - 0.9 - 1 - 1.1 - 1.2 - 1.3 - 1.4 - 1.5 - 1.6 - 1.7 - 1.8 - 1.9 - 2 -
2.1 - 2.2 - 2.3 - 2.4 - 2.5 - 2.6 - 2.7 - 2.8 - 2.9 - 3 - 3.1 - 3.2 - 3.3 - 3.4 - 3.5 - 3.6 - 3.7 - 3.8 - 3.9 - 4 - 4.1 -
4.2 - 4.3 - 4.4 - 4.5 - 4.6 - 4.7 - 4.8 - 4.9 - 5 - 5.1 - 5.2 - 5.3 - 5.4 - 5.5 - 5.6 - 5.7 - 5.8 - 5.9 - 6 - 6.1 - 6 -
6.2 - 6.3 - 6.4 - 6.5 - 6.6 - 6.7 - 6.8 - 6.9 - 7 - 7.1 - 7.2 - 7.3 - 7.4 - 7.5 - 7.6 - 7.7 - 7.8 - 7.9 - 8 - 8.1 - 8.2 -
8.3 - 8.4 - 8.5 - 8.6 - 8.7 - 8.8 - 8.9 - 9 - 9.1 - 9.2 - 9.3 - 9.4 - 9.5 - 9.6 - 9.7 - 9.8 - 9.9 - 10 -
Vetor de tb:
-inf - 0.691482 - 0.897558 - 1.07487 - 1.24039 - 1.39947 - 1.55446 - 1.70664 - 1.85675 - 2.00529 - 2.15259 - 2.29888 -
4.4435 - 2.58911 - 2.73328 - 2.87694 - 3.02016 - 3.16298 - 3.30546 - 3.44763 - 3.58952 - 3.73116 - 3.87257 - 4.01378 -
4.15498 - 4.29561 - 4.43593 - 4.57596 - 4.71575 - 4.85529 - 4.99458 - 5.13363 - 5.27245 - 5.41106 - 5.54946 - 5.68765 -
5.82563 - 5.96339 - 6.10093 - 6.23825 - 6.37536 - 6.51226 - 6.64895 - 6.78543 - 6.92171 - 7.05779 - 7.19367 - 7.32935 - 7.46483 - 7.60011 - 7.73529 - 7.87037 - 8.00535 - 8.14013 - 8.27471 - 8.40909 - 8.54327 - 8.67725 - 8.81103 - 8.94461 - 9.07799 - 9.21117 - 9.34415 - 9.47693 - 9.60951 - 9.74189 - 9.87407 - 10.00605 - 10.13783 - 10.26941 - 10.40079 - 10.53197 - 10.66295 - 10.79373 - 10.92431 - 11.05469 - 11.18487 - 11.31485 - 11.44463 - 11.57431 - 11.70379 - 11.83307 - 11.96215 - 12.09103 - 12.21971 - 12.34819 - 12.47647 - 12.60455 - 12.73243 - 12.86011 - 12.98759 - 13.11487 - 13.24195 - 13.36883 - 13.49551 - 13.62199 - 13.74827 - 13.87435 - 14.00023 - 14.12591 - 14.25139 - 14.37667 - 14.50175 - 14.62663 - 14.75131 - 14.87579 - 15.00007 - 15.12415 - 15.24803 - 15.37171 - 15.49519 - 15.61847 - 15.74155 - 15.86443 - 15.98711 - 16.10959 - 16.23187 - 16.35395 - 16.47583 - 16.59751 - 16.71899 - 16.84027 - 16.96135 - 17.08223 - 17.20291 - 17.32339 - 17.44367 - 17.56375 - 17.68363 - 17.80331 - 17.92279 - 18.04207 - 18.16115 - 18.28003 - 18.39871 - 18.51719 - 18.63547 - 18.75355 - 18.87143 - 18.98911 - 19.10659 - 19.22387 - 19.34095 - 19.45783 - 19.57451 - 19.69099 - 19.80727 - 19.92335 - 20.03923 - 20.15491 - 20.27039 - 20.38567 - 20.50075 - 20.61563 - 20.73031 - 20.84479 - 20.95907 - 21.07315 - 21.18703 - 21.30071 - 21.41419 - 21.52737 - 21.64025 - 21.75283 - 21.86511 - 21.97719 - 22.08907 - 22.20075 - 22.31223 - 22.42351 - 22.53459 - 22.64547 - 22.75615 - 22.86663 - 22.97691 - 23.08699 - 23.19677 - 23.30625 - 23.41543 - 23.52431 - 23.63299 - 23.74137 - 23.84945 - 23.95723 - 24.06471 - 24.17189 - 24.27877 - 24.38535 - 24.49163 - 24.59761 - 24.70329 - 24.80867 - 24.91375 - 25.01853 - 25.12301 - 25.22719 - 25.33107 - 25.43465 - 25.53793 - 25.64091 - 25.74359 - 25.84597 - 25.94805 - 26.04983 - 26.15131 - 26.25249 - 26.35337 - 26.45395 - 26.55423 - 26.65421 - 26.75389 - 26.85327 - 26.95235 - 27.05113 - 27.14961 - 27.24779 - 27.34567 - 27.44325 - 27.54053 - 27.63751 - 27.73419 - 27.83057 - 27.92665 - 28.02243 - 28.11791 - 28.21309 - 28.30787 - 28.40235 - 28.49653 - 28.59041 - 28.68399 - 28.77727 - 28.87025 - 28.96293 - 29.05531 - 29.14739 - 29.23917 - 29.33065 - 29.42183 - 29.51271 - 29.60329 - 29.69357 - 29.78355 - 29.87323 - 29.96261 - 30.05169 - 30.14047 - 30.22895 - 30.31713 - 30.40501 - 30.49259 - 30.57987 - 30.66685 - 30.75353 - 30.83991 - 30.92599 - 31.01177 - 31.09725 - 31.18243 - 31.26731 - 31.35189 - 31.43617 - 31.52015 - 31.60383 - 31.68721 - 31.77029 - 31.85307 - 31.93555 - 32.01773 - 32.09961 - 32.18119 - 32.26247 - 32.34345 - 32.42413 - 32.50451 - 32.58459 - 32.66437 - 32.74385 - 32.82303 - 32.90191 - 32.98049 - 33.05877 - 33.13675 - 33.21443 - 33.29181 - 33.36889 - 33.44567 - 33.52215 - 33.59833 - 33.67421 - 33.74979 - 33.82507 - 33.90005 - 33.97473 - 34.04911 - 34.12319 - 34.19697 - 34.27045 - 34.34363 - 34.41651 - 34.48909 - 34.56137 - 34.63335 - 34.70503 - 34.77641 - 34.84749 - 34.91827 - 34.98875 - 35.05893 - 35.12881 - 35.19839 - 35.26767 - 35.33665 - 35.40533 - 35.47371 - 35.54179 - 35.60957 - 35.67705 - 35.74423 - 35.81111 - 35.87769 - 35.94397 - 36.00995 - 36.07563 - 36.14101 - 36.20609 - 36.27087 - 36.33535 - 36.39953 - 36.46341 - 36.527 - 36.59029 - 36.65327 - 36.71595 - 36.77833 - 36.84041 - 36.90219 - 36.96367 - 37.02485 - 37.08573 - 37.14631 - 37.20659 - 37.26657 - 37.32625 - 37.38563 - 37.44471 - 37.50349 - 37.56187 - 37.61985 - 37.67743 - 37.73461 - 37.79149 - 37.84807 - 37.90425 - 37.95993 - 38.01521 - 38.07019 - 38.12487 - 38.17925 - 38.23333 - 38.28711 - 38.34059 - 38.39377 - 38.44665 - 38.49923 - 38.55151 - 38.60349 - 38.65517 - 38.70665 - 38.75783 - 38.80871 - 38.85929 - 38.90957 - 38.95955 - 39.00923 - 39.05861 - 39.10769 - 39.15647 - 39.20495 - 39.25313 - 39.30101 - 39.34859 - 39.39587 - 39.44285 - 39.48953 - 39.53591 - 39.58199 - 39.62777 - 39.67325 - 39.71843 - 39.76331 - 39.80789 - 39.85217 - 39.89615 - 39.93983 - 39.98321 - 40.02629 - 40.06907 - 40.11155 - 40.15373 - 40.19561 - 40.23719 - 40.27847 - 40.31945 - 40.36013 - 40.40051 - 40.44059 - 40.48037 - 40.51985 - 40.55903 - 40.59791 - 40.63649 - 40.67477 - 40.71275 - 40.75043 - 40.78781 - 40.82489 - 40.86157 - 40.89795 - 40.93403 - 40.96981 - 