

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

TFS “*Turtle Flow Simulation*”  
TRABALHO DA DISCIPLINA DESENVOLVIMENTO DE SOFTWARES  
ORIENTADO A OBJETO EM C++

PEDRO HENRIQUE LINHARES  
WAGNER QUEIROZ BARROS

MACAÉ - RJ  
JANEIRO - 2014

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	2
<b>2</b>	<b>Especificação</b>	<b>3</b>
2.1	Especificação do programa . . . . .	3
2.2	Casos de uso do programa . . . . .	3
2.2.1	Caso de uso “Inserir Dados do Problema” . . . . .	4
2.2.2	Caso de uso “Gerar Arquivos de Configuração” . . . . .	5
2.2.3	Caso de uso “Executar Simulação” . . . . .	6
2.2.4	Caso de uso “Plotar Gráficos” . . . . .	7
<b>3</b>	<b>Elaboração</b>	<b>8</b>
3.1	Formulação teórica . . . . .	8
3.1.1	Equação da difusividade . . . . .	8
3.1.2	Discretização da equação da difusividade . . . . .	9
3.2	Identificação de pacotes . . . . .	10
3.3	Análise de domínio . . . . .	11
<b>4</b>	<b>AOO - Análise Orientada a Objeto</b>	<b>14</b>
4.1	Diagramas de classes . . . . .	14
4.1.1	Classes gerais . . . . .	14
4.1.2	Classes de grid . . . . .	16
4.1.3	Classes de modelos de discretização . . . . .	18
4.1.4	Classes de solver . . . . .	18
4.1.5	Classes de interface gráfica . . . . .	19
4.2	Diagrama de sequência - eventos e mensagens . . . . .	21
4.2.1	Diagrama de sequência “Inserir Dados do Problema” . . . . .	21
4.2.2	Diagrama de sequência “Gerar Arquivos de Configuração” . . . . .	22
4.2.3	Diagrama de sequência “Executar Simulação” . . . . .	22
4.3	Diagrama de atividades . . . . .	24

<b>5</b>	<b>Projeto</b>	<b>26</b>
5.1	Projeto do sistema . . . . .	26
5.2	Plataformas suportadas . . . . .	27
5.3	Bibliotecas externas . . . . .	28
5.4	Ambiente de desenvolvimento . . . . .	28
5.5	Requisitos do sistema . . . . .	28
<b>6</b>	<b>Implementação</b>	<b>29</b>
<b>7</b>	<b>Teste</b>	<b>35</b>
7.1	Teste 1: . . . . .	35
7.2	Teste 2: . . . . .	36
<b>8</b>	<b>Documentação</b>	<b>38</b>
8.1	Interface gráfica inicial . . . . .	38
8.2	Configurações iniciais . . . . .	39
8.3	Determinando os valores da simulação . . . . .	39
8.3.1	Guia “Geometry” . . . . .	40
8.3.2	Guia “Rock” . . . . .	40
8.3.3	Guia “Fluid” . . . . .	42
8.3.4	Guia “Wells” . . . . .	43
8.3.5	Guia “Numeric Properties” . . . . .	43
8.3.6	Guia “Initialize Solution” . . . . .	44
8.3.7	Guia “Boundary Condition” . . . . .	45
8.4	Executando a simulação . . . . .	47
8.5	Plotando gráficos . . . . .	50
<b>9</b>	<b>Manual do desenvolvedor</b>	<b>52</b>
9.1	Dependências do software . . . . .	52
9.2	Compilando o software . . . . .	52
9.3	Executando . . . . .	53
<b>A</b>	<b>Arquivos de inicialização da simulação</b>	<b>56</b>
A.1	ArqBlock.in . . . . .	56
A.2	ArqBoundCond1d.in . . . . .	56
A.3	ArqCell1d.in . . . . .	57
A.4	ArqGas.in, ArqOil.in e ArqWater.in . . . . .	57
A.5	ArqInit1d.in . . . . .	58
A.6	ArqNumeric.in . . . . .	58
A.7	ArqRes1d.in . . . . .	58
A.8	ArqWell1d.in . . . . .	58

---

<b>B Sistema de unidades do TFS</b>	<b>60</b>
<b>C Tutoriais</b>	<b>61</b>
C.1 Reservatório horizontal unidimensional homogêneo e fechado . . . . .	61
C.2 Reservatório anticlinal com duas litologias e alimentação de pressão . . . .	72

# Capítulo 1

## Introdução

Segundo [Ertekin et al., 2001] , simulação de reservatórios é a combinação de modelos físicos de engenharia de reservatórios, ferramentas matemáticas, ferramentas numéricas e programação computacional, de forma a se desenvolver uma ferramenta capaz de se prever o desenvolvimento de um reservatório de petróleo, sujeito a diversas condições operacionais.

Pretende-se criar um simulador numérico baseado no modelo unidimensional de um reservatório monofásico, isto é, o escoamento será calculado apenas em uma direção de forma a se prever o comportamento em uma seção linear de um reservatório de hidrocarbonetos. A Figura 1.1 mostra um esquema de uma representação unidimensional de um reservatório de hidrocarbonetos. É possível se identificar os seguintes elementos:

- $q$ , vazão mássica [lbm/s];
- $L$ , comprimento do reservatório [ft];
- $h$ , espessura do reservatório [ft];
- $\Delta y$ , largura do reservatório [ft].

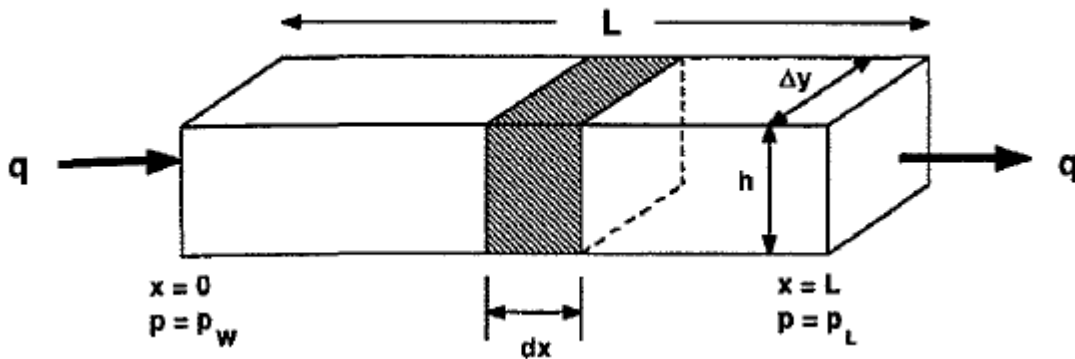


Figura 1.1: Representação unidimensional de um reservatório de hidrocarbonetos

Apesar de se utilizar o modelo unidimensional para representar o reservatório, cada célula do reservatório terá altura, comprimento e espessura variável; sendo possível o estudo de uma seção linear com altura variável.

### 1.1 Objetivos

O objetivo do trabalho é a criação de um simulador unidimensional de reservatórios monofásicos de petróleo. O simulador desenvolvido (TFS - *Turtle Flow Simulator*) possui as seguintes características:

- Interface gráfica amigável com o usuário;
- Desenvolvimento multiplataforma (Windows e GNU/Linux);
- Desacoplamento de códigos entre a interface gráfica e o *kernel* numérico.

O desacoplamento entre a interface gráfica e o *kernel* numérico foi criada de forma a se facilitar a modificação do software, assim é possível se modificar toda a interface gráfica sem a necessidade de modificação do código numérico, ou modificar o *kernel* numérico sem modificar a interface.

Será utilizada a linguagem C++ orientada à objeto com a vantagem de adaptação e expansão futura do código. Para a criação da interface gráfica será utilizada a biblioteca Qt, que possui as vantagens de portabilidade e de ser um software livre.

# Capítulo 2

## Especificação

Apresenta-se neste capítulo a concepção e a especificação do sistema a ser modelado e desenvolvido.

### 2.1 Especificação do programa

O software de simulação de reservatórios TFS deverá ser capaz de simular um reservatório monofásico e unidimensional de petróleo. Será implementado o algoritmo de Gauss-Seidel tridiagonal para solução do sistema de equações lineares.

De forma a facilitar o entendimento do código e a utilização do *software*, os códigos foram implementados e comentados em Inglês.

O *software* deverá possuir uma interface gráfica amigável com o usuário para facilitar o preenchimento dos dados do reservatório. A interface gráfica será construída de forma desacoplada do kernel numérico do programa, dessa forma o kernel numérico poderá ser executado sem a necessidade da interface. Esse desacoplamento entre kernel e interface possui as seguintes vantagens:

1. Independência entre o código da interface e o código numérico;
2. Possibilidade de criar rotinas automatizadas de execução do código numérico;
3. Maior velocidade de execução e economia de memória do sistema.

Durante a criação do código numérico foram criadas classes abstratas visando a expansão do software, aproveitando o conceito de polimorfismo.

### 2.2 Casos de uso do programa

Será criado um software de simulação de reservatórios, capaz de executar as seguintes tarefas básicas:

1. Inserir Dados do Problema (características geométricas, físicas e numéricas do problema);
2. Gerar Arquivos de Configuração para o Kernel Numérico;
3. Executar Simulação (Executar o Kernel Numérico);
4. Plotar Gráficos com Soluções do Problema.

A Figura 2.1 mostra o diagrama geral de casos de uso. Cada uma das tarefas listadas serão detalhadas nos diagramas posteriores.

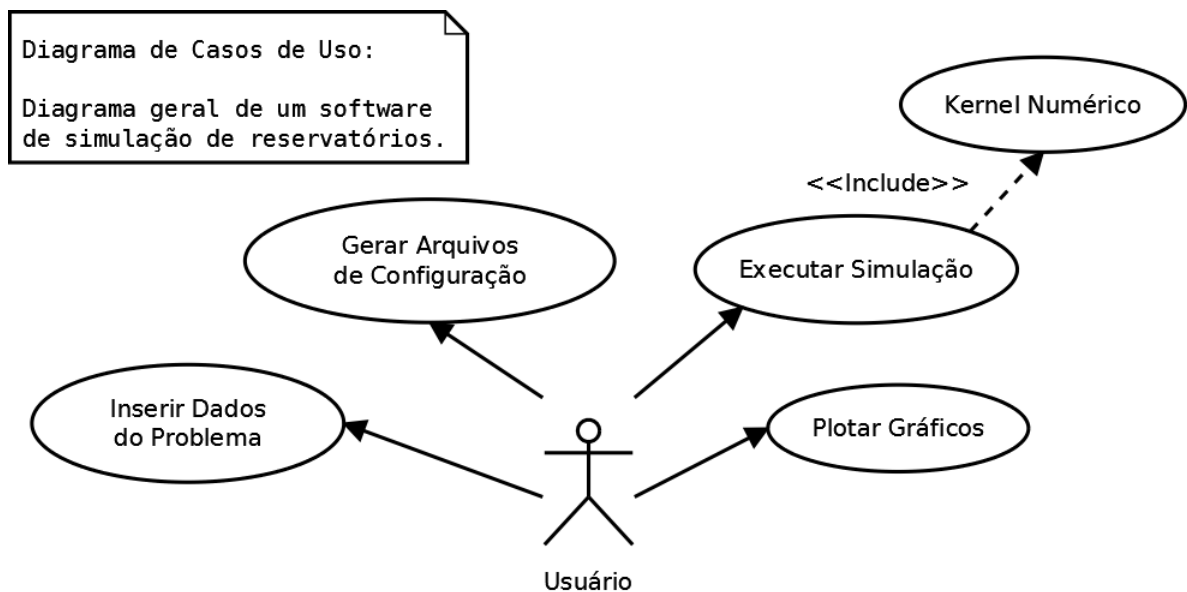


Figura 2.1: Diagrama de casos de uso geral do programa

O software possuirá uma interface gráfica desacoplada, o que significa que o usuário poderá realizar simulações através do preenchimento de arquivos de dados (.in). Ao final do processamento serão criados arquivos de saída (.out) com os resultados da simulação para cada bloco do domínio e contendo os resultados da simulação para cada poço do reservatório. Esses arquivos serão identificados pela sigla OUT no começo do arquivo.

### 2.2.1 Caso de uso “Inserir Dados do Problema”

Em simulação numérica de reservatórios uma das etapas mais importante é a delimitação clara das propriedades físicas e geométricas do problema. Entende-se por propriedades geométricas o formato do reservatório, e por propriedades físicas todos os valores de atributos com significados físicos. O problema possui o seguinte grupo de dados:

1. Propriedades geométricas do reservatório:
  - \* Comprimento (*Length*);
  - \* Largura (*Width*);



- \* Número de blocos (*Block's Number*);
  - \* Espessura de cada bloco (*Thickness*);
  - \* Profundidade do topo de cada bloco (*Depth*).
2. Propriedades físicas da rocha:
- \* Permeabilidade (*Permeability*);
  - \* Porosidade (*Porosity*);
  - \* Pressão de Referência da Porosidade (*Porosity Reference Pressure*);
  - \* Compressibilidade (*Compressibility*).
3. Propriedades físicas dos fluidos:
- \* Pressão PVT (*PVT Pressure*);
  - \* Densidade (*Density*);
  - \* Viscosidade (*Viscosity*);
  - \* Fator volume-formação (FVF).
4. Propriedades de poços:
- \* Localização (*Cell Id*);
  - \* Vazão (*Well Rate*).
5. Propriedades Numéricas:
- \* Tipo de Solver (*Solver Type*);
  - \* Máximo de Iterações do Solver (*Max Solver Iterations*);
  - \* Precisão do Solver (*Solver Precision*);
  - \* Variação de Tempo (*Delta Time*);
  - \* Tempo Final de Simulação (*Final Time Simulation*);
  - \* Máximo de Iterações de Linearização (*Max of Linearization*);
  - \* Precisão da Linearização (*Linearization Precision*).

A Figura 2.2 mostra o diagrama de casos de uso utilizados na entrada das propriedades do problema.

### 2.2.2 Caso de uso “Gerar Arquivos de Configuração”

Após a etapa de entrada de dados na interface gráfica o usuário deverá exportar essas configurações para o *kernel* numérico. Para isso a interface criará arquivos (.in) especificados (Anexo A) que serão carregados pelo *kernel*. A Tabela 2.1 mostra os detalhes desse caso de uso.

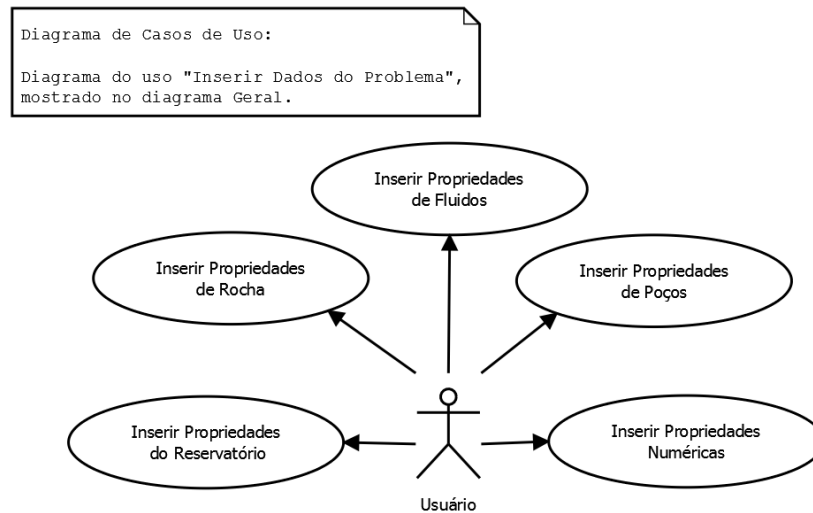


Figura 2.2: Diagrama do caso de uso “inserir dados do problema”

Tabela 2.1: Caso de uso “gerar arquivos de configuração”

Nome do caso de uso:	Gerar Arquivos de Configuração.
Resumo/descrição:	Carregar as propriedades da interface gráfica e salvá-las em arquivos .dat.
Etapas:	1. Carregar propriedades da interface; 2. Salvar propriedades em arquivo.
Cenário alternativo:	Usuário não preencheu todas as propriedades, nesse caso o software imprime na tela que não é possível gerar os arquivos de configuração.

### 2.2.3 Caso de uso “Executar Simulação”

Essa etapa do código será destinada a solução do problema numérico. O *kernel* numérico foi implementado como sendo um software externo à interface, assim, a interface deverá ser capaz de executar o *kernel*. A Tabela 2.2 mostra os detalhes para esse caso de uso.

Tabela 2.2: Caso de uso “executar simulação”

Nome do caso de uso:	Executar Simulação.
Resumo/descrição:	Inicializa o <i>kernel</i> numérico e executa a simulação numérica do problema.
Etapas:	1. Inicializar o <i>kernel</i> . 2. Simular o problema. 3. Criar arquivos .dat com as soluções numéricas do problema.
Cenário alternativo:	Alguma propriedade foi preenchida incorretamente, nesse caso o <i>kernel</i> numérico informa ao usuário o erro ocorrido e congela a tela de execução.

### 2.2.4 Caso de uso “Plotar Gráficos”

Existem vários resultados que podem ser obtidos em uma simulação numérica de reservatórios, assim, é necessário um código para o usuário escolher quais resultados ele deseja visualizar. É possível se plotar as seguintes soluções:

1. Gerar histórico de produção por poço;
2. Gerar curvas de pressão por tempo para cada poço;
3. Gerar curvas de pressão por tempo para uma célula específica;
4. Gerar um perfil da distribuição de pressão no reservatório para um tempo qualquer.

Para se gerar os gráficos será utilizado o software externo gnuplot.

A Figura 2.3 mostra o diagrama de casos de uso para o caso plotar as soluções do problema.

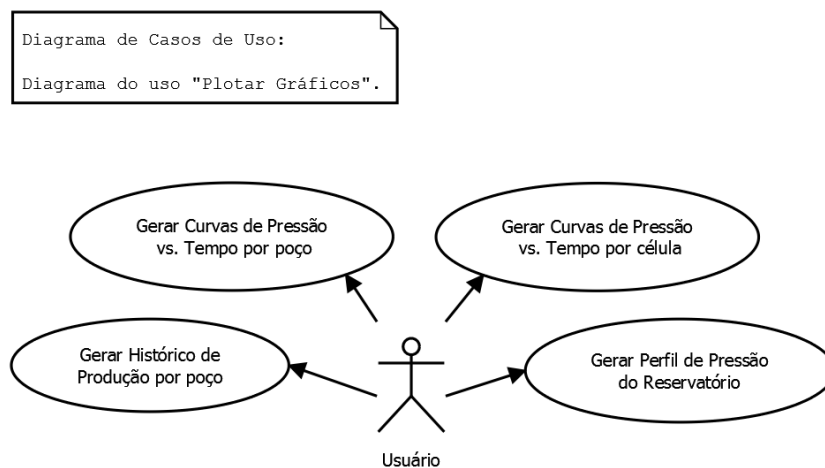


Figura 2.3: Diagrama de casos de uso para o uso “plotar gráficos”

# Capítulo 3

## Elaboração

### 3.1 Formulação teórica

Nesse capítulo será apresentada a formulação teórica, uma análise de pacotes e de domínio que será utilizada no TFS.

#### 3.1.1 Equação da difusividade

Considerando o meio poroso como contínuo, a equação que governa o escoamento de fluidos no reservatório é dado por:

$$\frac{\partial}{\partial t} \{ \phi S_j \rho_j \omega_{ij} + (1 - \phi) \rho_s \omega_{is} \} + \vec{\nabla} \cdot \left( \rho_j \omega_{ij} \vec{v}_j - \phi S_j \overleftrightarrow{D}_{ij} \nabla (\rho_j \omega_{ij}) \right) = \phi S_j r_{ij} + (1 - \phi) r_{is} + R_{iWell} \quad (3.1)$$

essa equação é a forma geral da conservação de espécies químicas em um reservatório. O primeiro termo do lado esquerdo representa a variação da espécie química no volume poroso, o segundo termo é o fluxo convectivo dessa espécie no meio poroso. Os dois primeiros termos do lado direito da equação são a geração da espécie química por reação entre a parte fluida e a rocha, o terceiro termo é a injeção ou produção dessa espécie por um poço de petróleo. A velocidade no meio poroso é calculada segundo a Lei de Darcy, dada por:

$$\vec{v}_j = - \frac{\overleftrightarrow{K} k_{rj}}{\mu_j} (\nabla P_j + \rho_j \vec{g}) \quad (3.2)$$

Combinando as Equações 3.1 e 3.2 encontra-se a equação da difusividade hidráulica composicional:

$$\begin{aligned} & \frac{\partial}{\partial t} \{ \phi S_j \rho_j \omega_{ij} + (1 - \phi) \rho_s \omega_{is} \} + \\ & \vec{\nabla} \cdot \left( -\rho_j \omega_{ij} \frac{\vec{K}_{rj}}{\mu_j} (\nabla P_j + \rho_j \vec{g}) - \phi S_j \overleftarrow{D}_{ij} \nabla (\rho_j \omega_{ij}) \right) = \\ & \phi S_j r_{ij} + (1 - \phi) r_{is} + R_{iWell} \end{aligned} \quad (3.3)$$

A Equação 3.3 será resolvida considerando apenas uma fase no reservatório (modelo monofásico). Esse modelo simplificado será discretizado e resolvido numericamente de forma a prever o comportamento de um reservatório de petróleo.