41.00529 - 41.04047 - 41.07535 - 41.10993 - 41.14421 - 41.17819 - 41.21187 - 41.24525 - 41.27833 - 41.31111 - 41.34359 - 41.37577 - 41.40765 - 41.43923 - 41.47051 - 41.50149 - 41.53217 - 41.56255 - 41.59263 - 41.62241 - 41.65189 - 41.68107 - 41.71 - 41.73879 - 41.76727 - 41.79545 - 41.82333 - 41.85091 - 41.87819 - 41.90517 - 41.93185 - 41.95823 - 41.98431 - 42.01009 - 42.03557 - 42.06075 - 42.08563 - 42.11021 - 42.13449 - 42.15847 - 42.18215 - 42.20553 - 42.22861 - 42.25139 - 42.27387 - 42.29605 - 42.31793 - 42.33951 - 42.36079 - 42.38177 - 42.40245 - 42.42283 - 42.44291 - 42.46269 - 42.48217 - 42.50135 - 42.52023 - 42.53881 - 42.55709 - 42.57507 - 42.59275 - 42.61013 - 42.62721 - 42.64409 - 42.66067 - 42.67695 - 42.69293 - 42.70861 - 42.72409 - 42.73927 - 42.75415 - 42.76873 - 42.78301 - 42.79699 - 42.81067 - 42.82405 - 42.83713 - 42.85 - 42.86259 - 42.87487 - 42.88685 - 42.89853 - 42.90991 - 42.92099 - 42.93177 - 42.94225 - 42.95243 - 42.96231 - 42.97189 - 42.98117 - 42.99015 - 43.0 - 43.00879 - 43.01717 - 43.02525 - 43.03303 - 43.04051 - 43.04769 - 43.05457 - 43.06115 - 43.06743 - 43.07341 - 43.07909 - 43.08447 - 43.08955 - 43.09433 - 43.09881 - 43.103 - 43.10689 - 43.11037 - 43.11355 - 43.11643 - 43.11901 - 43.12129 - 43.12327 - 43.12495 - 43.12633 - 43.12741 - 43.12819 - 43.12867 - 43.12885 - 43.12873 - 43.12831 - 43.12759 - 43.12657 - 43.12525 - 43.12363 - 43.12171 - 43.11949 - 43.11697 - 43.11415 - 43.11103 - 43.10761 - 43.10389 - 43.1 - 43.09587 - 43.09135 - 43.08643 - 43.08111 - 43.07549 - 43.06957 - 43.06335 - 43.05683 - 43.05001 - 43.04289 - 43.03547 - 43.02775 - 43.01973 - 43.01141 - 43.00279 - 42.99387 - 42.98465 - 42.97513 - 42.96531 - 42.95519 - 42.94477 - 42.93405 - 42.92303 - 42.91171 - 42.89999 - 42.88787 - 42.87535 - 42.86253 - 42.84941 - 42.83599 - 42.82227 - 42.80825 - 42.79393 - 42.77931 - 42.76439 - 42.74917 - 42.73365 - 42.71783 - 42.70171 - 42.68529 - 42.66857 - 42.65155 - 42.63423 - 42.61661 - 42.59869 - 42.58047 - 42.56195 - 42.54313 - 42.52401 - 42.50459 - 42.48487 - 42.46485 - 42.44453 - 42.42391 - 42.40299 - 42.38177 - 42.36025 - 42.33843 - 42.31631 - 42.29389 - 42.27117 - 42.24815 - 42.22483 - 42.20121 - 42.17729 - 42.15307 - 42.12855 - 42.10373 - 42.07861 - 42.05319 - 42.02747 - 42.00145 - 41.97513 - 41.94851 - 41.92159 - 41.89437 - 41.86685 - 41.83903 - 41.81091 - 41.78249 - 41.75377 - 41.72475 - 41.69543 - 41.66581 - 41.63589 - 41.60567 - 41.57515 - 41.54433 - 41.51321 - 41.48179 - 41.45007 - 41.41805 - 41.38573 - 41.35311 - 41.32019 - 41.28697 - 41.25345 - 41.21963 - 41.18551 - 