### 3.1.2 Discretização da equação da difusividade

Considerando o caso monofásico e linear, a forma discretizada da Equação 3.3 é dada por:

$$C_i^{(\nu)(n+1)} P_i^{(\nu+1)(n+1)} + E_i^{(\nu)(n+1)} P_{i+1}^{(\nu+1)(n+1)} + W_i^{(\nu)(n+1)} P_{i-1}^{(\nu+1)(n+1)} = Q_i^{(\nu)(n+1)} \quad (3.4)$$

onde o índice  $n$  indica o passo temporal e o índice  $\nu$  o passo de ajuste não linear. O índice  $i$  indica a discretização espacial do reservatório que é o número de células do problema. Os coeficientes da Equação 3.4 são dados por:

$$E_i^{(\nu)(n+1)} = T_{x(i+\frac{1}{2})}^{(\nu)}$$

$$W_i^{(\nu)(n+1)} = T_{x(i-\frac{1}{2})}^{(\nu)}$$

$$C_i^{(\nu)(n+1)} = -T_{x(i-\frac{1}{2})}^{(\nu)} - T_{x(i+\frac{1}{2})}^{(\nu)} - \frac{\Gamma_i}{\Delta t^n}$$

$$Q_i^{(\nu)(n+1)} = \gamma_{(i+\frac{1}{2})}^{(\nu)} T_{x(i+\frac{1}{2})}^{(\nu)} (Z_{i+1} - Z_i) - \gamma_{(i-\frac{1}{2})}^{(\nu)} T_{x(i-\frac{1}{2})}^{(\nu)} (Z_i - Z_{i-1}) - q_i - \frac{\Gamma_i}{\Delta t^n} P_i^n$$

$\gamma_{(i)}$  é o peso específico do fluido dentro da célula  $i$ . O termo  $T_x$  é conhecido como a transmissibilidade na direção  $x$  e expressa a dificuldade que um fluido tem de escoar entre

dois blocos. Esse termo é dado por:

$$T_x = G_x \left( \frac{1}{\mu B} \right)$$

onde  $G_x$  é o termo geométrico da transmissibilidade, dado por:

$$G_x = \frac{\beta_c k_x A_x}{\Delta x}$$

O termo  $\Gamma$  é expresso como:

$$\Gamma = \frac{V_b}{\alpha_c} \left[ \frac{\phi^0 C_r}{(\nu)} + \frac{\phi^n C_l}{B^0} \right]_{B(n+1)}$$

Quando a Equação 3.4 é aplicada em todas as células do domínio, obtém-se um sistema tridiagonal de equações lineares, dado por:

$$\begin{bmatrix} C_0 & E_0 & 0 & 0 & \cdots & 0 \\ W_1 & C_1 & E_1 & 0 & \cdots & 0 \\ 0 & W_2 & C_2 & E_2 & \cdots & 0 \\ 0 & 0 & W_3 & C_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & W_n & C_n \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_n \end{bmatrix} \quad (3.5)$$

A Equação 3.5 será resolvida para cada passo de iteração linear, e para cada passo de tempo, de forma a obter a solução do problema proposto.

## 3.2 Identificação de pacotes

Com base nas especificações do software, e na formulação físico-matemática do problema, identificou-se que o TFS deverá ser composto por 3 pacotes distintos, o pacote da interface gráfica, o do kernel numérico e o pacote que faz a comunicação entre os dois. Abaixo segue a descrição de cada pacote.

### Interface gráfica

Composto pelas classes responsáveis pelos elementos gráficos do simulador. Trata toda comunicação com o usuário, armazenando toda informação gerada pelo usuário. É através dele que o usuário fornece os dados necessários para que o módulo do kernel execute a simulação e também é através dele que o usuário poderá obter os resultados da simulação na forma de gráficos.

## Kernel

Composto pelas classes que fazem a simulação. Cria a estrutura fornecida na interface gráfica pelo usuário e gera os resultados da simulação.

## Configurador do kernel

Obtém as informações fornecidas pelo usuário da interface gráfica e gera um conjunto de arquivos no formato entendido pelo kernel numérico. De forma que a interface nada sabe sobre o funcionamento do kernel e o kernel também nada sabe sobre a interface.

A Figura 3.1 apresenta o diagrama de pacotes do Simulador. Na figura podemos ver que o controlador tem uma relação de dependência com os demais pacotes para gerar a comunicação

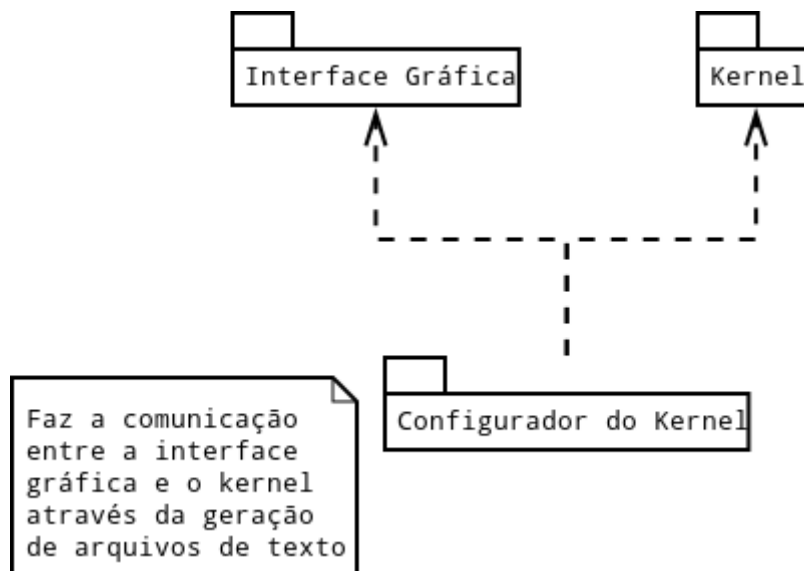


Figura 3.1: Diagrama de pacotes do TFS

## 3.3 Análise de domínio

O *Turtle Flow Simulator* abrange a área de engenharia de reservatório na qual modelos computacionais são usados para prever o fluxo de fluídos, tipicamente petróleo, água e gás natural através de meios porosos.

A importância da simulação de reservatórios se dá na tentativa de estudar e aprender mais sobre um campo de petróleo antes mesmo de perfurá-lo e/ou para prever seu comportamento ao longo de anos. São usados por companhias de petróleo e gás no desenvolvimento de novos campos e em campos desenvolvidos, onde as previsões de produção são necessárias para ajudar a tomar decisões de investimento. Como construir e manter um modelo experimental robusto e confiável de um campo é muitas vezes demorado e caro, os modelos são normalmente construídos apenas onde decisões de investimento vultosos estão em jogo. Melhorias no software de simulação reduziram o tempo necessário

para desenvolver um modelo numérico com precisão aceitável capaz de substituir modelos experimentais de reservatórios.

### Softwares relacionados:

#### Merlin

Criado pela empresa *Gemini Solutions Inc* ([www.Gemini.com](http://www.Gemini.com), acessado em 03/02/2014) o Merlin é um software de simulação de reservatório de óleo e gás para aplicações convencionais, pseudo composicional e de porosidade dupla. Permite construir, executar e visualizar os resultados de uma simulação de reservatório com relativa rapidez. A Figura 3.2 apresenta uma imagem representativa do simulador.

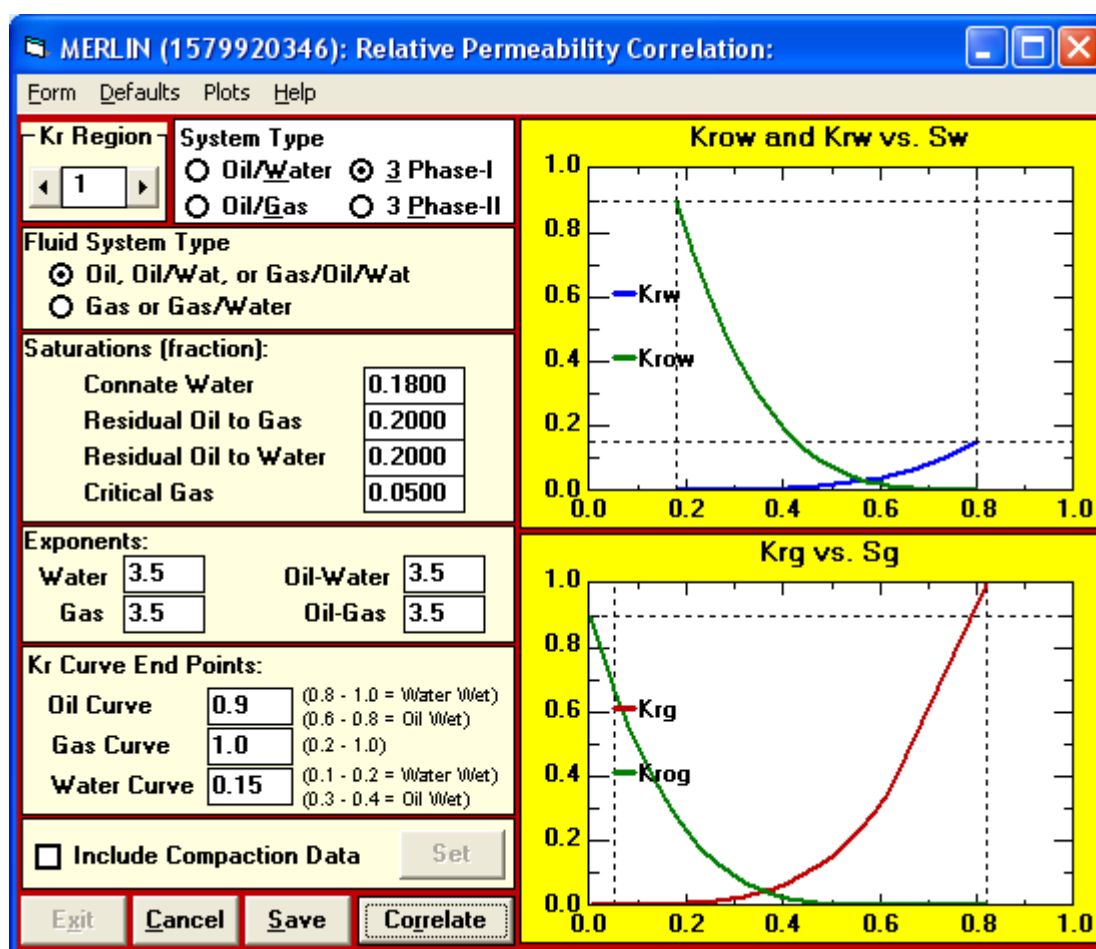


Figura 3.2: Ilustração do simulador Merlin

#### JewelSuite

O JewelSuite foi desenvolvido pela empresa Baker Hughes ([www.Jewelsuite.com](http://www.Jewelsuite.com), acessado em 03/02/2014) e obtém resultados precisos, possuindo um modelo patentado de criação de grids. Simulação de modelos dinâmicos pode ser controlada diretamente a partir do software e ainda pode ser conectado a vários outros sistemas externos como, por



exemplo, os simuladores IMEX, GEM, SENSOR e ECLIPSE. Na Figura 3.3 é possível ver uma captura de tela do programa em execução.

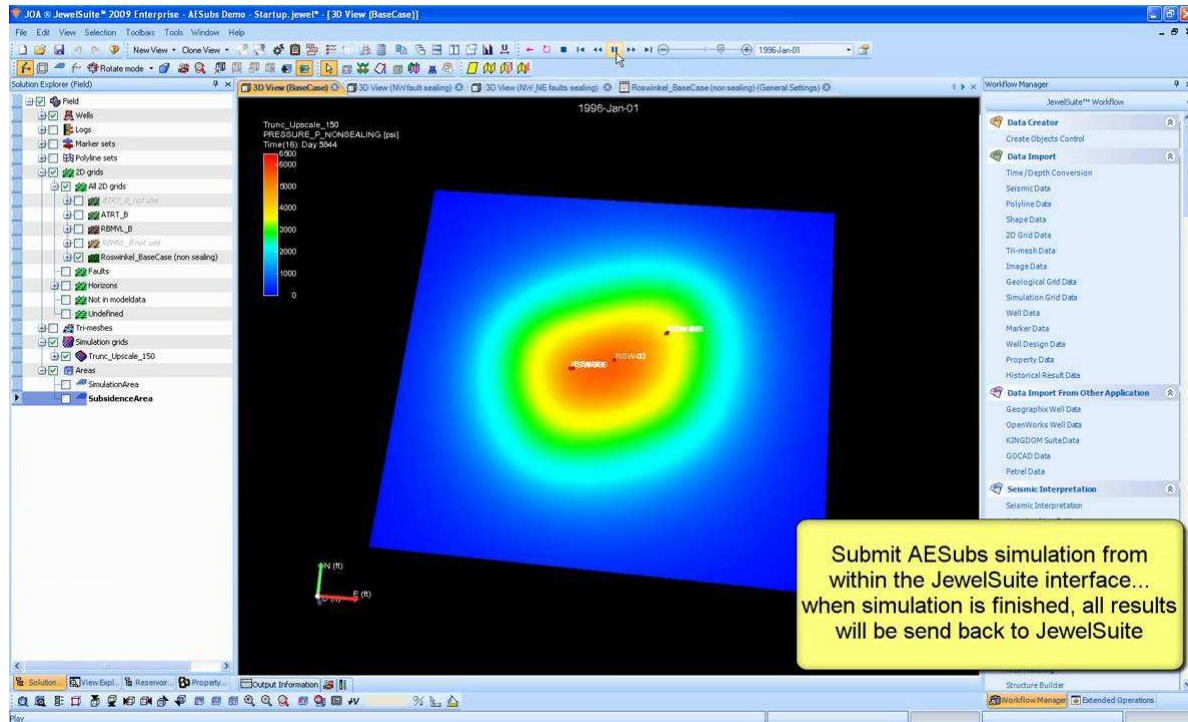


Figura 3.3: Captura da tela do simulador JewelSuite em execução

# Capítulo 4

## AOO - Análise Orientada a Objeto

Nesse capítulo serão mostrados alguns diagramas da linguagem UML (*Unified Modeling Language*) que foram desenvolvidos durante a idealização e concepção do software.

A linguagem UML é uma linguagem de modelagem não proprietária, que permite que desenvolvedores visualizem a estrutura interna de um software através de diagramas padronizados.

### 4.1 Diagramas de classes

O diagrama de classes é uma representação das relações que as classes de um software orientado a objetos possuem entre si. É um dos diagramas mais úteis da linguagem UML pois mostra de forma clara as conexões entre as classes de um software.

#### 4.1.1 Classes gerais

A classe CReservoir é a principal classe do programa, responsável por inicializar todos os objetos e comandar a execução do código numérico. Essa classe é considerada como o centro do programa. Foram implementadas três classes abstratas CGrid, CResModelType e CSolverMatrix que são as classes que modelam a estrutura principal do simulador. A medida que novos modelos forem implementados, utilizar-se-á o conceito de polimorfismo para essas classes. A classe CDataControl é utilizada para carregar os arquivos .dat do disco, e será utilizado nos construtores das classes principais do programa.

A Figura 4.1 mostra a interação entre as classes principais, que serão discutidas detalhadamente nos próximos tópicos.

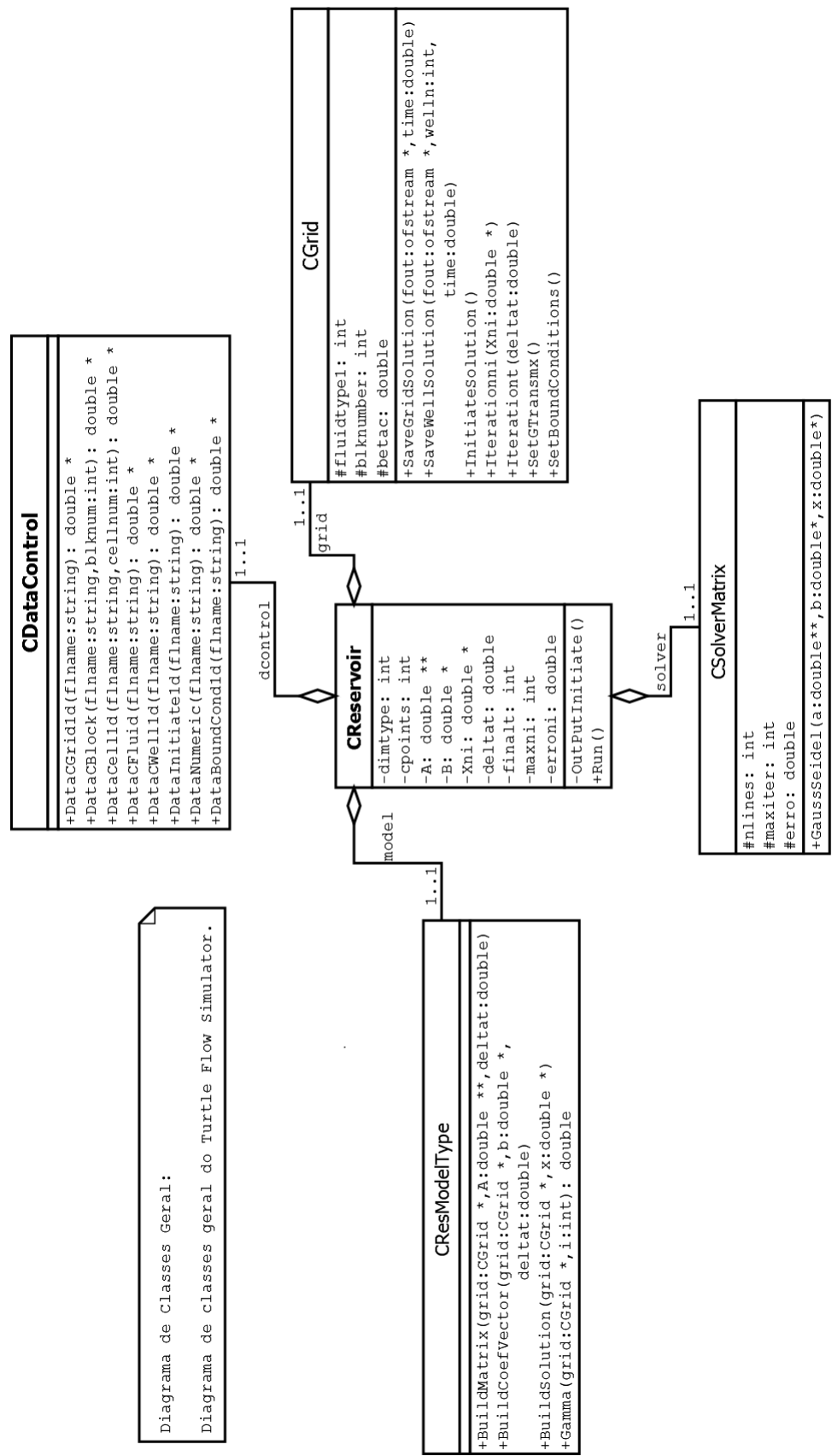


Figura 4.1: Diagrama geral de classes

### 4.1.2 Classes de grid

Em uma simulação de reservatórios, o número de informações que devem ser armazenadas na memória é muito grande, dessa forma as informações são tratadas em três níveis: Célula, Bloco e Malha (Figura 4.2). A célula (Figura 4.2-(c)) é a menor unidade da simulação onde serão resolvidas as equações governantes do problema. O bloco (Figura 4.2-(b)) é um conjunto de células com propriedades de rocha iguais. Dessa forma, evita-se a repetição de informação. A malha (ou grid) (Figura 4.2-(a)) é a maior unidade possível, e expressa todo o domínio do reservatório. Nessa figura as células pintadas com a mesma cor representam um mesmo bloco, ou seja, as propriedades de rocha dessas células são iguais.

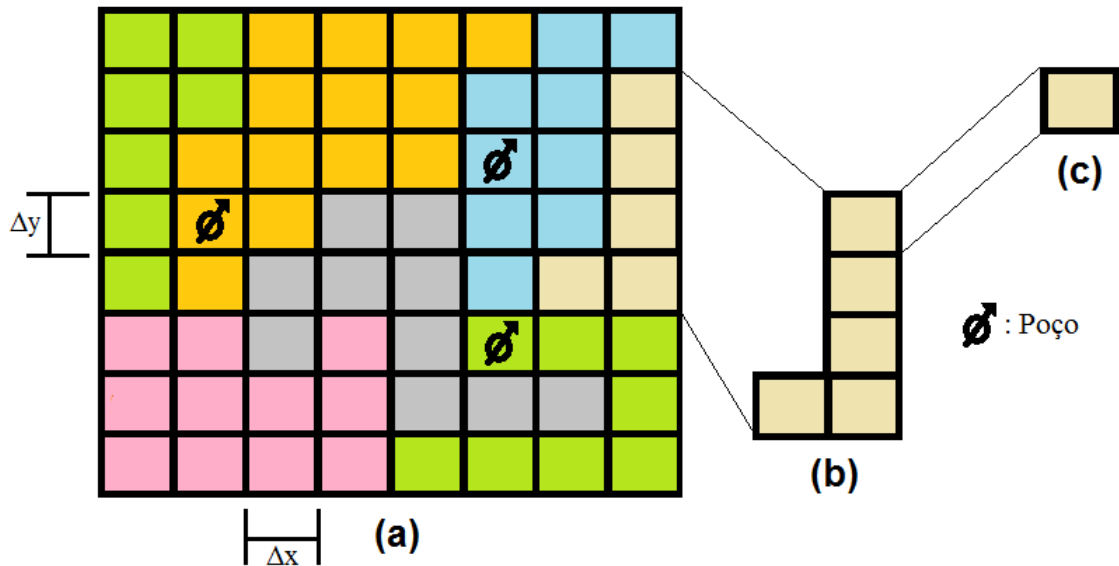


Figura 4.2: Diagrama representando a malha computacional do problema. (a) é a representação de um grid, (b) é um bloco do problema e (c) é uma única célula do domínio

Observa-se a presença de três poços nesse exemplo, assim os poços são armazenados na célula. Como as malhas são regulares, as informações de espaçamento horizontal são armazenados na malha. A Figura 4.3 mostra o diagrama de classes CGrid de forma detalhada. Ressalta-se entretanto que apenas as classes unidimensionais foram implementadas nessa versão do TFS.

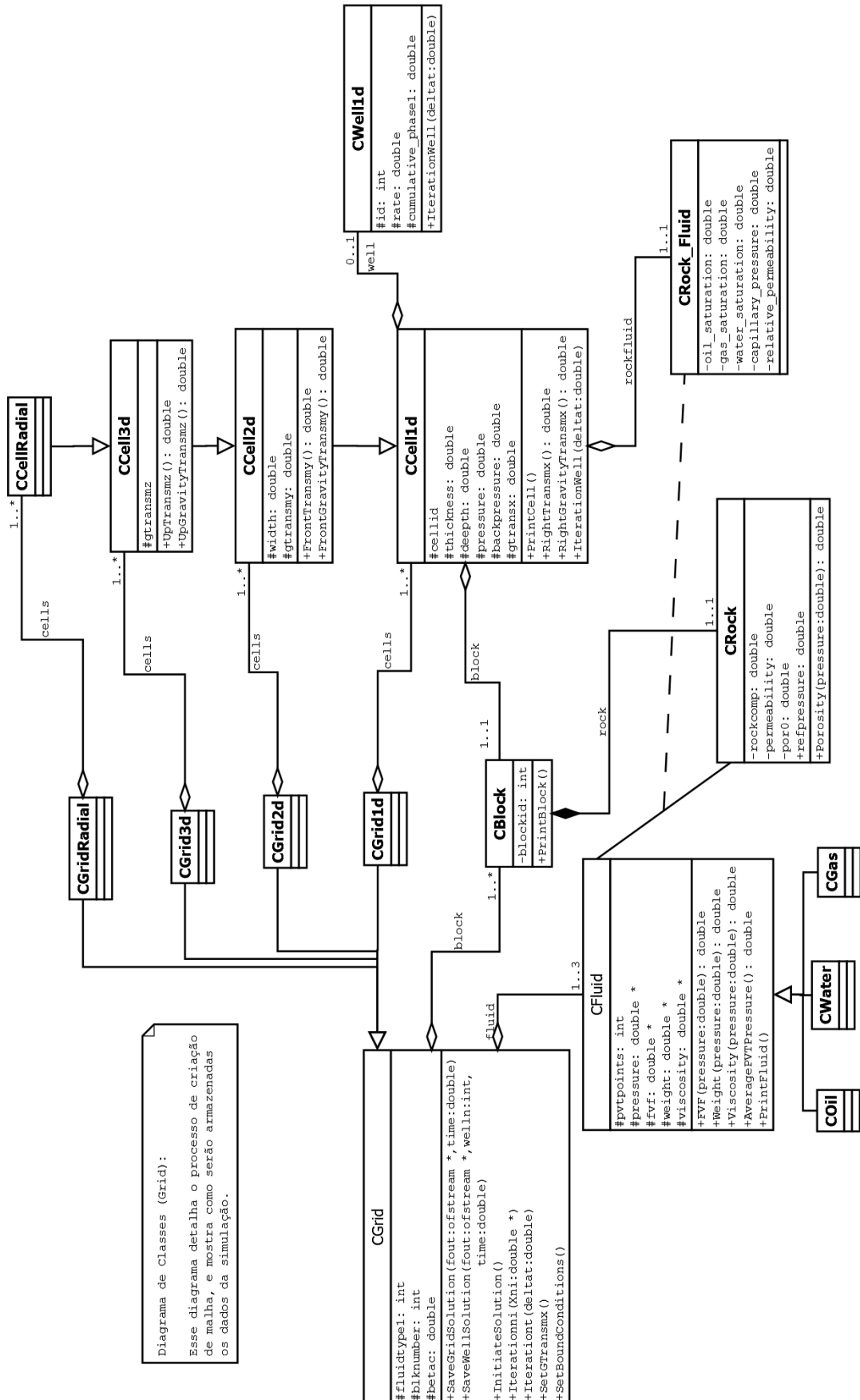


Figura 4.3: Diagrama de classes para um software de reservatórios

### 4.1.3 Classes de modelos de discretização

Essa classe é responsável pela discretização das equações diferenciais parciais e construção do sistema linear de equações que serão resolvidas pelo solver numérico. A Figura 4.4 mostra o diagrama de classes para essa classe.

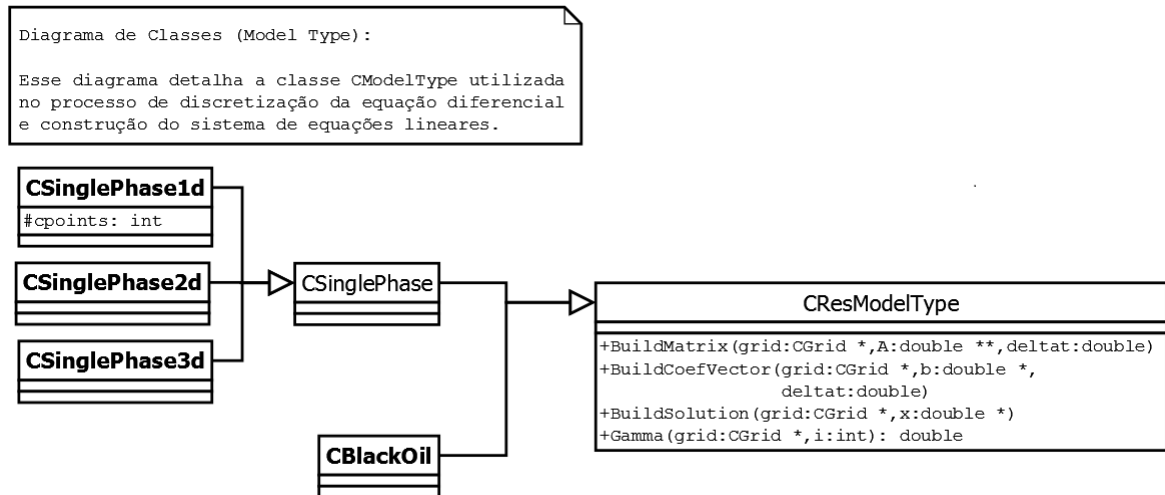


Figura 4.4: Diagrama de classes CModelType

Como é possível perceber, essa classe possui as seguintes classes filho:

1. CSinglePhase: Essa classe discretiza as equações de transporte considerando que existe somente uma fase líquida saturando o meio poroso.
2. CBlackOil: Essa classe discretiza as equações de transporte considerando o modelo “*Black Oil*” de fluido. Na versão atual do TFS essa classe não foi implementada.

### 4.1.4 Classes de solver

Essa classe é o código numérico principal do problema. Representa as funções necessárias para se resolver os sistemas de equações lineares do problema. As equações serão constantemente resolvidas, dessa forma, o desempenho do software está diretamente ligado com o desempenho dessa classe. A Figura 4.5 mostra o diagrama de classes para o solver.

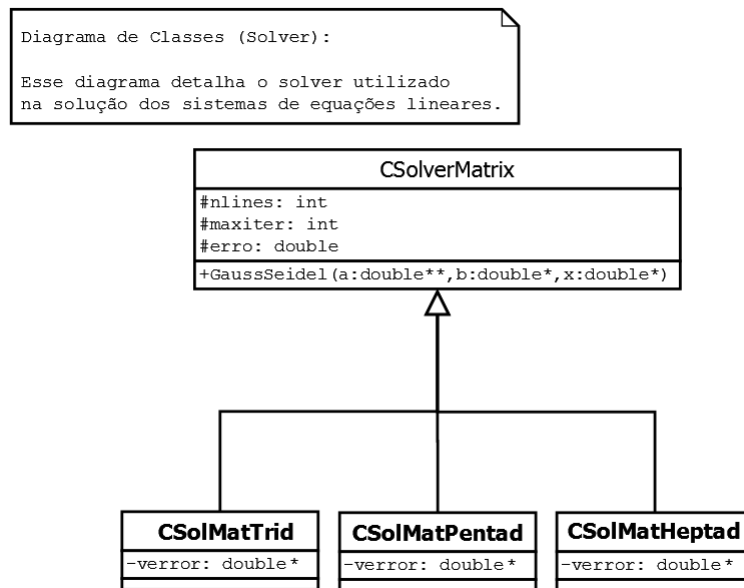


Figura 4.5: Diagrama de classes CSolver

A classe CSolverMatrix possui três classes filho:

1. CSolMatTrid: Responsável por resolver problemas unidimensionais;
2. CSolMatPentad: Responsável por resolver problemas bidimensionais (não implementado);
3. CSolMatHeptad: Responsável por resolver problemas tridimensionais (não implementado).

#### 4.1.5 Classes de interface gráfica

A Figura 4.6 mostra as classes utilizadas na interface gráfica do TFS.

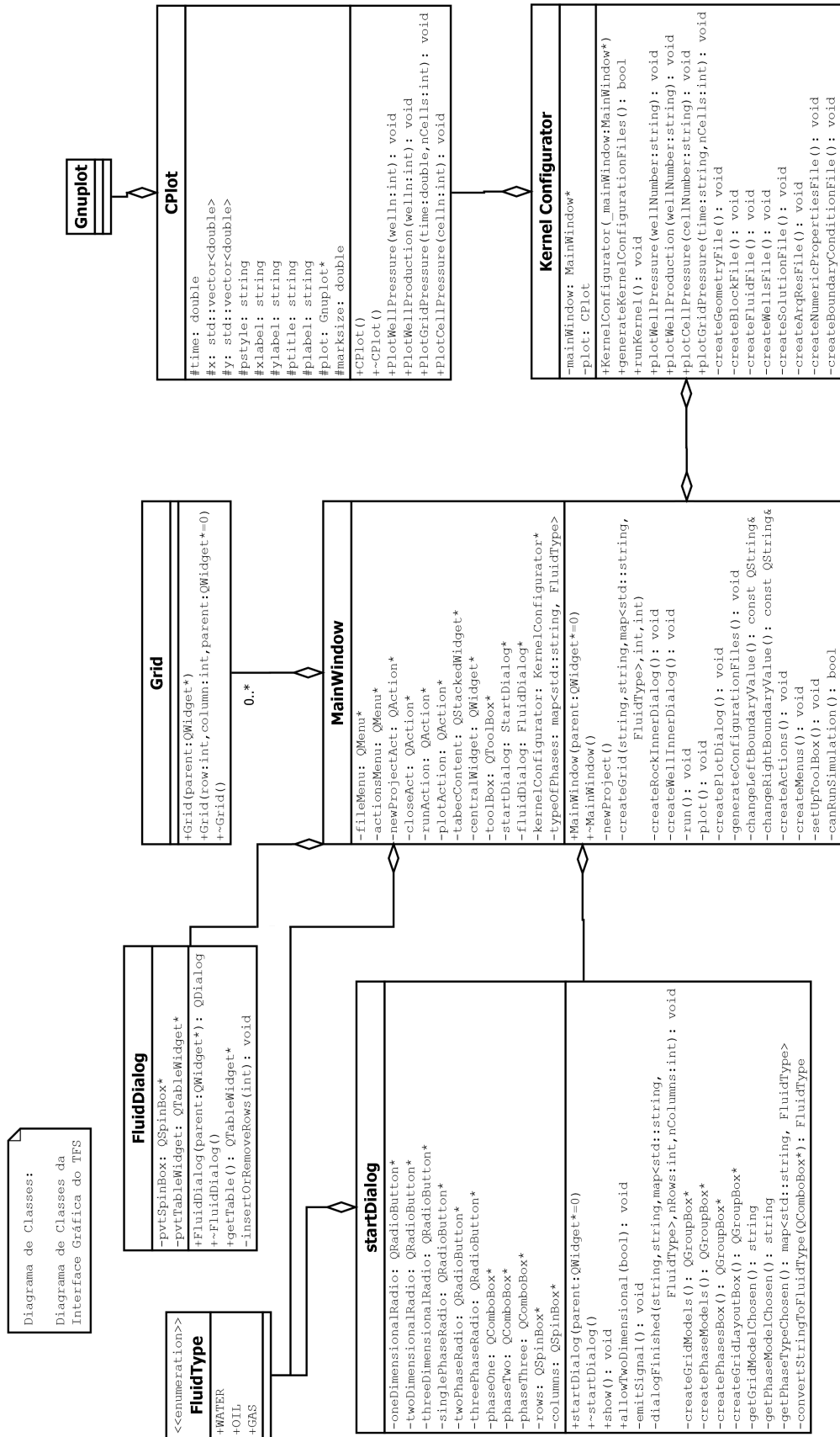


Figura 4.6: Diagrama de classes interface gráfica



## 4.2 Diagrama de sequência - eventos e mensagens

Baseado nos casos de uso do TFS, foram criados diagramas de sequências mostrando a interação entre os objetos do software.

### 4.2.1 Diagrama de sequência “Inserir Dados do Problema”

A Figura 4.7 mostra o diagrama de sequência para o caso de uso “Inserir Dados do Problema”. Observa-se a constante troca de informação entre a interface gráfica e o usuário.

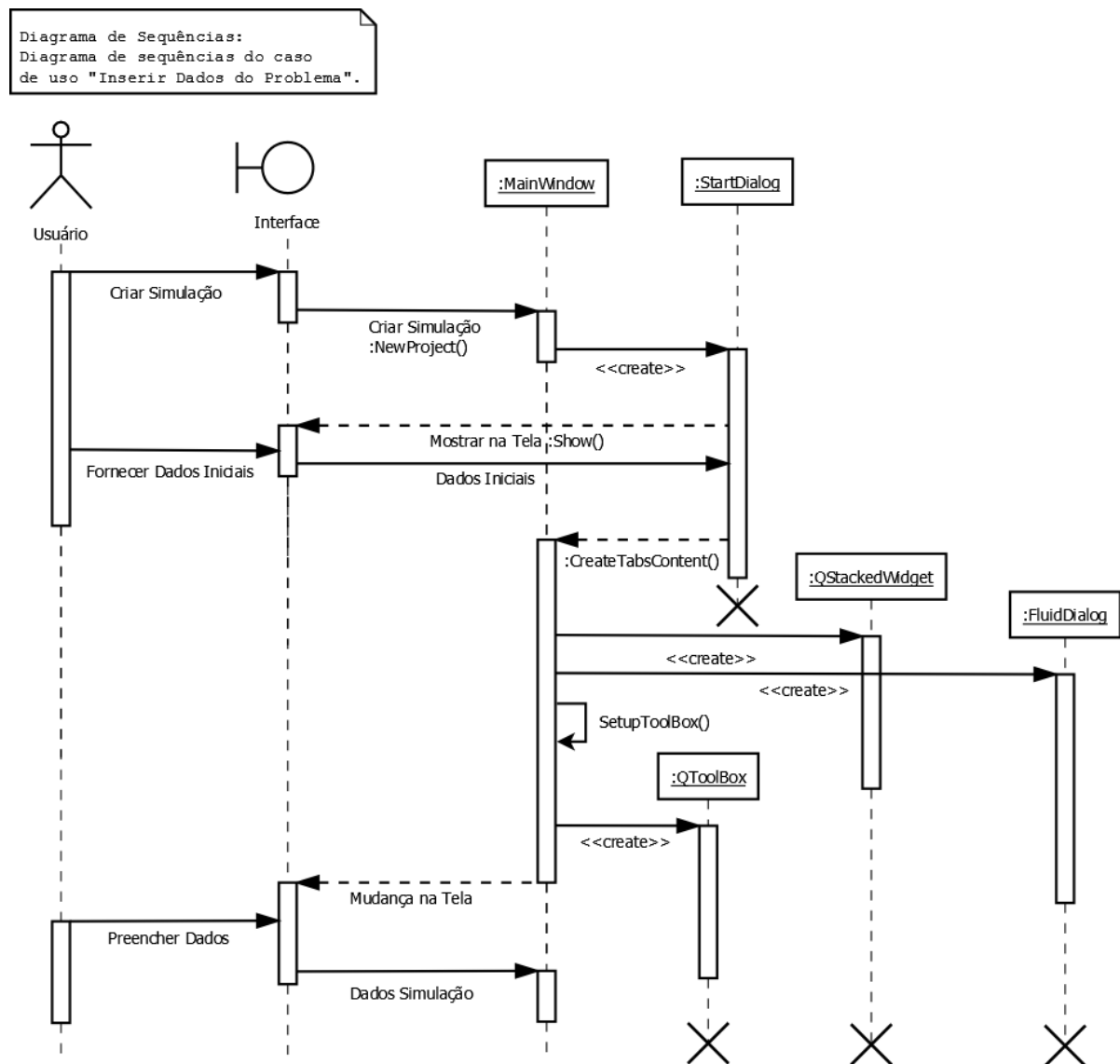


Figura 4.7: Diagrama de sequência “Inserir Dados do Problema”

### 4.2.2 Diagrama de sequência “Gerar Arquivos de Configuração”

A Figura 4.8 mostra o diagrama de sequência para o caso de uso “Gerar Arquivos de Configuração”.

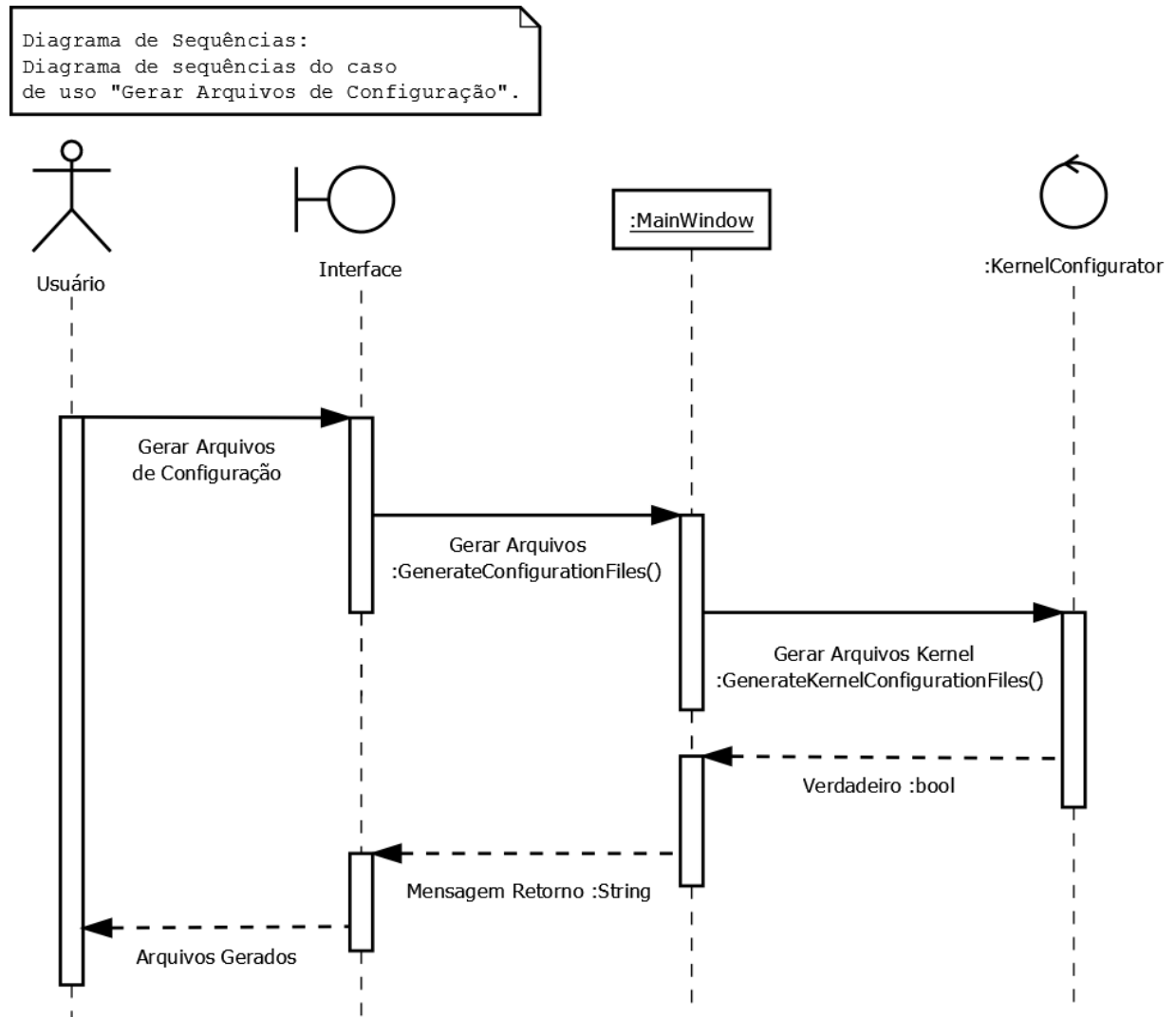


Figura 4.8: Diagrama de sequência para o caso de uso “Gerar Arquivos de Configuração”

### 4.2.3 Diagrama de sequência “Executar Simulação”

Esse diagrama foi dividido em duas partes distintas, a primeira parte mostra como o caso de uso “Executar Simulação” é modelado pela interface gráfica (Figura 4.9).