41.15109 - 41.11637 - 41.08135 - 41.04603 - 41.01041 - 40.97449 - 40.93827 - 40.90175 - 40.86493 - 40.82781 - 40.79039 - 40.75267 - 40.71465 - 40.67633 - 40.63771 - 40.59879 - 40.55957 - 40.51995 - 40.48003 - 40.43981 - 40.39929 - 40.35847 - 40.31735 - 40.27593 - 40.23421 - 40.19219 - 40.14987 - 40.10725 - 40.06433 - 40.02111 - 39.97759 - 39.93377 - 39.88965 - 39.84523 - 39.80051 - 39.75549 - 39.71017 - 39.66455 - 39.61863 - 39.57241 - 39.52589 - 39.47907 - 39.43195 - 39.38453 - 39.33681 - 39.28879 - 39.24047 - 39.19185 - 39.14293 - 39.09371 - 39.04419 - 38.99437 - 38.94425 - 38.89383 - 38.84311 - 38.79209 - 38.74077 - 38.68915 - 38.63723 - 38.58501 - 38.53249 - 38.47967 - 38.42655 - 38.37313 - 38.31941 - 38.26539 - 38.21107 - 38.15645 - 38.10153 - 38.04631 - 37.99079 - 37.93497 - 37.87885 - 37.82243 - 37.76571 - 37.70869 - 37.65137 - 37.59375 - 37.53583 - 37.47761 - 37.41909 - 37.36027 - 37.30115 - 37.24173 - 37.18201 - 37.12209 - 37.06187 - 37.00135 - 36.94053 - 36.87941 - 36.81799 - 36.75627 - 36.69425 - 36.63193 - 36.56931 - 36.50639 - 36.44317 - 36.37965 - 36.31583 - 36.25171 - 36.18729 - 36.12257 - 36.05755 - 35.99223 - 35.92661 - 35.86069 - 35.79447 - 35.72795 - 35.66113 - 35.59401 - 35.52659 - 35.45887 - 35.39085 - 35.32253 - 35.25391 - 35.18499 - 35.11577 - 35.04625 - 34.97643 - 34.90631 - 34.83589 - 34.76517 - 34.69415 - 34.62283 - 34.55121 - 34.47929 - 34.40707 - 34.33455 - 34.26173 - 34.18861 - 34.11519 - 34.04147 - 33.96745 - 33.89313 - 33.81851 - 33.74359 - 33.66837 - 33.59285 - 33.51703 - 33.44091 - 33.36449 - 33.28777 - 33.21075 - 33.13343 - 33.05581 - 32.97789 - 32.89967 - 32.82115 - 32.74233 - 32.66321 - 32.58379 - 32.50407 - 32.42405 - 32.34373 - 32.26311 - 32.18219 - 32.10097 - 32.01945 - 31.93763 - 31.85551 - 31.77309 - 31.69037 - 31.60735 - 31.52403 - 31.44041 - 31.35649 - 31.27227 - 31.18775 - 31.10293 - 31.01781 - 30.93239 - 30.84667 - 30.76065 - 30.67433 - 30.58771 - 30.50079 - 30.41357 - 30.32605 - 30.23823 - 30.15011 - 30.06169 - 29.97297 - 29.88395 - 29.79463 - 29.70501 - 29.61509 - 29.52487 - 29.43435 - 29.34353 - 29.25241 - 29.16109 - 29.06947 - 28.97755 - 28.88533 - 28.79281 - 28.70009 - 28.60717 - 28.51395 - 28.42043 - 28.32661 - 28.23249 - 28.13807 - 28.04335 - 27.94833 - 27.85301 - 27.75739 - 27.66147 - 27.56525 - 27.46873 - 27.37191 - 27.27479 - 27.17737 - 27.07965 - 26.98163 - 26.88331 - 26.78469 - 26.68577 - 26.58655 - 26.48703 - 26.38721 - 26.28709 - 26.18657 - 26.08565 - 25.98433 - 25.88261 - 25.78049 - 25.67797 - 25.57505 - 25.47173 - 25.36801 - 25.26389 - 25.15937 - 25.05445 - 24.94913 - 24.84341 - 24.73729 - 24.63077 - 24.52385 - 24.41653 - 24.30881 - 24.20069 - 24.09217 - 23.98325 - 23.87393 - 23.76421 - 23.65409 - 23.54357 - 23.43265 - 23.32133 - 23.20961 - 23.09749 - 22.98487 - 22.87185 - 22.75843 - 22.64461 - 22.53039 - 22.41577 - 22.30075 - 22.18533 - 22.06951 - 21.95329 - 21.83667 - 21.71965 - 21.60223 - 21.48441 - 21.36619 - 21.24757 - 21.12855 - 21.00913 - 20.88931 - 20.76909 - 20.64847 - 20.52745 - 20.40603 - 20.28421 - 20.16199 - 20.03937 - 19.91635 - 19.79293 - 19.66911 - 19.54489 - 19.42027 - 19.29525 - 19.16983 - 19.04401 - 18.91779 - 18.79117 - 18.66415 - 18.53673 - 18.40891 - 18.28069 - 18.15207 - 18.02305 - 17.89363 - 17.76381 - 17.63359 - 17.50297 - 17.37195 - 17.24053 - 17.10871 - 16.97649 - 16.84387 - 16.71085 - 16.57743 - 16.44361 - 16.30939 - 16.17477 - 16.03975 - 15.90433 - 15.76851 - 15.63229 - 15.49567 - 15.35865 - 15.22123 - 15.08341 - 14.94519 - 14.80657 - 14.66755 - 14.52813 - 14.38831 - 14.24809 - 14.10747 - 13.96645 - 13.82503 - 13.68321 - 13.54099 - 13.39837 - 13.25535 - 13.11193 - 12.96811 - 12.82389 - 12.67927 - 12.53425 - 12.38883 - 12.24301 - 12.09679 - 11.95017 - 11.80315 - 11.65573 - 11.50791 - 11.35969 - 11.21107 - 11.06205 - 10.91263 - 10.76281 - 10.61259 - 10.46197 - 10.31095 - 10.15953 - 10.00771 - 9.85549 - 9.70287 - 9.54985 - 9.39643 - 9.24261 - 9.08839 - 8.93377 - 8.77875 - 8.62333 - 8.46751 - 8.31129 - 8.15467 - 8.0 - 7.84265 - 7.68433 - 7.52561 - 7.36649 - 7.20697 - 7.04705 - 6.88673 - 6.72601 - 6.56489 - 6.40337 - 6.24145 - 6.07913 - 5.91641 - 5.75329 - 5.58977 - 5.42585 - 5.26153 - 5.09681 - 4.93169 - 4.76617 - 4.60025 - 4.43393 - 4.26721 - 4.10019 - 3.93287 - 3.76515 - 3.59703 - 3.42851 - 3.25959 - 3.09027 - 2.92055 - 2.75043 - 2.58 - 2.40957 - 2.23885 - 2.06823 - 1.89771 - 1.72729 - 1.55697 - 1.38675 - 1.21663 - 1.04661 - 0.87669 - 0.70687 - 0.53715 - 0.36753 - 0.19801 - 0.02859 - 0.14017 - 0.29085 - 0.48153 - 0.71221 - 0.98289 - 1.29357 - 1.64425 - 1.93493 - 2.16561 - 2.42629 - 2.71697 - 2.93765 - 3.08833 - 3.16901 - 3.18069 - 3.12337 - 3.00805 - 2.8347
```



*malha_0.500000.txt - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

malha	sigma_a	-	sigma_s	-	diff_sigma_a
0	0.0251011		0		-0.00101069
0.0001	0.025397		0.0397823		-0.00396992
0.0002	0.0265594		0.0338566		-0.0155935
0.0003	0.031125		0.0179736		-0.0612502
0.0004	0.04		0		0
0.0005	0.04		0		0
0.0006	0.04		0		0
0.0007	0.04		0		0
0.0008	0.04		0		0
0.0009	0.04		0		0
0.001	0.04		0		0

Figura 7.14: Valores das concentrações calculadas no tempo = 0,5s em vários pontos(x).