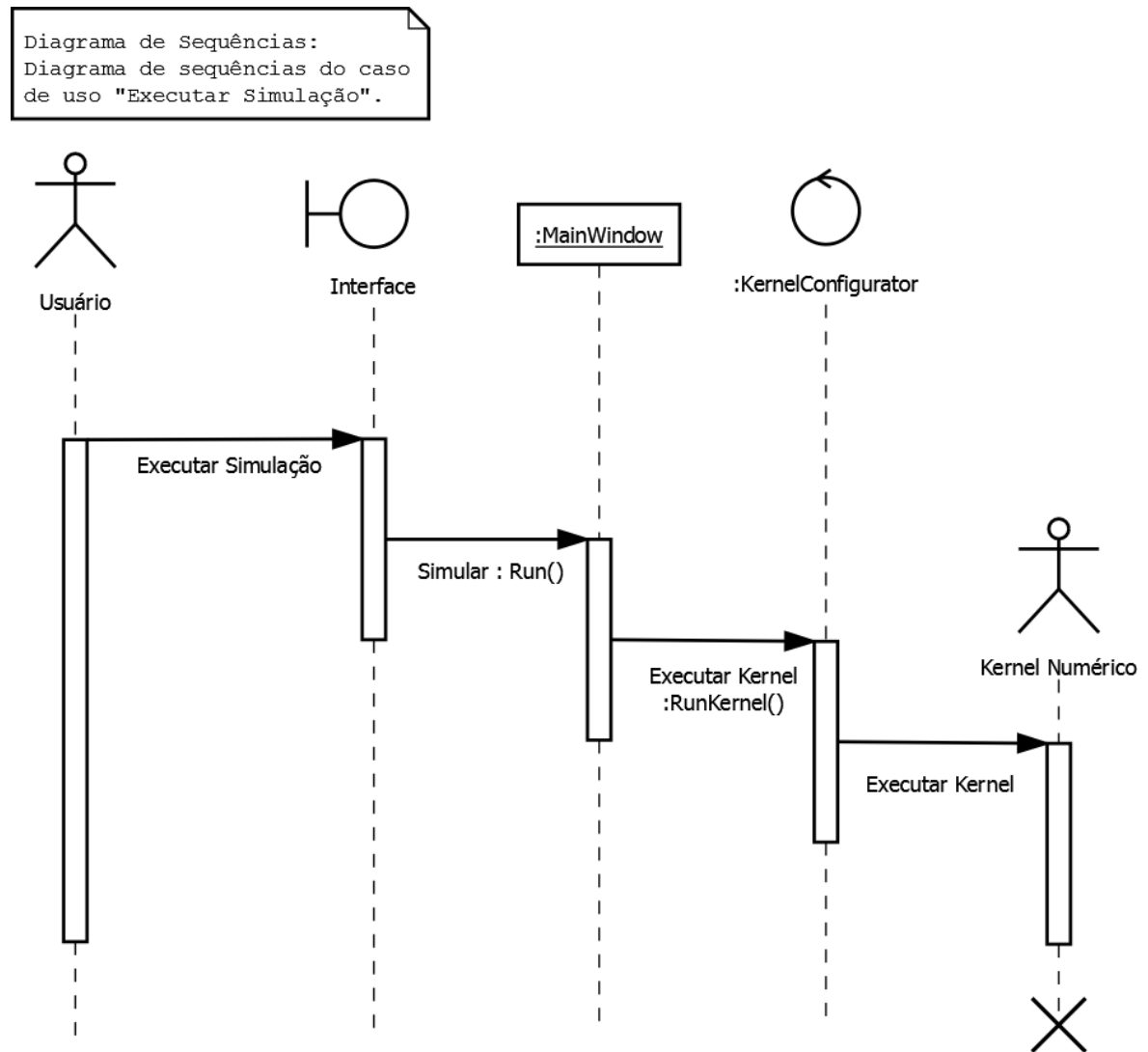


Figura 4.9: Diagrama de sequência para o caso de uso “Executar Simulação” sob o ponto de vista da interface gráfica

A Figura 4.10 mostra a segunda parte desse diagrama de sequências, que especifica o caso de uso “Executar Simulação” do ponto de vista do *kernel* numérico.

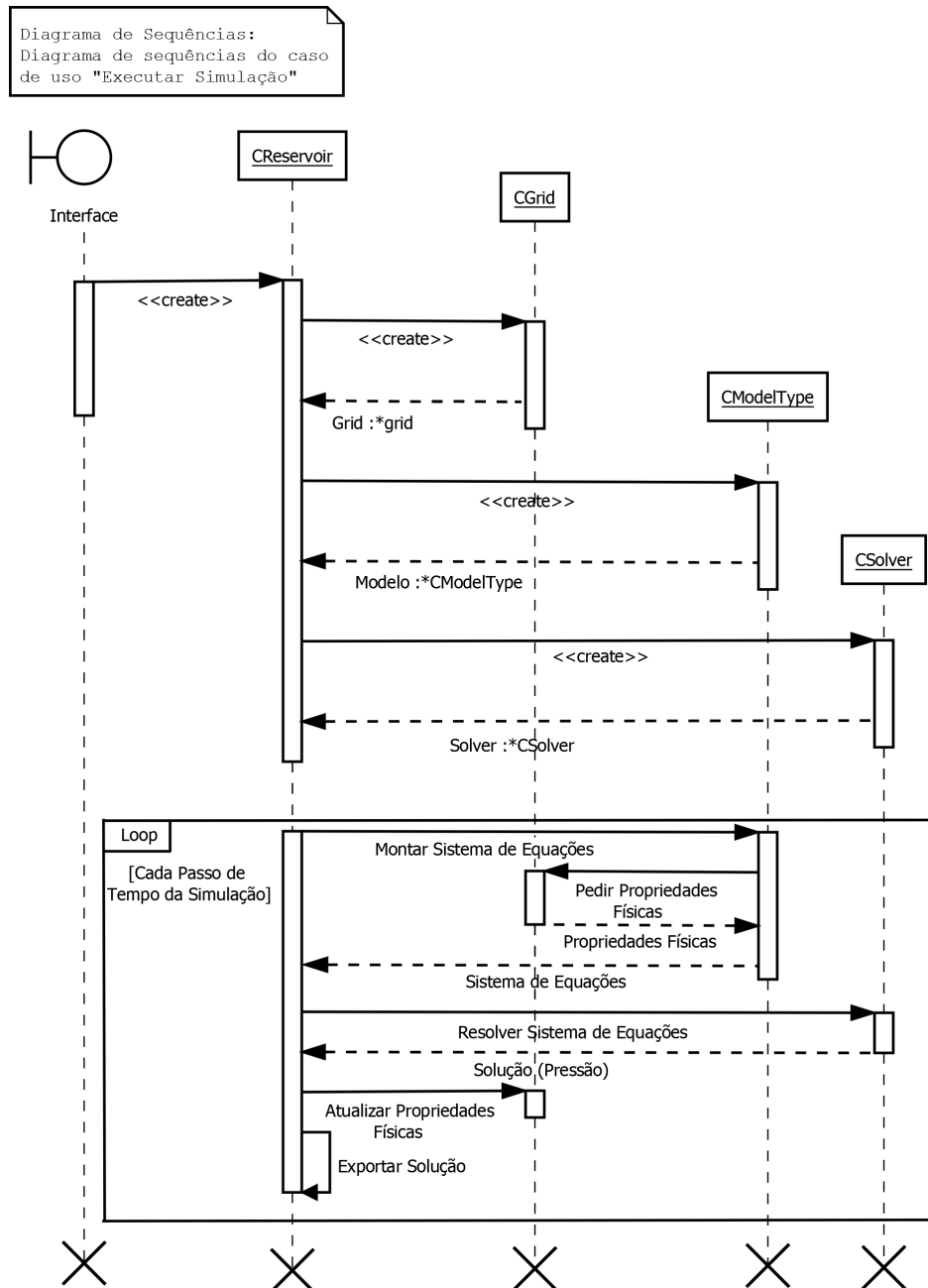


Figura 4.10: Diagrama de sequência para o caso de uso “Executar Simulação” sob o ponto de vista do *kernel* numérico

### 4.3 Diagrama de atividades

O método `Simulate` da classe `CReservoir` é o mais importante do *kernel* numérico, pois controla todo o fluxo da simulação numérica. A Figura 4.11 mostra o diagrama de atividades para esse método.

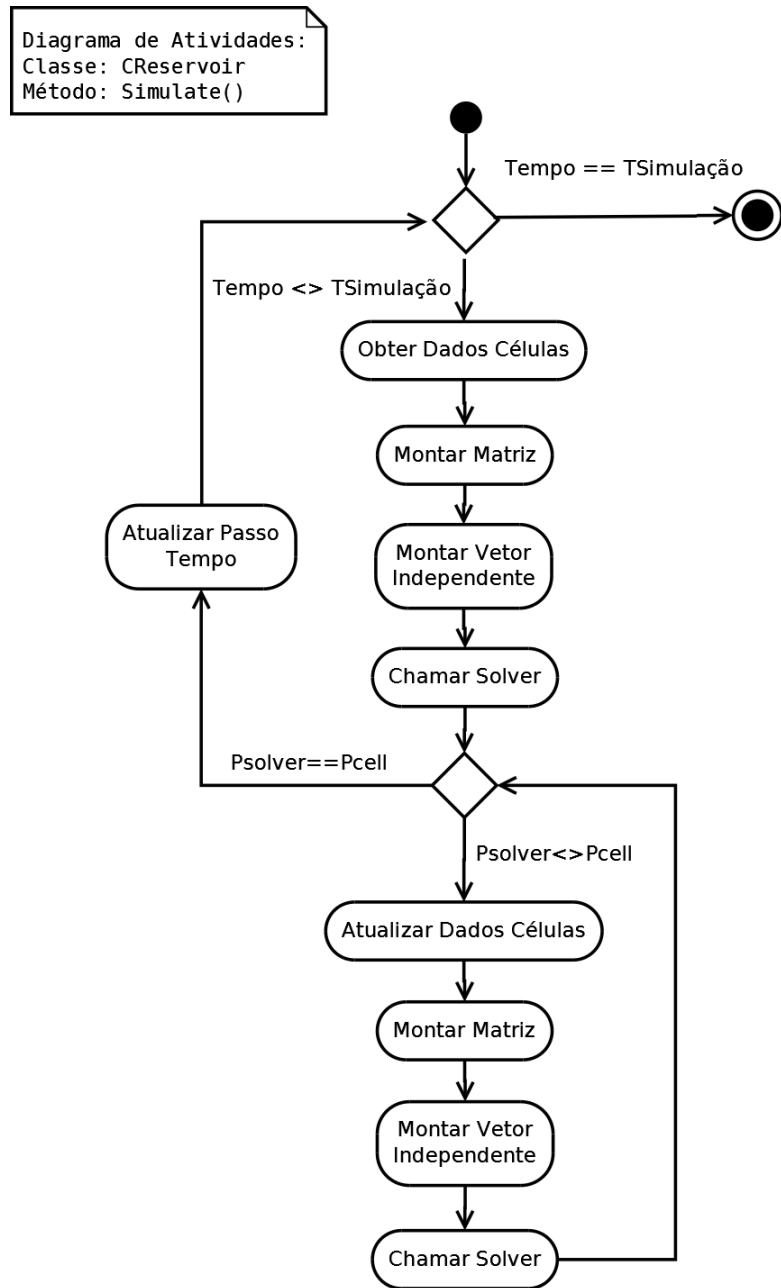


Figura 4.11: Diagrama de atividades do método Simulate da classe CReservoir

# Capítulo 5

## Projeto

São apresentados nesse capítulo os projetos de software utilizados para o desenvolvimento do TFS.

### 5.1 Projeto do sistema

Como esse projeto baseia-se na criação e uso de módulos, a comunicação entre esses módulos, principalmente entre os módulos da interface e o módulo do kernel numérico é de extrema importância para o sistema, e sendo assim, adotamos o uso de arquivos no formato txt.

A comunicação funciona com o módulo da interface gráfica recebendo os dados e comandos do usuário para a geração dos arquivos de configuração. O módulo do configurador do Kernel gera os arquivos pedidos pelo usuário e executa o kernel numérico que então lê os dados presentes nos arquivos de configuração. Esse processo pode ser visto na Figura 5.1.

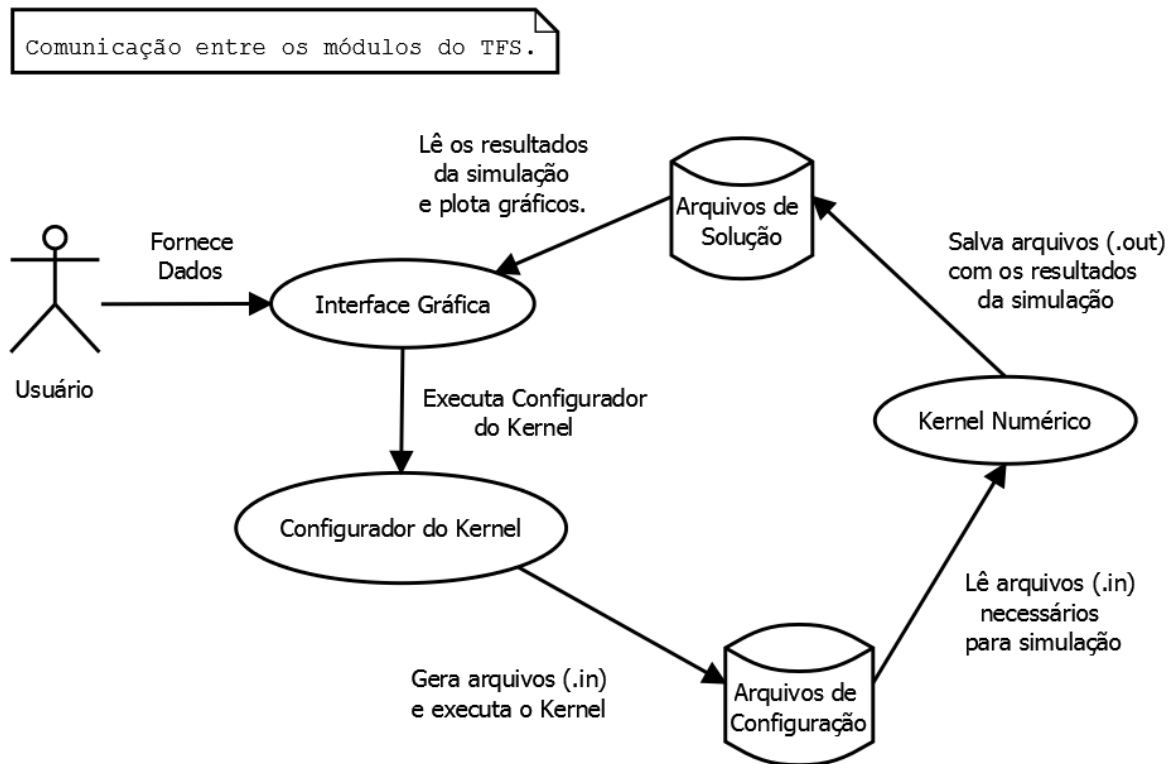


Figura 5.1: Comunicação entre os módulos do sistema

Na Figura 5.2 temos um exemplo de um arquivo de texto utilizado onde o número cinco representa o número de pontos de pressão e a tabela, os respectivos valores de Pressão, FVF (Fator Volume-Formação), peso específico e viscosidade. Como os dados são simples, não há necessidade da utilização de um SGBD (Sistema Gerenciador de Banco de Dados) como o PostgreSQL ou o MySQL. O Apêndice A descreve detalhadamente a estrutura dos arquivos utilizados pelo TFS.

5			
1000	1.02527	0.43214	0.5200
2000	1.02224	0.43342	0.5200
3000	1.01621	0.43599	0.5200
4000	1.01024	0.43856	0.5200
5000	1.00506	0.44113	0.5200

Figura 5.2: Exemplo de arquivo de saída .dat

## 5.2 Plataformas suportadas

O sistema suporta os sistemas operacionais Windows e GNU/Linux. Conta com arquivos de configuração que no momento da compilação detectam o ambiente no qual está sendo executado e carrega as propriedades que são características de cada sistema de forma que para o usuário fique transparente.

### 5.3 Bibliotecas externas

O sistema utiliza a biblioteca de interface gráfica Qt ([www.qt-project.org](http://www.qt-project.org), acessado em 03/02/2014), o qual é um framework multiplataforma (Windows, GNU/Linux, Mac OS) para desenvolvimento de interfaces gráficas em C++ originalmente criado pela empresa norueguesa TrollTech. Dessa forma, havendo interesse futuro, será possível modificar o programa para que funcione facilmente em outras plataformas, como por exemplo, o Mac OS.

Para o desenvolvimento dos gráficos utilizou-se as bibliotecas Gnuplot ([www.gnuplot.info](http://www.gnuplot.info), acessado em 03/02/2014), que possuem a facilidade de ser um software livre e multiplataforma. Assim é necessário que o sistema possua o GnuPlot instalado para a perfeita execução do TFS.

### 5.4 Ambiente de desenvolvimento

Para o desenvolvimento desse projeto, foi escolhido o ambiente de desenvolvimento Dev C++ ([www.bloodshed.net/devcpp](http://www.bloodshed.net/devcpp), acessado em 03/02/2014) para desenvolvimento no Windows. Para desenvolvimento no GNU/Linux, foi utilizado o editor de texto Sublime ([www.sublimetext.com](http://www.sublimetext.com), acessado em 03/02/2014) em conjunto com o compilador g++.

O controle de versão do código foi feito utilizando-se o git ([www.git-scm.com](http://www.git-scm.com), acessado em 03/02/2014). O controle de versão é importante pois possibilita o acompanhamento da evolução do código e recuperação de versões anteriores.

### 5.5 Requisitos do sistema

Para execução do simulador é preferível um computador com um processador rápido e bastante memória disponível dado que a simulação requer muito processamento e resolução de operações com matrizes. O software foi idealizado para economizar o consumo de memória RAM, pois esse é o recurso crítico em simulações de reservatório.

Recomenda-se um Processador Intel Core i5 de 2.5 GHz com 4 GB de memória RAM ou equivalente.



# Capítulo 6

## Implementação

A Figura 6.1 mostra o diagrama global de componentes do sistema. É possível ver todos os arquivos .h e .cpp necessários para a compilação do TFS.

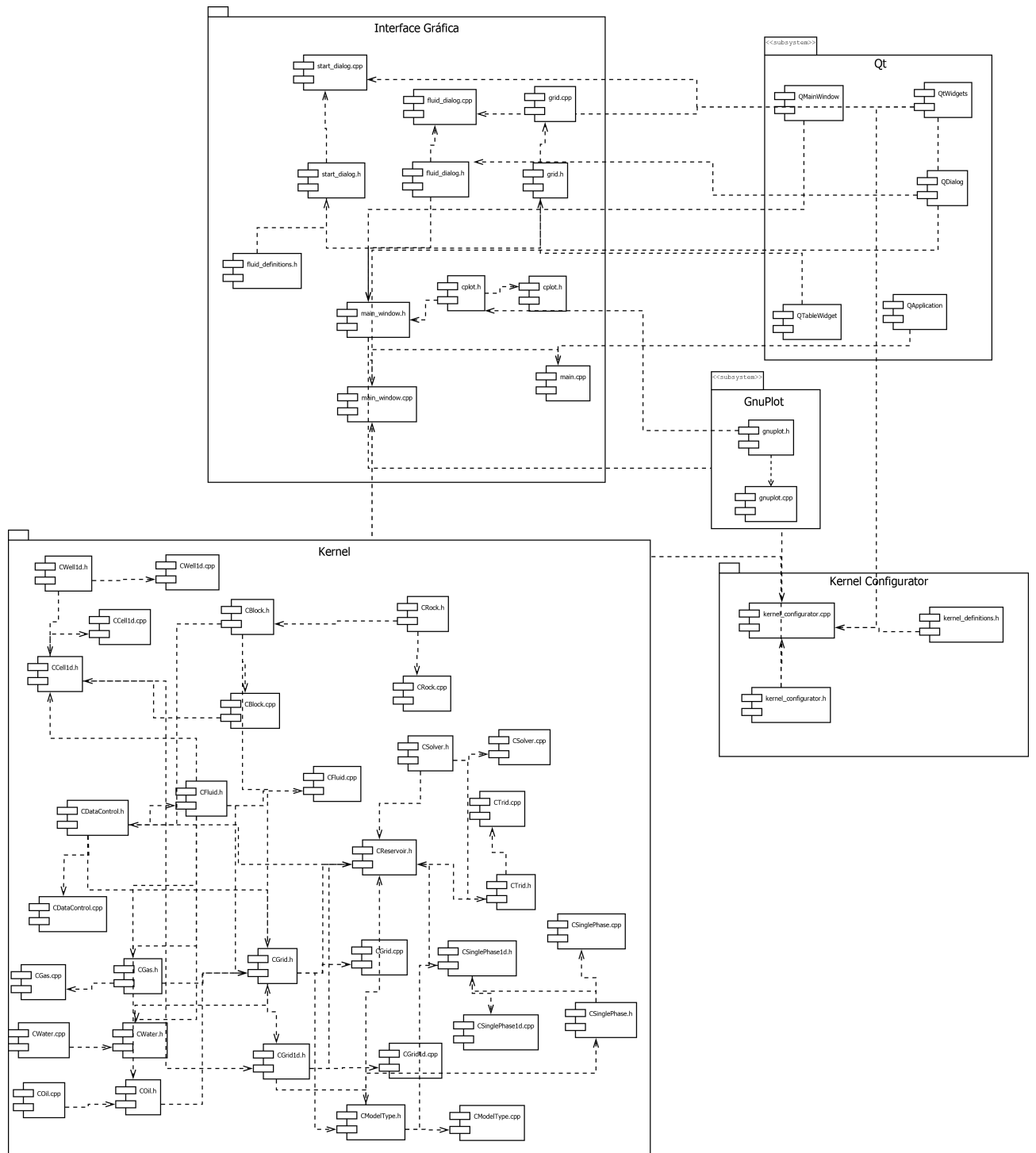


Figura 6.1: Diagrama global de componentes do sistema

A Figura 6.2 mostra um detalhamento do pacote “Interface Gráfica” da Figura 6.1.