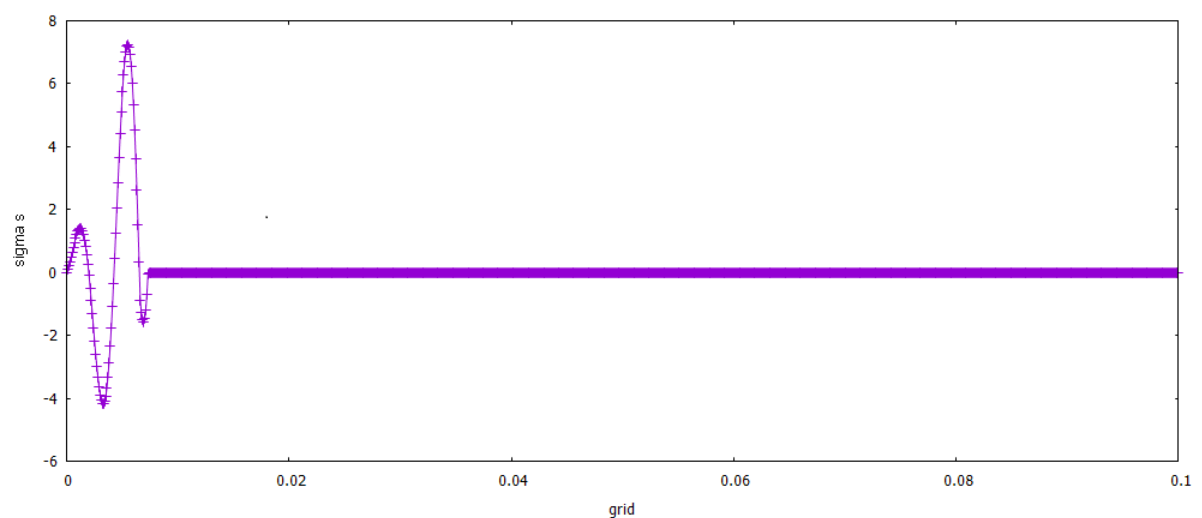


Figura 7.15: Gráfico mostrando o valor de Sigma_s pelo espaço(x)

Capítulo 8

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.

8.1 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.

8.1.1 Como instalar o software

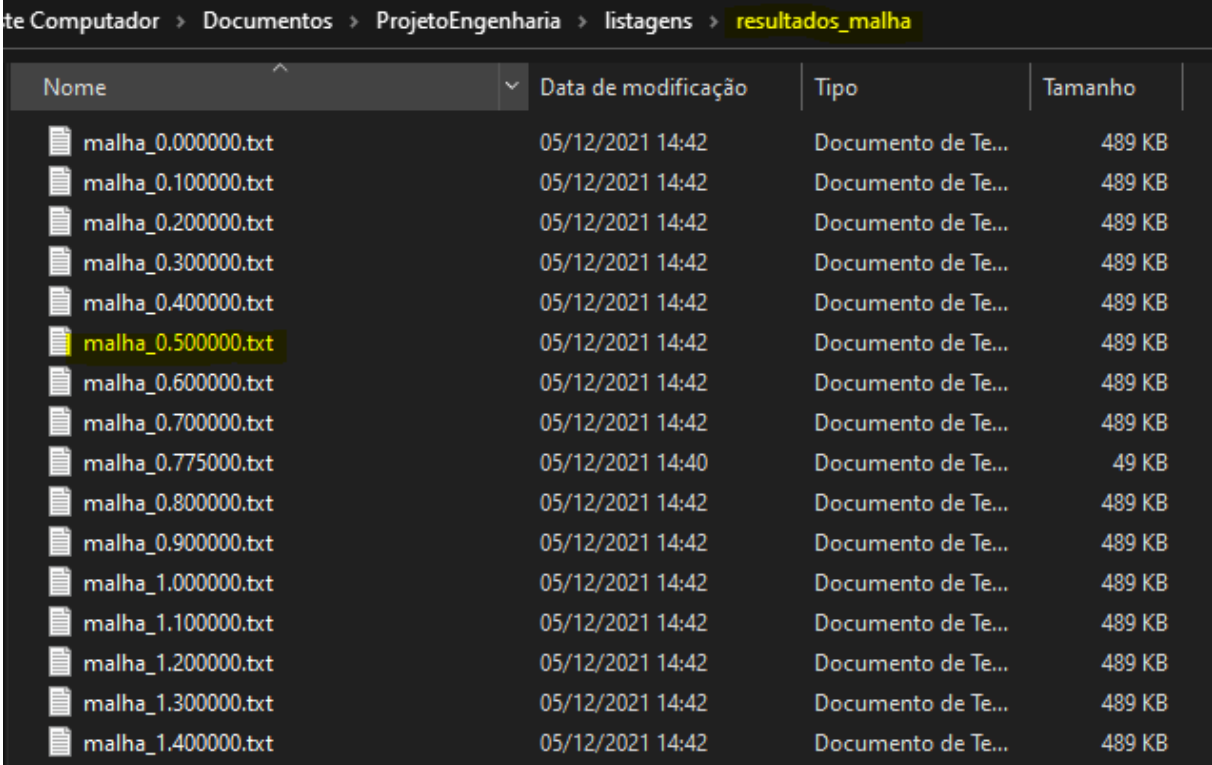
Para instalar o software execute o seguinte passo:

- 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.

8.1.2 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 8.1: Tela do programa mostrando xxx

Veja no Capítulo 7 - Teste, exemplos de uso do software.

8.2 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.

8.2.1 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.

8.2.2 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	CGnuplot.cpp
ficheiro	CGnuplot.hpp [código]
ficheiro	CGrid.cpp
ficheiro	CGrid.hpp [código]
ficheiro	CParticulaFluido.cpp
ficheiro	CParticulaFluido.hpp [código]
ficheiro	CRocha.cpp
ficheiro	CRocha.hpp [código]
ficheiro	CSimuladorParticulas.cpp
ficheiro	CSimuladorParticulas.hpp [código]
ficheiro	funcao.hpp [código]
ficheiro	main.cpp
ficheiro	metodointegracaonumerica.hpp [código]
ficheiro	metodosimpson.cpp
ficheiro	metodosimpson.hpp [código]