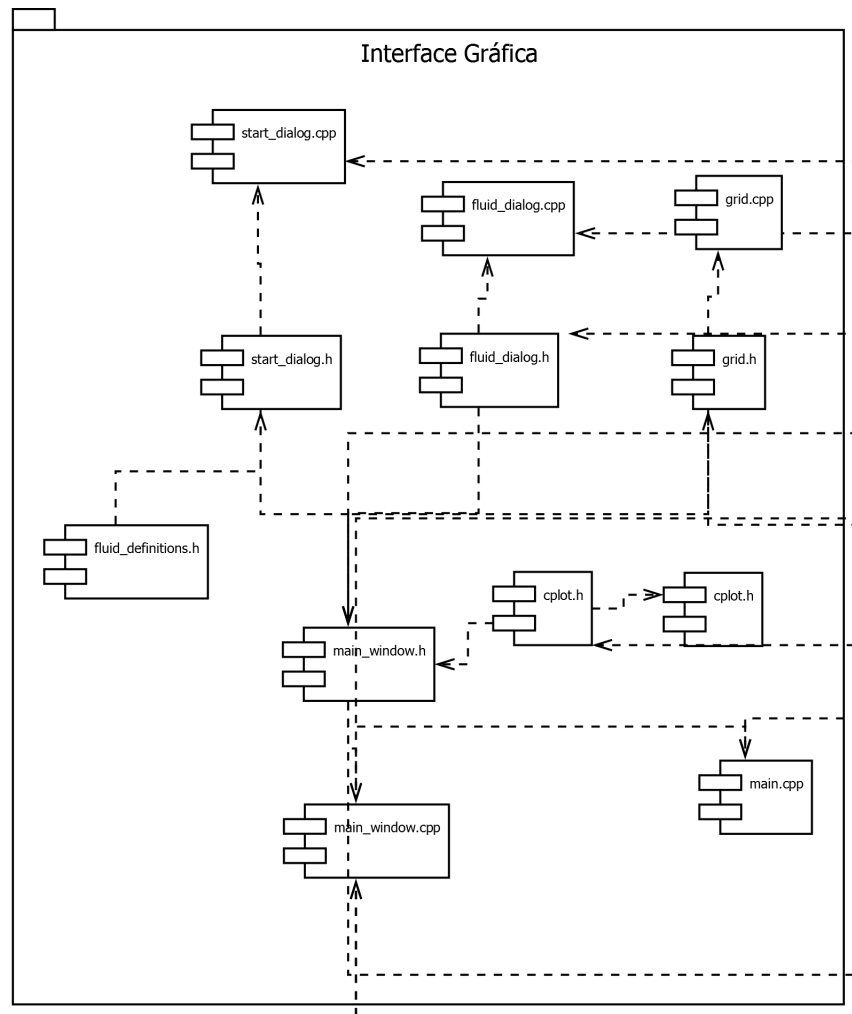


Figura 6.2: Detalhamento do pacote “Interface Gráfica” da Figura 6.1

A Figura 6.3 mostra um detalhamento do pacote “Qt” da Figura 6.1.

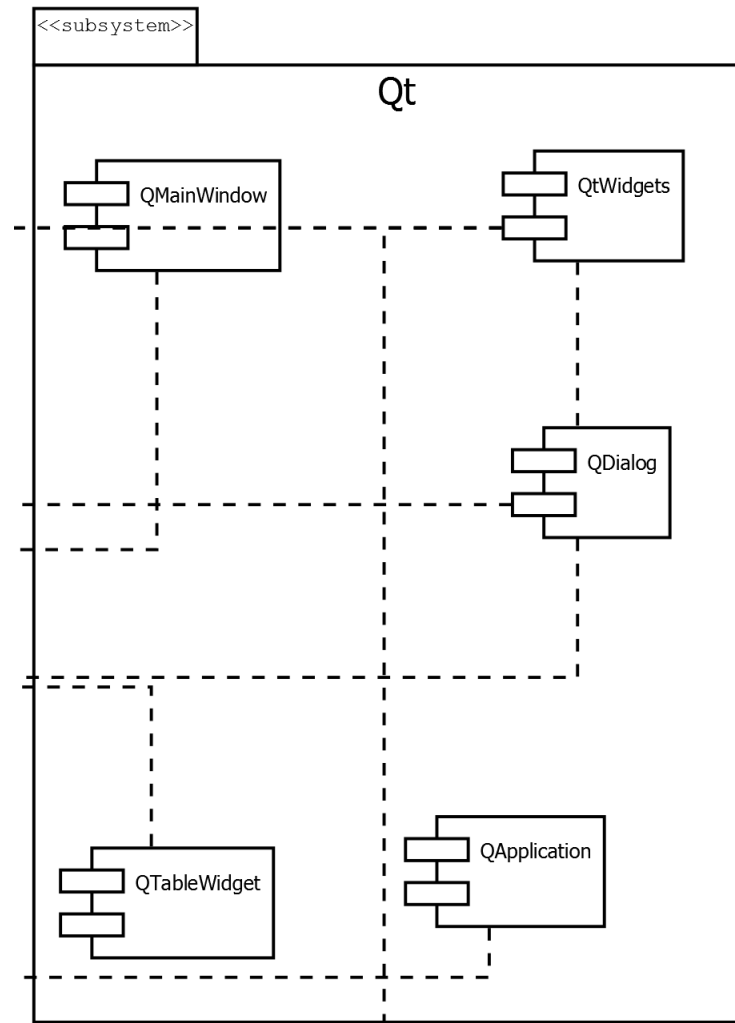


Figura 6.3: Detalhamento do pacote “Qt” da Figura 6.1

A Figura 6.4 mostra um detalhamento dos pacotes “GnuPlot” e “Kernel Configurator” da Figura 6.1.

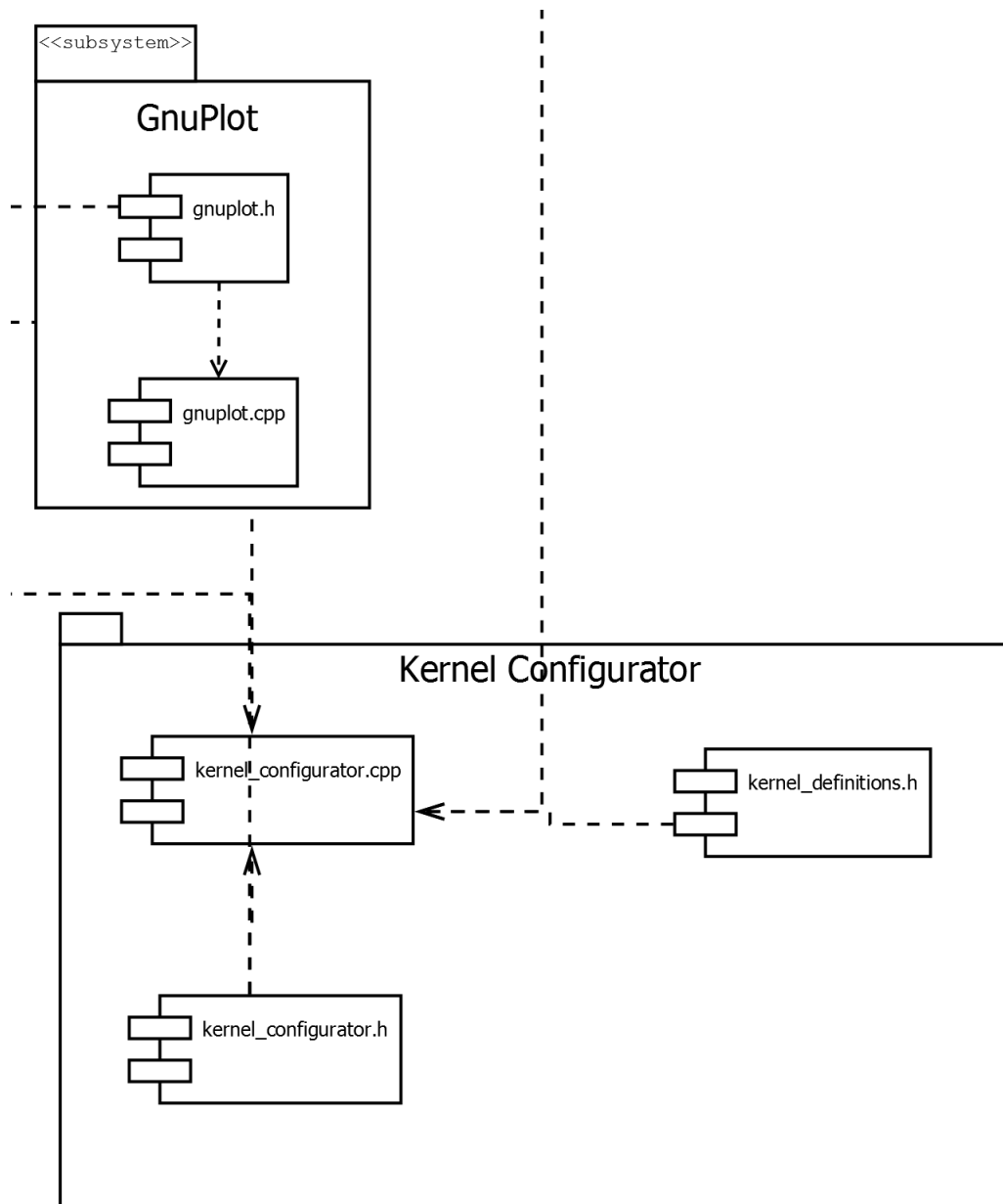


Figura 6.4: Detalhamento dos pacotes “GnuPlot” e “Kernel Configurator” da Figura 6.1

A Figura 6.5 mostra um detalhamento do pacote “Kernel” da Figura 6.1.

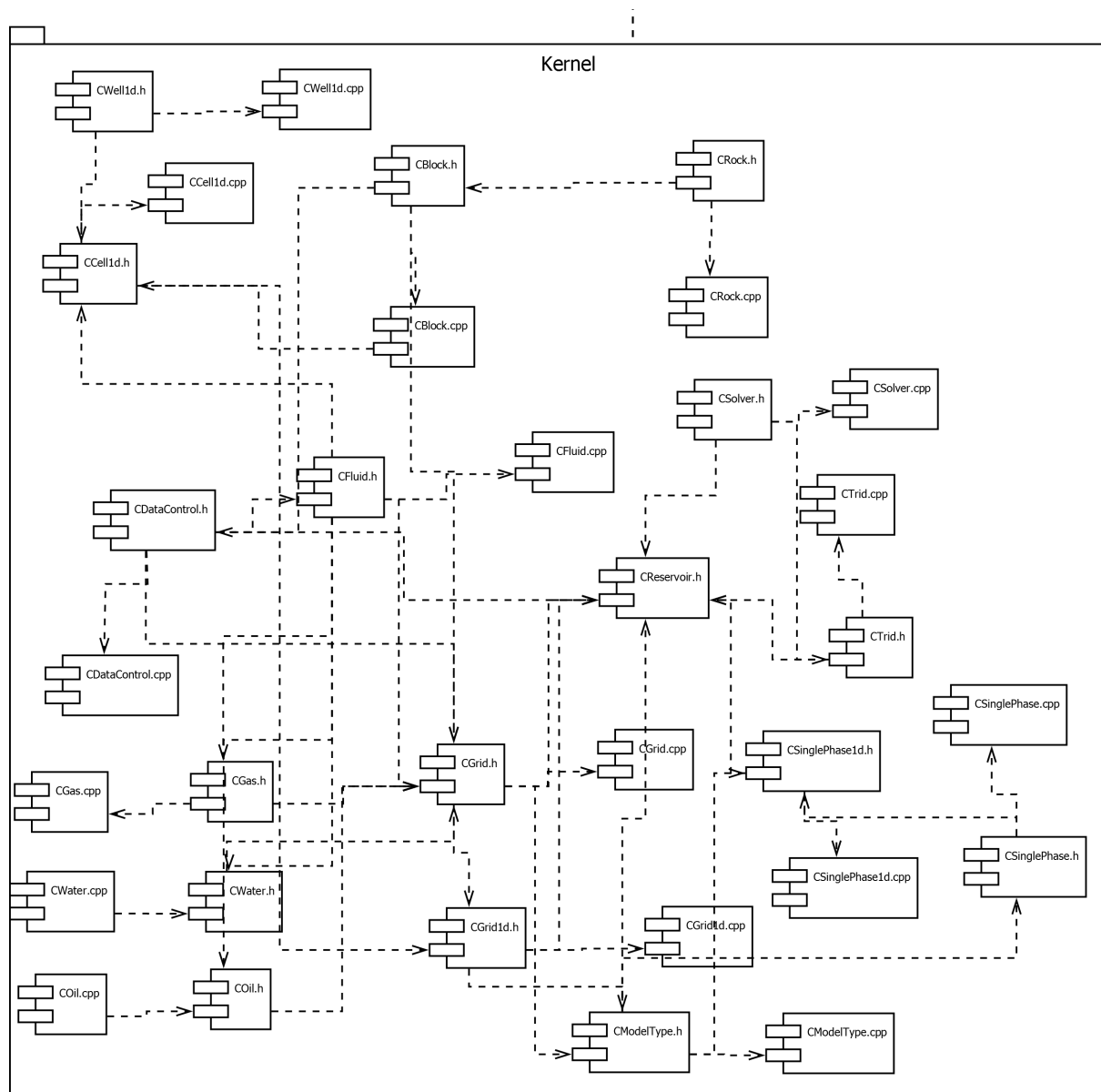


Figura 6.5: Detalhamento do pacote “Kernel” da Figura 6.1

# Capítulo 7

## Teste

São apresentados nesse capítulo os testes realizados com o TFS de forma a se validar os resultados numéricos obtidos com o software.

### 7.1 Teste 1:

O primeiro teste do TFS foi uma comparação com o Problema 5.8 do livro [Ertekin et al., 2001], mostrado na Figura 7.1.

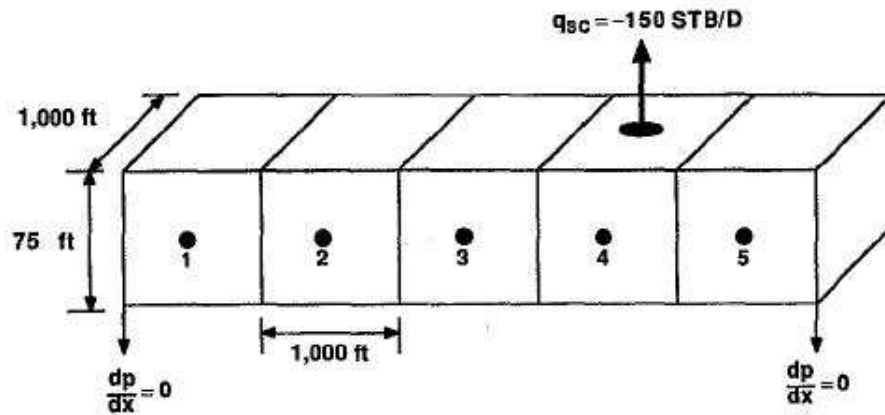


Figura 7.1: Reservatório do Exemplo 5.8. (Figura 5.26 [Ertekin et al., 2001])

Considerando um tempo total de simulação de 360 dias os resultados do simulador foram comparados com os resultados explícito e implícito apresentados por Ertekin. A Tabela 7.1 mostra as diferenças entre os resultados encontrados pelo TFS e os apresentados pelo Ertekin. É possível observar que os resultados entre o TFS e a formulação implícita do Ertekin possui erro máximo de 0.0025%. Esse erro é devido ao simulador TFS considerar a variação da porosidade com a pressão, enquanto o Ertekin considera a porosidade como constante. Como o TFS considera a variação da porosidade com a pressão é possível perceber que os resultados simulados no TFS são mais confiáveis que os apresentados em [Ertekin et al., 2001].

Tabela 7.1: Comparação entre os resultados obtidos pelo software TFS e os apresentados em [Ertekin et al., 2001]

360 dias						
	Delta t	Pressão [psi]				
		Bloco 1	Bloco 2	Bloco 3	Bloco 4	Bloco 5
Simulador TFS	30 dias	5323.82	5126.67	4716.98	4070.25	4345.62
Ertekin Explícito		5355.04	5146.14	4716.11	4051.12	4314.44
Simulador TFS	15 dias	5331.44	5131.26	4716.82	4065.55	4338.26
Ertekin Explícito		5347.07	5140.78	4716.58	4055.76	4322.66
Ertekin Implícito		5331.36	5131.18	4716.73	4065.45	4338.15
Simulador TFS	10 dias	5334.02	5132.82	4716.78	4063.96	4335.74
Ertekin Explícito		5344.41	5139.14	4716.60	4057.40	4325.31
Simulador TFS	5 dias	5336.61	5134.40	4716.75	4062.37	4333.20
Simulador TFS	1 dia	5338.70	5135.67	4716.73	4061.08	4331.15

A Figura 7.2 mostra a influência da discretização temporal nos resultados obtidos, onde as simulações foram comparadas com um passo de tempo de 1 dia. É possível observar que o simulador TFS se mostrou convergente do ponto de vista numérico para todas as células do domínio.

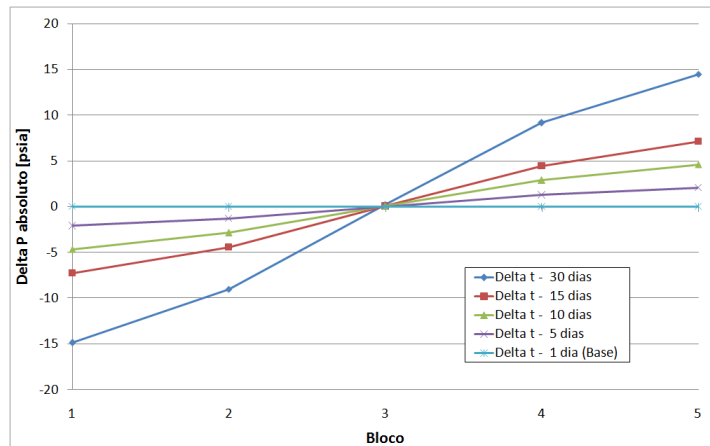


Figura 7.2: Comparação entre os valores obtidos para cada discretização temporal, mostrados em uma diferença absoluta de pressão para cada célula do domínio considerando uma simulação de 360 dias

## 7.2 Teste 2:

O software TFS foi comparado com o Problema 5.10 do livro [Ertekin et al., 2001], que está mostrado na Figura 7.3.



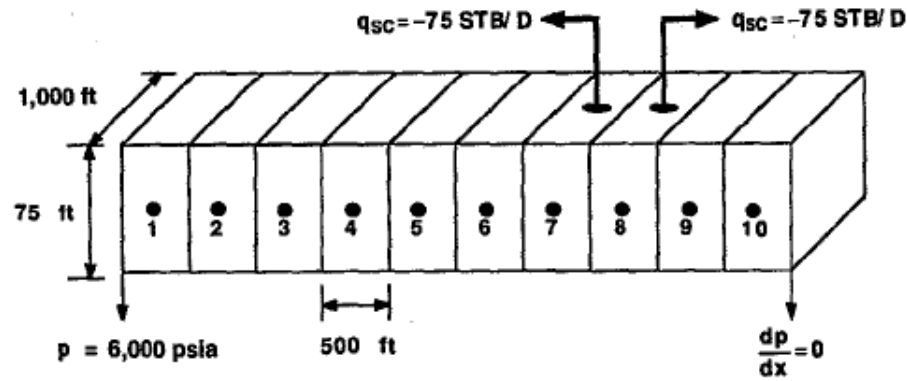


Figura 7.3: Reservatório do Exemplo 5.10. (Figura 5.28 [Ertekin et al., 2001])

Considerando um tempo total de simulação de 360 dias os resultados do software foram comparados com os resultados implícitos mostrados pelo Ertekin. A formulação explícita se mostrou instável no problema proposto. A Tabela 7.2 mostra a diferença entre os resultados encontrados para essa simulação.