Figura 8.2: 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 8.3: 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 8.4: Documentação da classe CSimuladorParticulas.hpp.

Referências Bibliográficas

- [Blaha and Rumbaugh, 2006] Blaha, M. and Rumbaugh, J. (2006). *Modelagem e Projetos Baseados em Objetos com UML 2*. Campus, Rio de Janeiro.
- [B.Plohr et al., 2001] B.Plohr, Marchesin, D., Bedrikovetsky, P., and Krause, P. (2001). Modeling hysteresys in porous media flow via relaxation. *Computacional Geosciences*, 5:225–256.
- [Brakel et al., 1981] Brakel, V. J., Modry, S., and Svata, M. (1981). Mercury porosimetry: State of the art. volume v.29, page p.1. Powder Technology.
- [Bueno, 1994] Bueno, A. D. (1994). Transferência de calor e umidade em telhas: Simulação e análise experimental. Tese de mestrado, Programa de Pós-graduação em Engenharia Civil da Universidade Federal de Santa Catarina, Florianópolis. Nota.
- [Bueno, 1997] Bueno, A. D. (1997). *Apostila de Programa de Orientada a Objeto*. UFSC-EMC-Laboratório de Meios Porosos e Propriedades Termofísicas, Florianópolis.
- [Bueno, 2001a] Bueno, A. D. (2001a). *Estudo Geométrico das Representações Tridimensionais da Estrutura Porosa e Grafos de Conexão Serial para a Determinação de Rochas – Reservatório de Petróleo*. PhD thesis, UFSC.