Tabela 7.2: Comparação entre os resultados obtidos pelo TFS e os apresentados em [Ertekin et al., 2001]

360 dias											
	Delta t	Pressão [psi]									
		Bloco 1	Bloco 2	Bloco 3	Bloco 4	Bloco 5	Bloco 6	Bloco 7	Bloco 8	Bloco 9	Bloco 10
Simulador TFS	30 dias	6000.00	5777.83	5545.64	5293.68	5012.70	4694.21	4330.72	4211.74	4332.38	4393.55
Simulador TFS	15 dias	6000.00	5775.36	5540.75	5286.46	5003.30	4682.86	4317.72	4197.43	4317.18	4377.89
Ertekin Implícito		6000.00	5775.34	5540.71	5286.41	5003.24	4682.79	4317.64	4197.34	4317.08	4377.79
Simulador TFS	10 dias	6000.00	5774.52	5539.08	5284.00	5000.11	4679.01	4313.31	4192.59	4312.03	4372.58
Simulador TFS	5 dias	6000.00	5773.66	5537.39	5281.52	4996.89	4675.13	4308.87	4187.70	4306.84	4367.24
Simulador TFS	1 dia	6000.00	5772.97	5536.03	5279.51	4994.29	4672.00	4305.29	4183.77	4302.66	4362.93

A Figura 7.4 mostra os valores da diferença absoluta em psia para cada bloco entre os resultados obtidos pelo TFS e o método implícito do Ertekin.

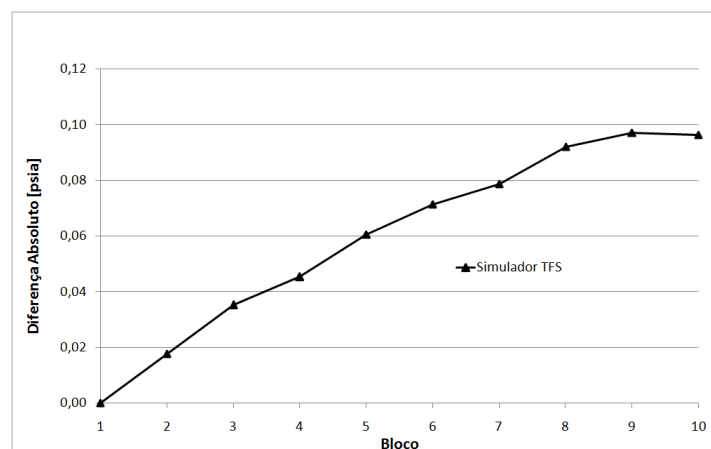


Figura 7.4: Diferença absoluta entre a pressão em cada célula obtida pelo simulador TFS e o método implícito apresentado em [Ertekin et al., 2001], considerando uma simulação de 360 dias

# Capítulo 8

## Documentação

A presente documentação refere-se ao uso do software TFS.

### 8.1 Interface gráfica inicial

A Figura 8.1 mostra uma captura de tela do TFS quando iniciado. É possível ver a presença de duas abas superiores que são utilizadas para se iniciar a simulação e para executar a simulação (*kernel*).

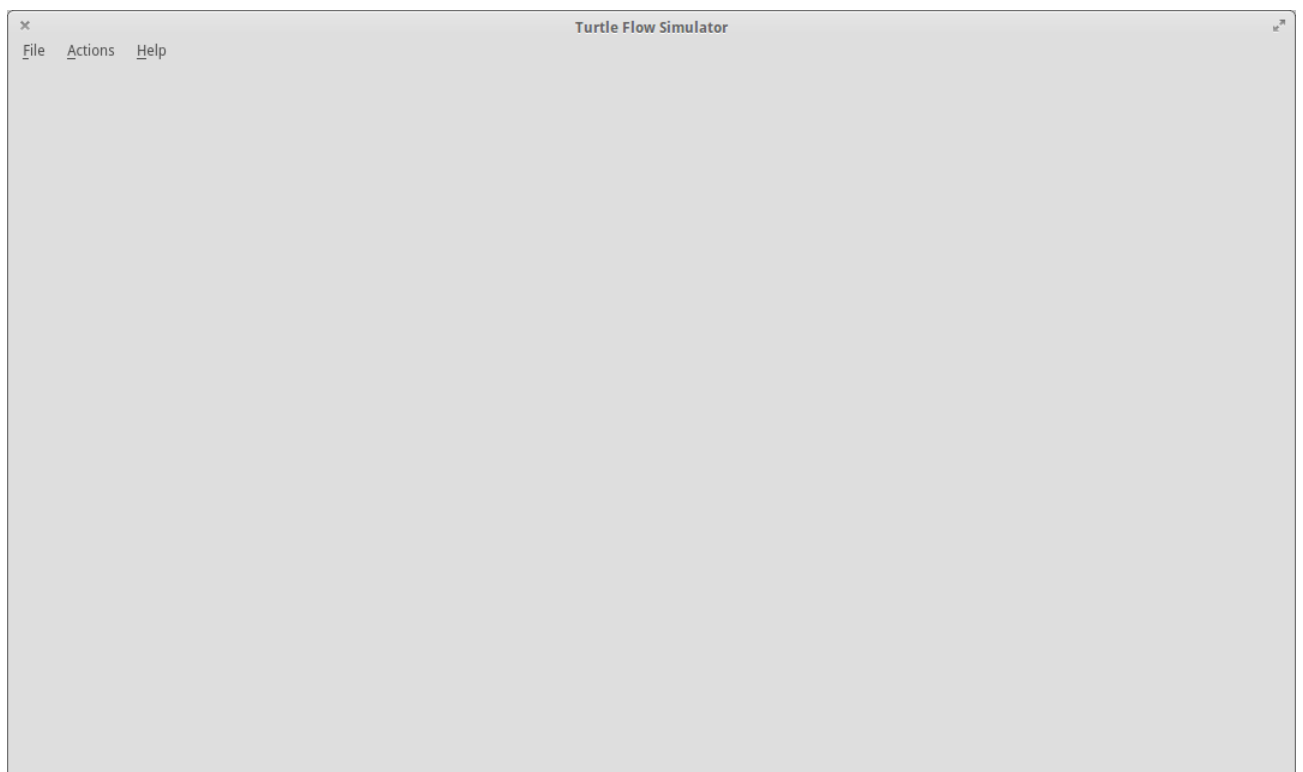


Figura 8.1: Interface inicial do TFS

## 8.2 Configurações iniciais

Clicando no menu arquivo (*File*) é possível ver a opção de configurações do simulador com as informações necessárias para iniciar a interface gráfica. Um diálogo como na Figura 8.2 se abrirá e será possível escolher as primeiras informações a serem preenchidas sobre a simulação.

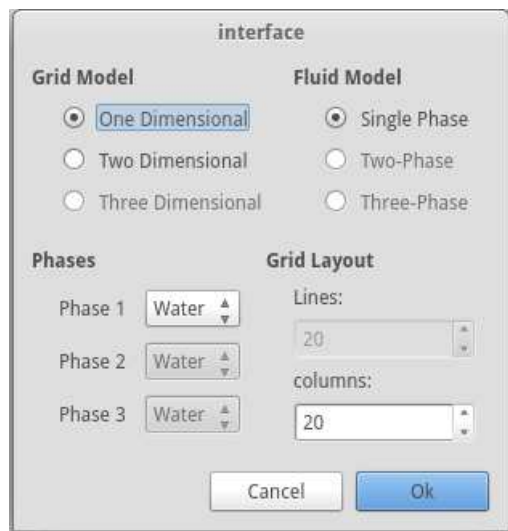


Figura 8.2: Janela com as configurações iniciais da simulação

O modelo do grid (*Grid model*) poderá ser de uma dimensão (*One-Dimensional*), duas dimensões (*Two-Dimensional*) ou Três dimensões (*Three-Dimensional*). O grid em duas e três dimensões não se encontram disponíveis nessa etapa de desenvolvimento do software, devendo então ser escolhida a opção de uma dimensão.

Agora é possível se escolher a configuração do grid, determinando quantas colunas (*Columns*) o grid terá. Ainda no mesmo diálogo pode-se definir o modelo de fluido que será utilizado, uma fase (*Single-Phase*), duas fases (*Two-Phase*) ou três fases (*Three-Phase*). Atualmente a única opção disponível é a de uma fase.

Para finalizar as configurações iniciais do modelo é necessário se escolher o tipo do fluido de cada fase, como só tem a opção de uma única fase implementada, só poderá modificar o tipo do fluido para a fase 1 (*Phase 1*), este poderá ser Água (*Water*), Óleo (*Oil*) ou Gás (*Gas*).

Após configuradas as informações iniciais, o botão Ok avança para a próxima etapa da simulação.

## 8.3 Determinando os valores da simulação

A Figura 8.3 mostra a tela apresentada após feita as configurações iniciais.

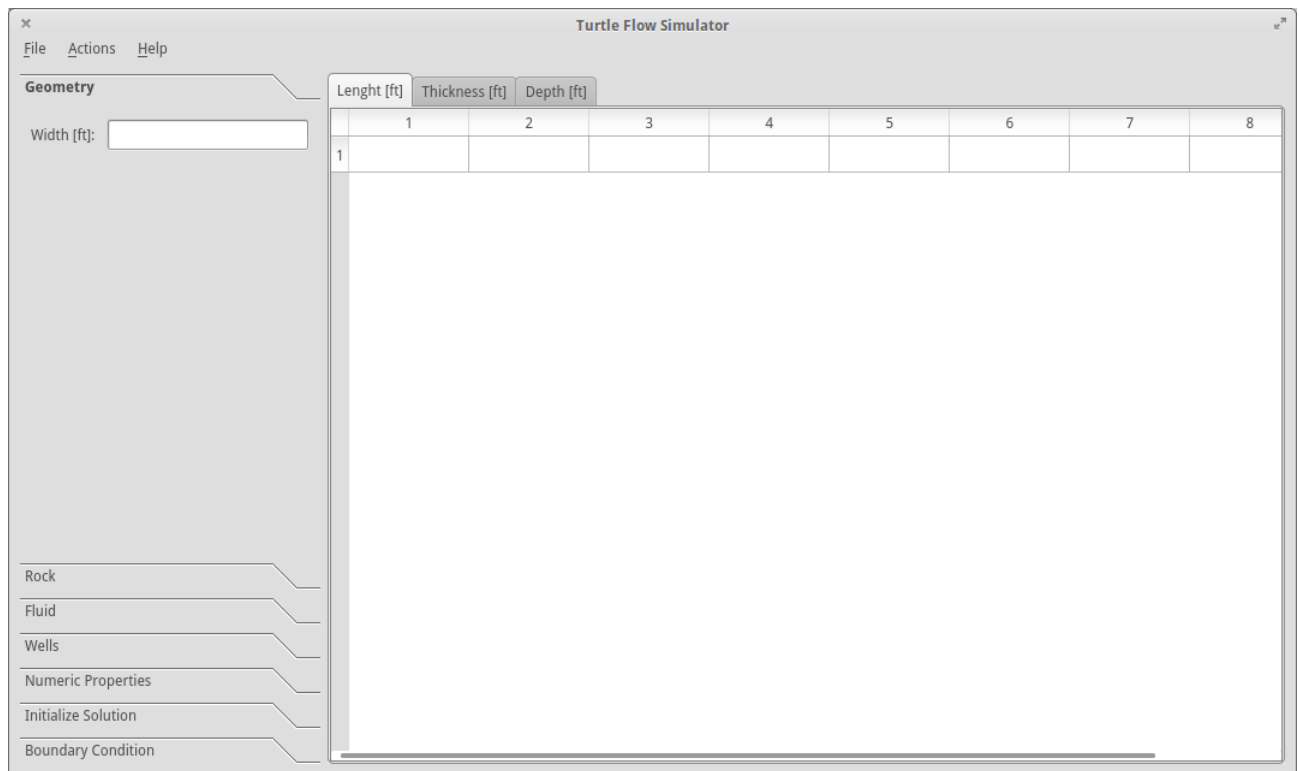


Figura 8.3: Tela apresentada após as configurações iniciais

A interface é composta por uma série de guias verticais localizadas a esquerda, e abas horizontais localizadas na parte superior da janela. Um conjunto de grids representam os valores correspondentes a aba ativa.

As guias representam um fluxo de informação na sequência em que devem ser preenchidas, portanto inicia-se a simulação na primeira guia, Geometria (*Geometry*) e finaliza o preenchimento na guia Condições de contorno (*Boundary Condition*).

### 8.3.1 Guia “Geometry”

A guia geometria (Figura 8.3) tem um único valor a ser preenchido correspondente a largura das células (*Width*). Esse valor é preenchido conforme os dados da simulação. Essa guia possui três abas horizontais, sendo respectivamente, comprimento (*Lenght*), espessura (*Thickness*) e profundidade (*Depth*) relativos a cada célula.

### 8.3.2 Guia “Rock”

A Figura 8.4 mostra a guia *Rock* aberta.

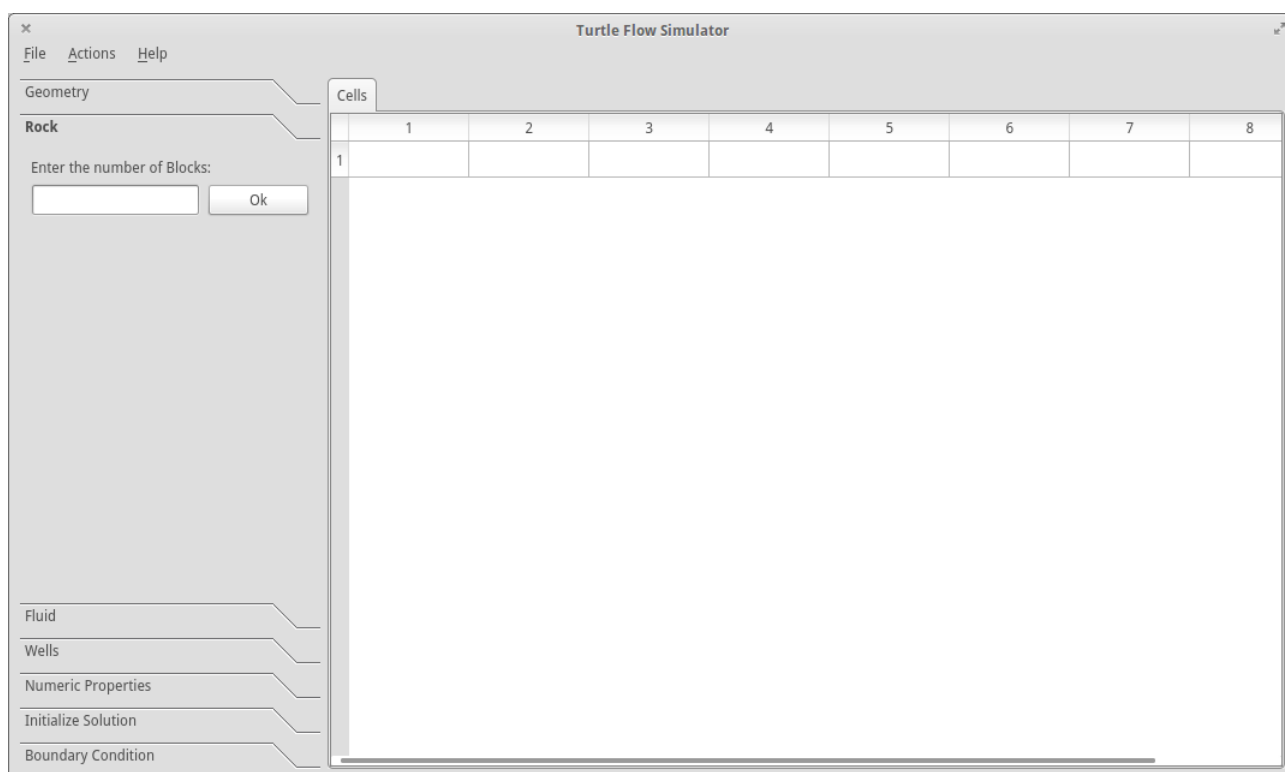


Figura 8.4: Aba Rock ativada

O primeiro dado que deve ser fornecido é o número de blocos (conjunto de células com propriedades de rochas iguais) existente na simulação. Após clicar em Ok essa guia é modificada como mostrada na Figura 8.5.

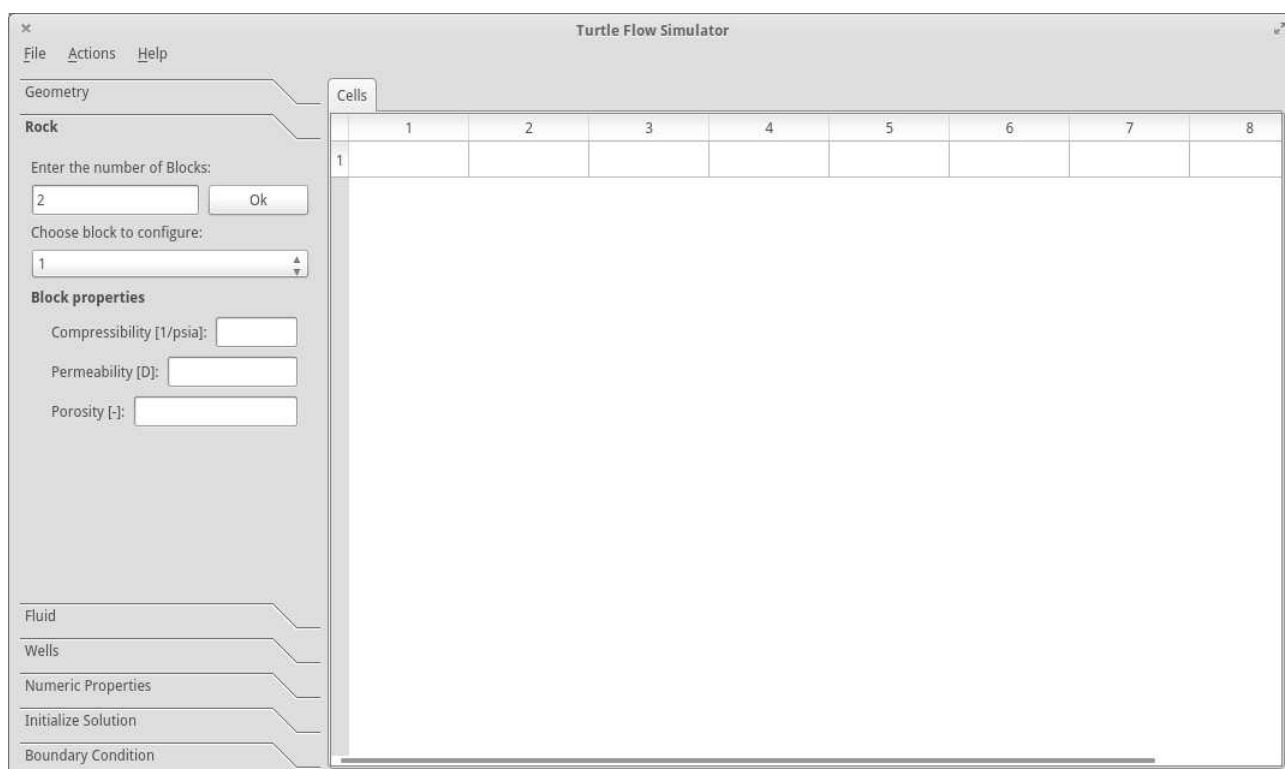


Figura 8.5: Aba Rock após a definição do número de blocos do sistema

Conforme visto na Figura 8.5, novas opções apareceram para o preenchimento das propriedades de rocha do problema. O seletor (*combo box*) que apareceu na parte superior da guia *rock* identifica o bloco ao qual está sendo configurado. As propriedades de cada bloco são: compressibilidade (*compressibility*), permeabilidade (*permeability*) e porosidade (*porosity*). Esses campos devem ser preenchidos para cada bloco.

A correlação entre célula e bloco é feita no grid identificado pela aba *Células (Cells)*. Na Figura 8.5 pode ver que o bloco 1 está relacionado com a célula 2 enquanto o bloco 2 está relacionado com a célula 6. Todas as células devem possuir um número indicando o bloco ao qual ela faz parte<sup>1</sup>.

### 8.3.3 Guia “Fluid”

Nesta guia é possível se configurar as propriedades do fluido conforme visto na Figura 8.6.

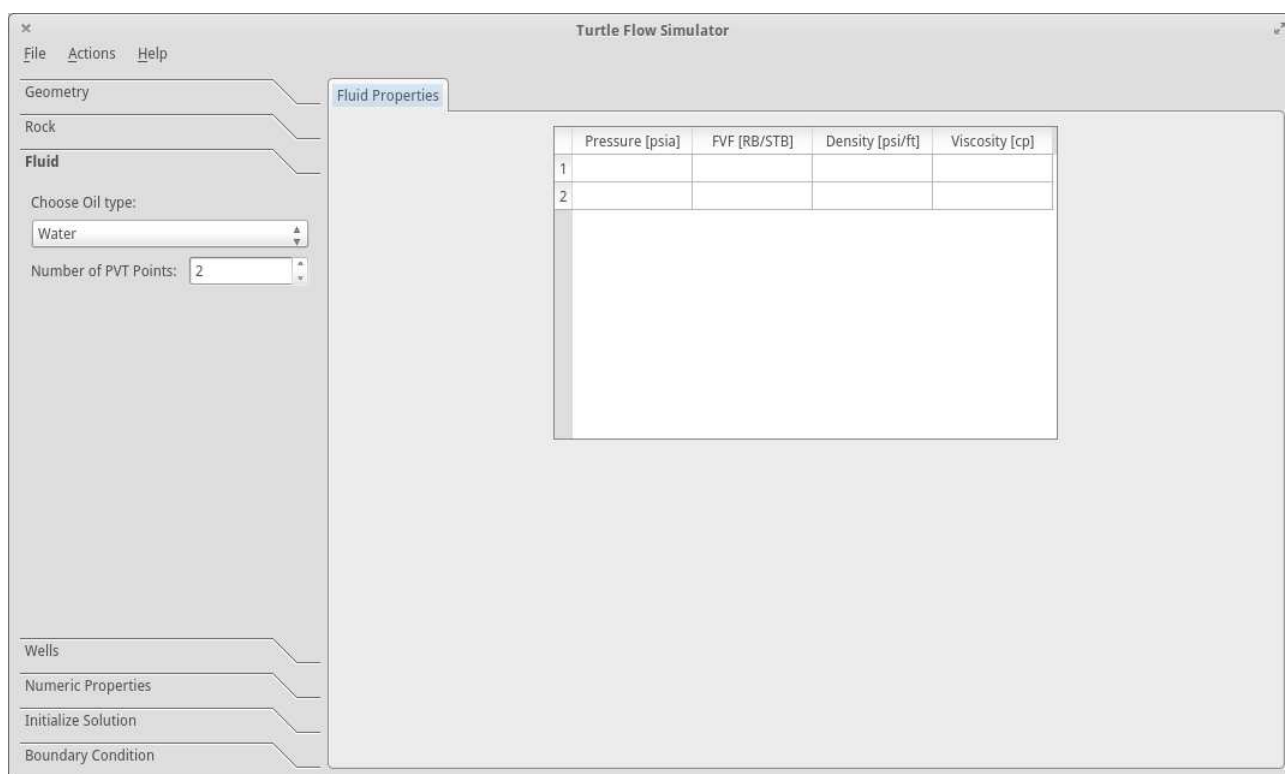


Figura 8.6: Aba Fluid ativada

Utiliza-se o seletor para se escolher o tipo de fluido a configurar e em seguida determina-se o número de pontos PVT para esse fluido. No grid central é fornecido os valores de pressão (*Pressure*), FVF, densidade (*Density*) e viscosidade (*Viscosity*) para cada ponto obtido no ensaio PVT.

<sup>1</sup>O Bloco foi criado com o intuito de se agrupar células com as mesmas propriedades de rocha, diminuindo a repetição dessa informação em cada célula. Assim é possível, e desejável, que mais de uma célula esteja relacionada com o mesmo bloco.

### 8.3.4 Guia “Wells”

Na guia *wells* (Figura 8.7) é possível se configurar os poços existentes no reservatório. O primeiro dado a ser fornecido é o número de poços a serem configurados. Cada poço necessita o Id da célula onde o mesmo está localizado, o tipo de fluido, e a vazão do poço, sendo que vazões negativas indicam poços produtores e vazões positivas indicam injetores.

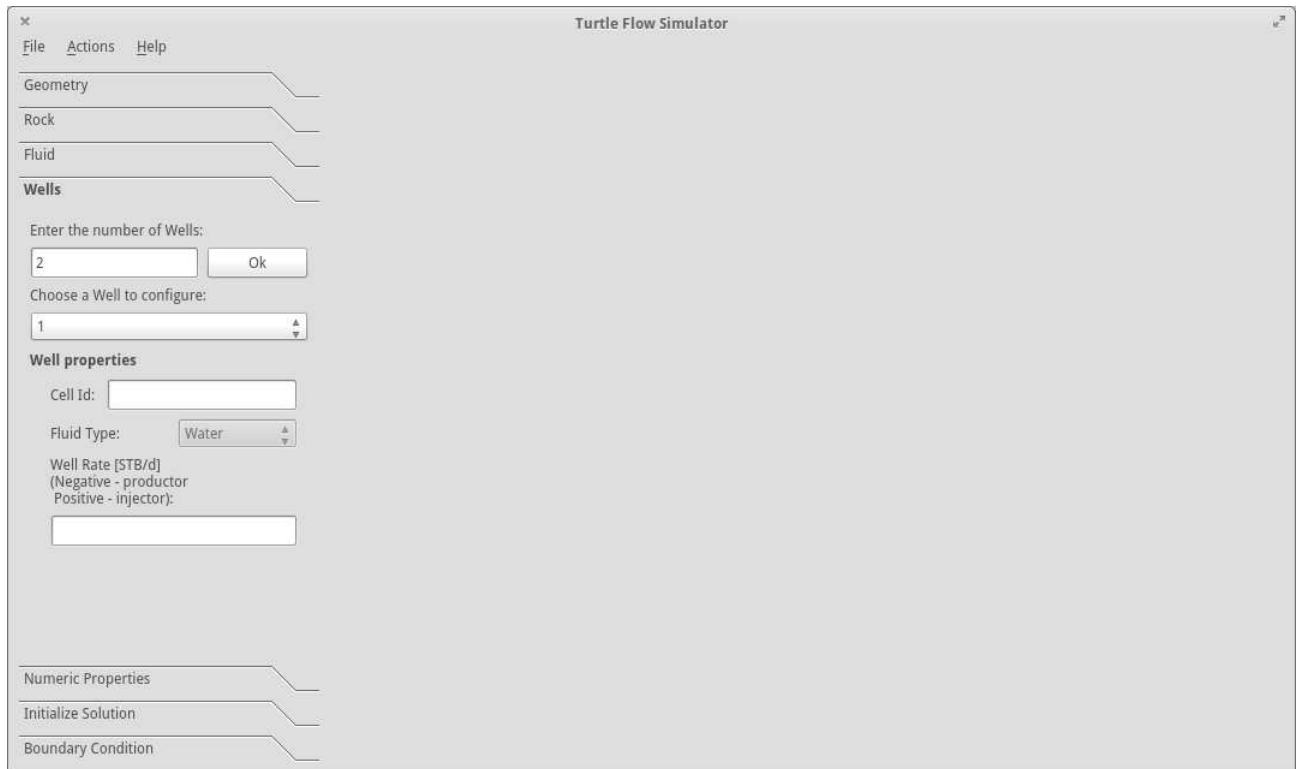


Figura 8.7: Aba Wells ativada

### 8.3.5 Guia “Numeric Properties”

A guia *numeric properties* (Figura 8.8) é utilizada para se fornecer os dados numéricos para o simulador, esses dados são: o tipo de solver (somente o Gauss-Seidel está disponível), número de iterações (*Iterations*), precisão do solver (*Solver Precision*), o passo de tempo da simulação ( $\Delta t$ ), tempo final da simulação (*Final simulation time*), número máximo de iterações para a linearização (*Linear Max Iteractions*) e a precisão relativa da linearização (*Thresholding Relative Precision*). Essa guia não possui um *widget* central, visto que manipula somente dados numéricos.

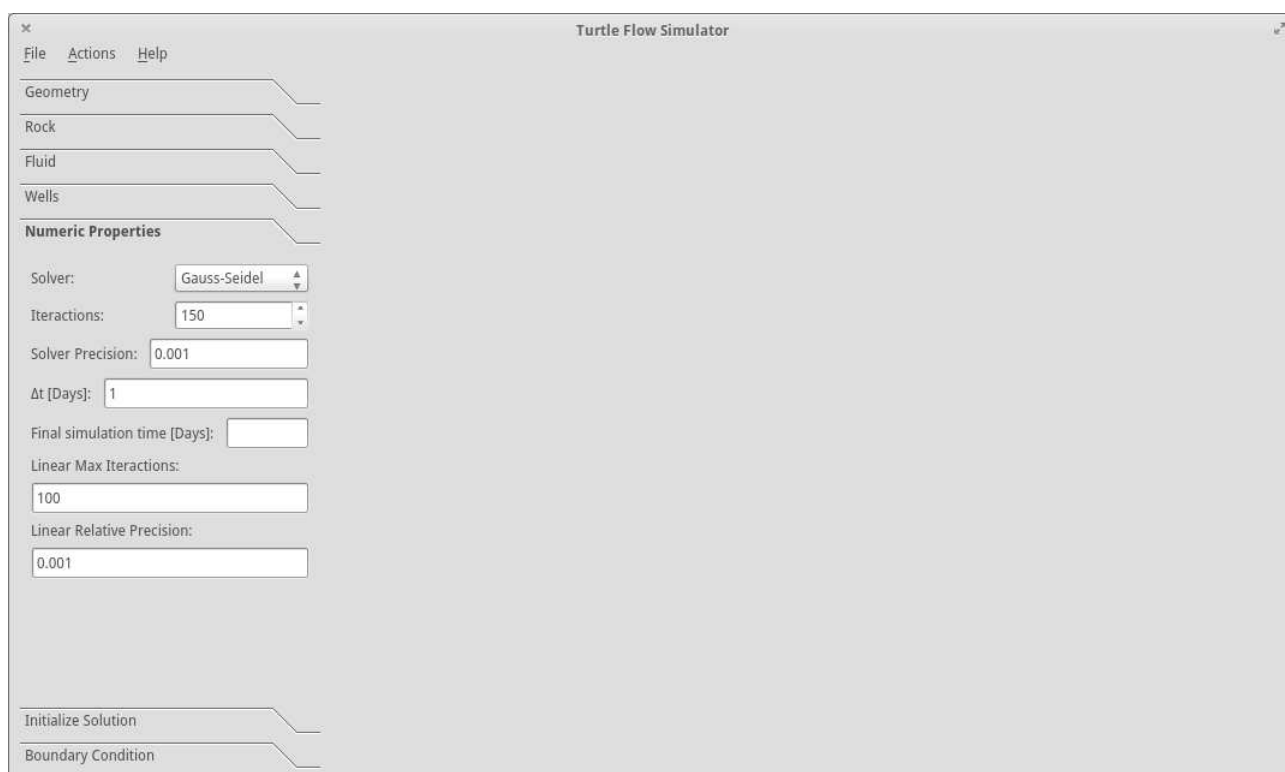


Figura 8.8: Aba Numeric Properties ativada

### 8.3.6 Guia “Initialize Solution”

A Figura 8.9 mostra a guia *Initialize Solution*. Nessa guia são fornecidos os dados de pressão de referência (*Reference Pressure*) e profundidade de referência (*Reference Depth*) para que o simulador possa calcular a hidrostática inicial do reservatório.



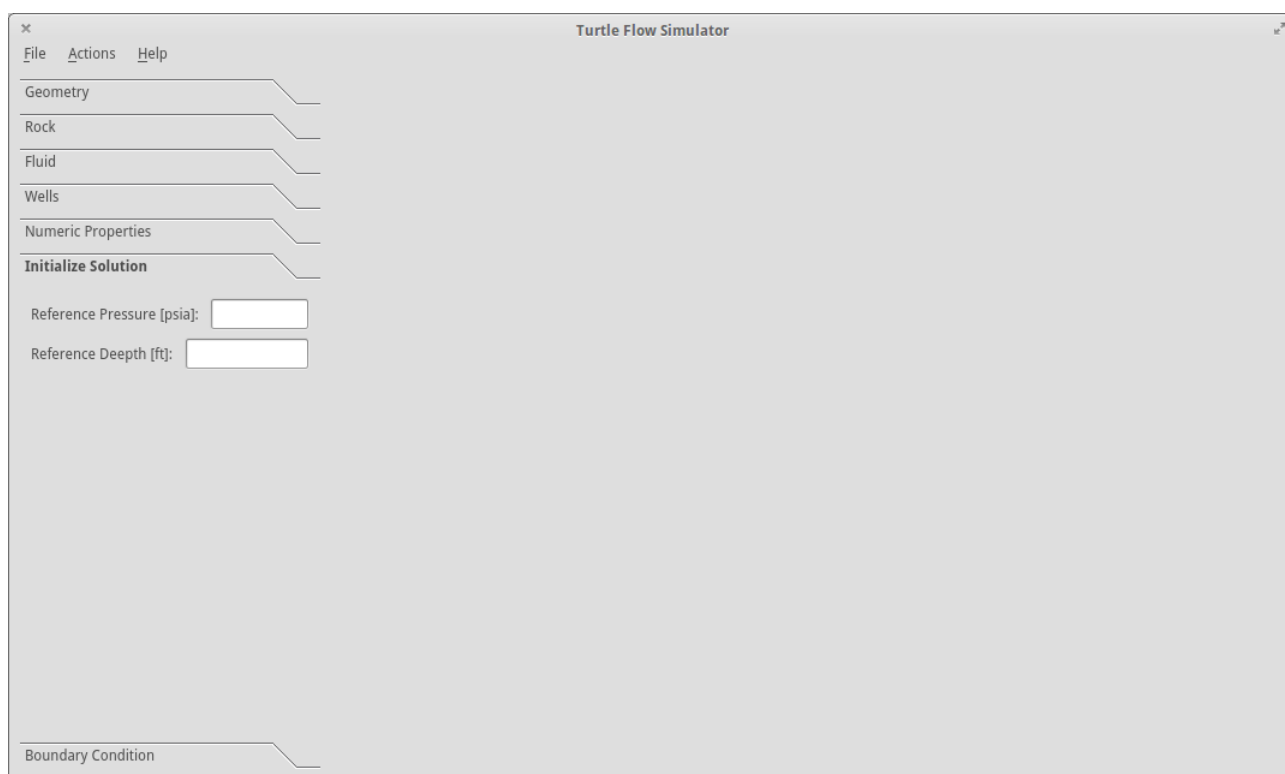


Figura 8.9: Aba Initialize Solution

### 8.3.7 Guia “Boundary Condition”

A Figura 8.10 mostra a guia *Boundary Condition*, onde são fornecidos os dados a respeito das condições de contorno do problema a ser resolvido pelo simulador.

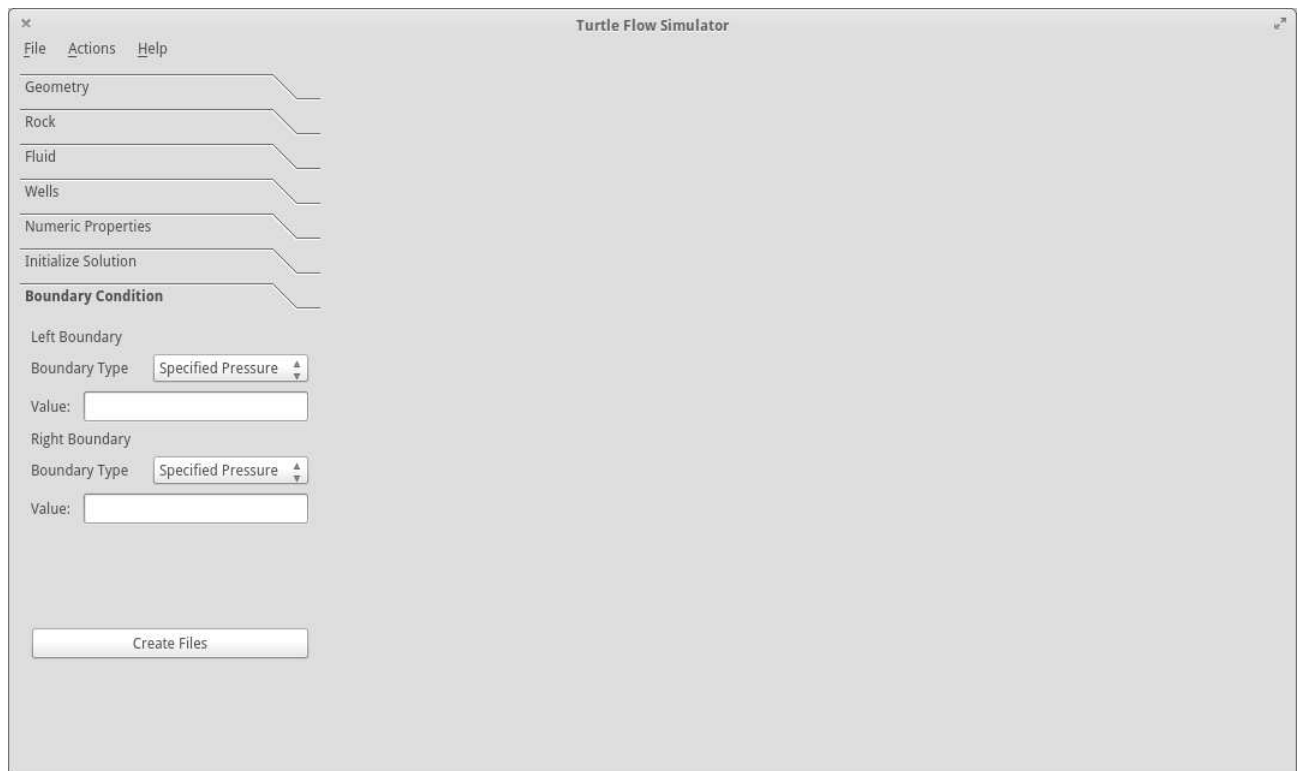


Figura 8.10: Aba Boundary Condition

É possível se fornecer diferentes condições de contorno para a fronteira esquerda (*Left Boundary*) e para a fronteira direita (*Right Boundary*) do problema. Os dados fornecidos são o tipo de fronteira (*Boundary Type*) que pode assumir os seguintes valores, fluxo específico (*Specified Flow Rate*), sistema fechado (*Closed System*) e pressão constante na fronteira (*Specified Pressure*).

Após o preenchimento de todas as guias com as propriedades da simulação, é necessário a criação dos arquivos de configuração. O botão criar arquivos (*Create Files*) na guia *boundary conditions* é utilizado para se criar esses arquivos. Após a geração dos arquivos de simulação, abrirá uma janela informando que os arquivos foram devidamente criados (Figura 8.11).

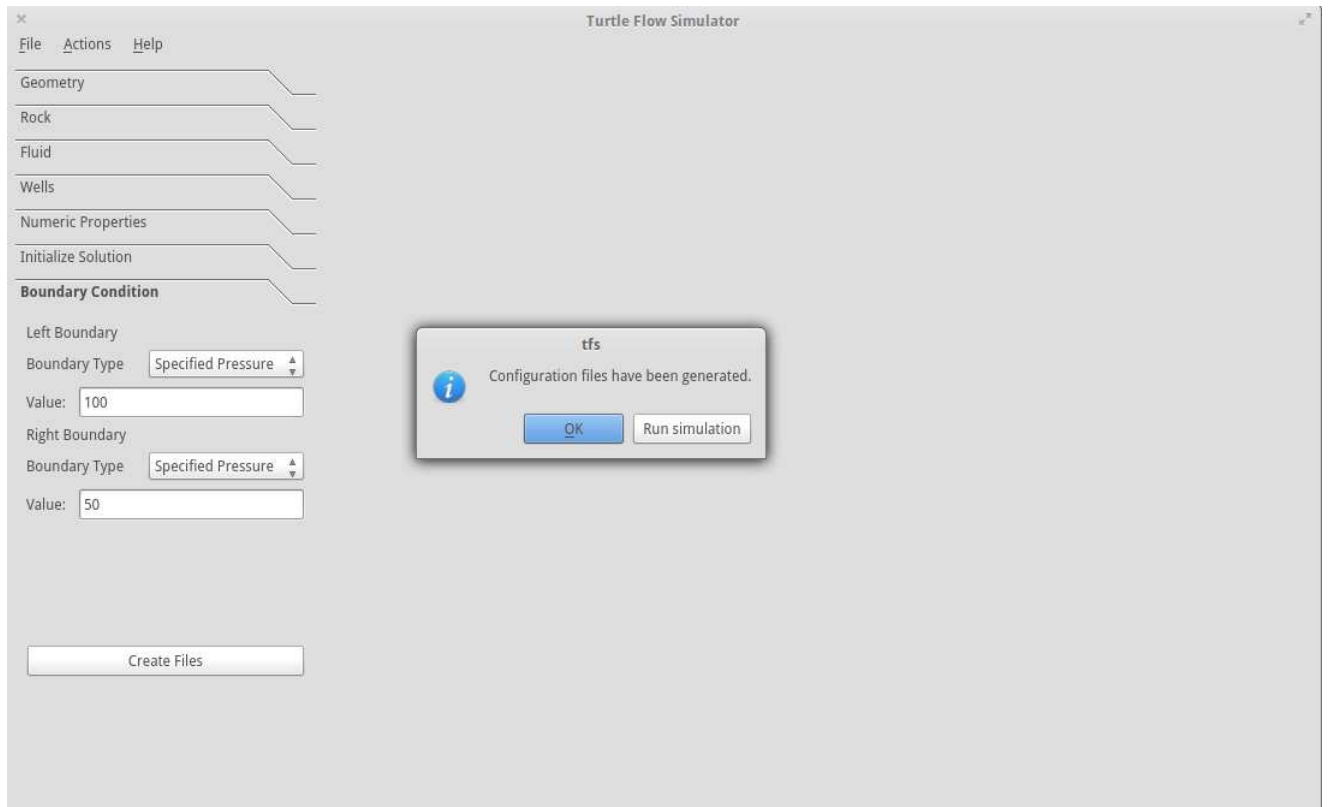


Figura 8.11: Janela mostrando que os arquivos da simulação foram gerados com êxito

## 8.4 Executando a simulação

A opção rodar simulação (*Run Simulation*) localizada no menu Ações (*Actions*) é utilizada para se iniciar o *kernel* numérico, conforme visto na Figura 8.12.