- [Bueno, 2001b] Bueno, A. D. (2001b). Método de reconstrução da gaussiana truncada 2: Análise das representações obtidas e resultados. *Technical report*, UFSC.
- [Bueno, 2003] Bueno, A. D. (2003). *Programa de Orientada a Objeto com C++ - Aprenda a Programar em Ambiente Multiplataforma com Software Livre*. Novatec, São Paulo.
- [Bueno and Lamberts, 1991] Bueno, A. D. and Lamberts, R. (1991). Proposta de zona de conforto na carta psicométrica. Florianópolis. *Anais do I Encontro Nacional de Normalização e Ligação ao Uso Racional de Energia e ao*
- [Bueno and Philippi, 1996] Bueno, A. D. and Philippi, P. C. (1996). Simulação da transferência de calor e umidade, em telhas expostas ao meio ambiente. *Incompleto*.

- [Bueno et al., 1994] Bueno, A. D., Philippi, P. C., and Lamberts, R. (1994). Influence of water vapour sorption on the thermal behavior of mortar samples. *Proceedings of the Third Internacional Masonry Conference*, pages 2–54.
- [Doe, 2009] Doe, R. (2009). This is a test entry of type @ONLINE.
- [e Patrick W. Daly, 1995] e Patrick W. Daly, H. K. (1995). *A Guide to Latex 2e*. Addison-Wesley, New York, 2 edition.
- [Grossens et al., 1993] Grossens, M., Mittelbach, F., and Samarin, A. (1993). *Latex Companion*. Addison-Wesley, New York.
- [Ioannidis and Chatzis, 1993] Ioannidis, M. and Chatzis, I. (1993). Network modeling of pore structure and transport properties of porous media. *Chem. Eng. Sci*, 48:951–972.
- [Karger, 2004] Karger, A. (2004). *O Tutorial de Lyx*. LyX Team - <http://www.lyx.org>.
- [Knuth, 1986] Knuth, D. E. (1986). *The TeXbook*. Addison-Wesley.
- [Lamport, 1985] Lamport, L. (1985). *Latex - A Document Preparation System*. Addison-Wesley.
- [LyX-Team, 2004a] LyX-Team, editor (2004a). *Extended LyX Features*. LyX Team - <http://www.lyx.org>.
- [LyX-Team, 2004b] LyX-Team, editor (2004b). *The LyX User's Guide*. LyX Team - <http://www.lyx.org>.
- [Mendes, 1997] Mendes, N. (1997). *Modelos Para Previsão da Transferência de Calor e de Umidade em Elementos Porosos de Edificações*. Tese de doutoramento, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis.
- [Moraes and Bueno, 2003] Moraes, F. S. and Bueno, A. D. (2003). *Relatório individual do projeto : Inferências estatísticas aplicadas a problemas de caracterização de LENEP*.
- [Rumbaugh et al., 1994] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1994). *Modelagem e Projetos Baseados em Objetos*. Edit. Campus, Rio de Janeiro.
- [Steding-Jessen, 2000] Steding-Jessen, K. (2000). *Latex demo: Exemplo com Latex 2e*.

Índice Remissivo

A

Análise orientada a objeto, 23

AOO, 23

Associações, 35

atributos, 34

C

colaboração, 29

comunicação, 29

Concepção, 4

Controle, 32

D

Diagrama de colaboração, 29

Diagrama de componentes, 35

Diagrama de execução, 36

Diagrama de máquina de estado, 29

Diagrama de sequência, 26

E

Efeitos do projeto nas associações, 35

Efeitos do projeto nas heranças, 34

Efeitos do projeto nos métodos, 34

Elaboração, 9

especificação, 4

estado, 29

Eventos, 26

H

Heranças, 34

heranças, 34

I

Implementação, 38

M

Mensagens, 26

métodos, 34

modelo, 33

O

otimizações, 35

P

Plataformas, 32

POO, 33

Projeto do sistema, 31

Projeto orientado a objeto, 33

Protocolos, 31

R

Recursos, 32