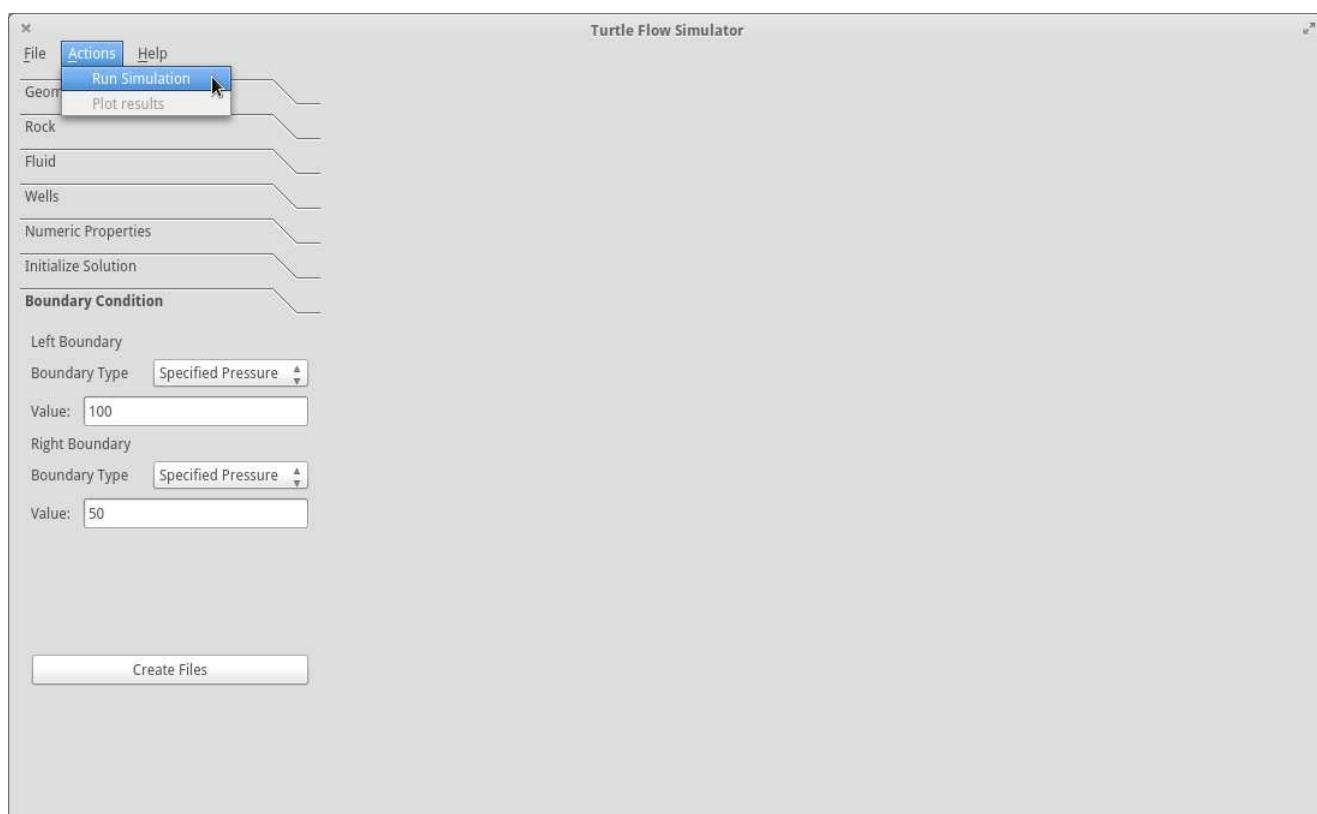


Figura 8.12: Opção rodar simulação do menu ações

Abrirá uma janela informando que a simulação será executada (Figura 8.13) e após isso se abrirá um terminal na tela com o fluxo de execução da simulação (Figura 8.14).

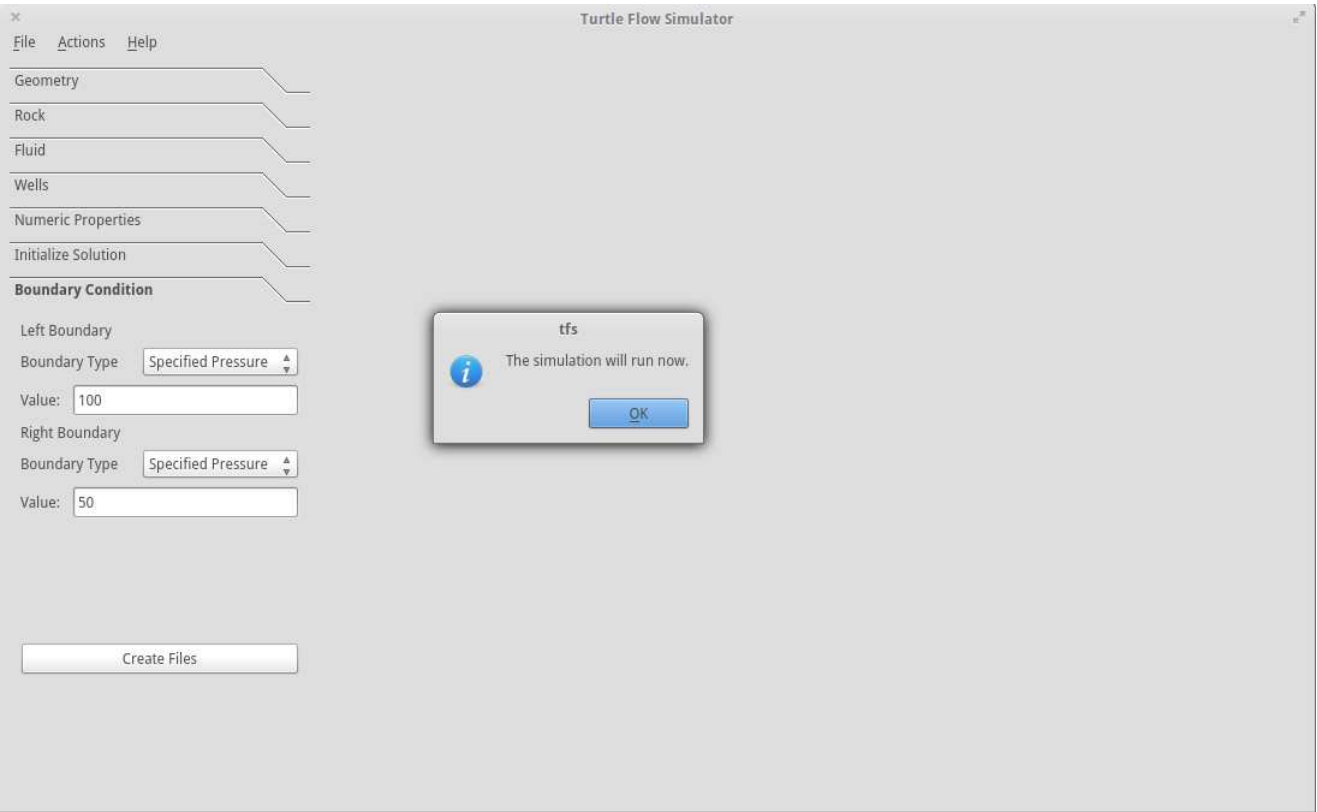


Figura 8.13: Janela informando o início da simulação

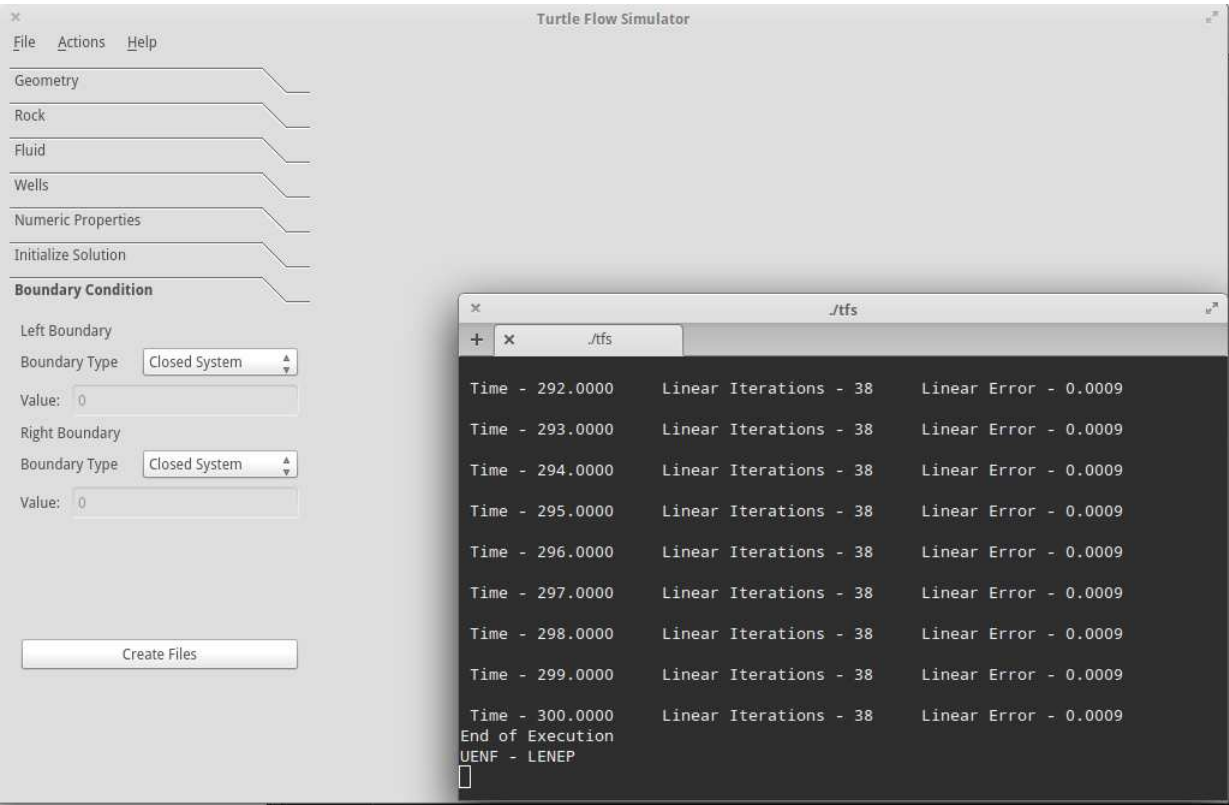


Figura 8.14: Simulação de reservatórios em execução

## 8.5 Plotando gráficos

Após a simulação ser executada é possível a geração de gráficos mostrando os resultados da simulação, clicando na opção plotar resultados (*Plot Results*) localizada no menu ações (*Actions*). Uma janela se abrirá pedindo informações sobre o tipo de gráfico que o usuário deseja gerar (Figura 8.15).

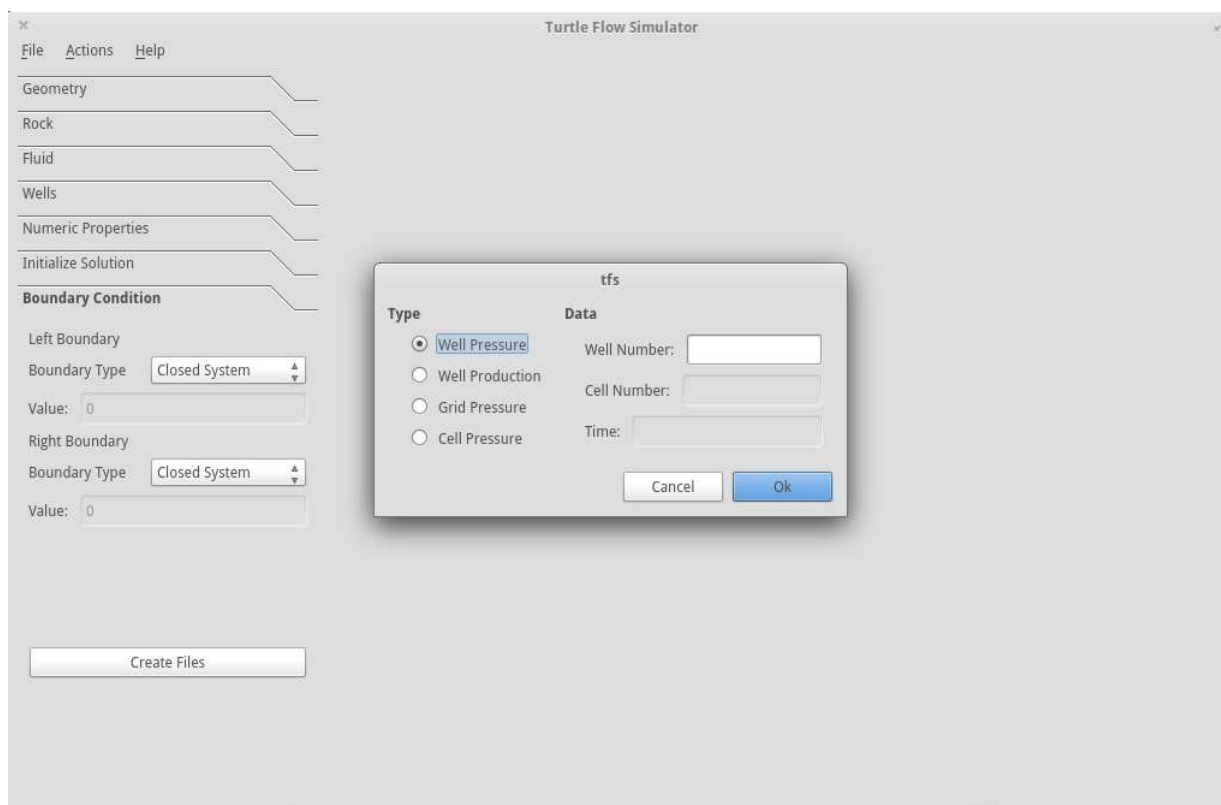


Figura 8.15: Escolha do tipo do gráfico resultado

Após a confirmação, uma janela com o gráfico plotado aparecerá na tela, como pode ser visto na Figura 8.16.

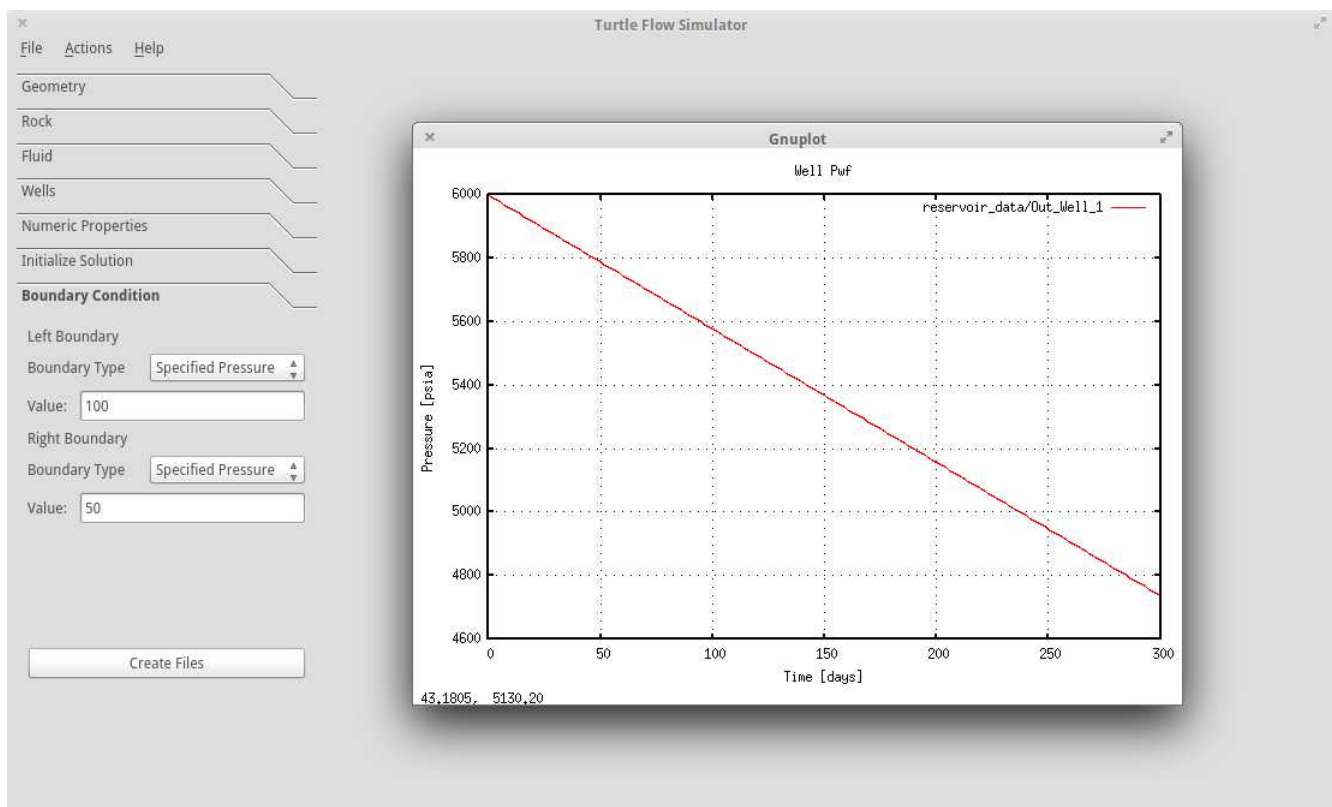


Figura 8.16: Exemplo de gráfico de pressão no fundo do poço, gerado após simulação

Se desejável é possível se acessar a pasta *kernel* situada no diretório do programa para se acessar diretamente os arquivos com os resultados da simulação.

# Capítulo 9

## Manual do desenvolvedor

O TFS é um simulador de reservatório capaz de simular o comportamento de um reservatório de uma dimensão e uma fase. O software apresenta uma interface gráfica através da qual o usuário pode configurar os parâmetros da simulação. Terminada a simulação, a interface apresenta os resultados através de um conjunto de gráficos.

### 9.1 Dependências do software

Para compilar e rodar o Simulador TFS é necessário um sistema Windows ou GNU/Linux com os seguintes softwares instalados:

1. Qt 5.0 ou maior com bibliotecas de desenvolvimento instaladas (principalmente a ferramenta qmake);
2. C++11 instalado no sistema;
3. Gnuplot 4.4 ou maior (<http://www.gnuplot.info>);

### 9.2 Compilando o software

Existem duas maneiras de se compilar software no sistema GNU/Linux:

1. Executando o script `build_all.sh` a partir da pasta raiz do projeto.
2. Executando os seguintes comandos abaixo na ordem especificada.
  - `$ qmake`
  - `$ make -f Makefile`
  - `$ cd ./src/kernel`
  - `$ make -f Makefile`



No sistema Windows, devem ser executados os seguintes procedimentos:

1. abrir o arquivo `tfs.pro` no Qt Creator;
2. executar o comando `build` para compilar a interface;
3. abrir o arquivo `Makefile.win` em uma IDE como o DevC++;
4. executar o `build` para compilar o kernel.

Os arquivos executáveis serão gerados na pasta `bin` a partir da raiz do projeto.

## 9.3 Executando

O software deve ser executado a partir da pasta `bin/` no diretório raiz do projeto. Execute os comandos de terminal exemplificados abaixo para o sistema GNU/Linux:

- `$ cd ./bin/`
- `$ ./tfs`



# Referências Bibliográficas

- [Ertekin et al., 2001] Ertekin, T., Abou-Kassem, J., and King, G. (2001). *Basic Applied Reservoir Simulation*. SPE Textbook Series. Henry L. Doherty Memorial Fund of AIME, Society of Petroleum Engineers. 1, 35, 36, 37, 60
- [Júnior, 2013] Júnior, B. G. S. (2013). Simulador numérico bidimensional para escoamento monofásico em meios porosos. Master's thesis, Universidade estadual do norte fluminense. 60

# Apêndice A

## Arquivos de inicialização da simulação

Estão descritas nesse apêndice o padrão de escrita utilizada nos arquivos de inicialização da simulação.

### A.1 ArqBlock.in

Esse arquivo descreve as propriedades de rocha utilizadas na simulação.

Listing A.1: Arquivo ArqBlock.in

```
1      --- [-] Identificador do Bloco
6e-6   --- [-] Compressibilidade da Rocha no bloco 1
0.1    --- [Darcy] Permeabilidade no bloco 1
0.2    --- [-] Porosidade no bloco 1
2000   --- [psi] Pressão de Referência da Porosidade

2      --- [-] Identificador do Bloco
5e-6   --- [-] Compressibilidade da Rocha no bloco 1
0.08   --- [Darcy] Permeabilidade no bloco 1
0.3    --- [-] Porosidade no bloco 1
2100   --- [psi] Pressão de Referência da Porosidade
```

### A.2 ArqBoundCond1d.in

Esse arquivo descreve as condições de contorno utilizadas na simulação do problema unidimensional.

Listing A.2: Arquivo ArqBoundCond1d.in

```

1      --- [-] Tipo condição esquerda (0 - Vazão Constante, 1 - Pressão
      Constante)
6000   --- [STB/d ou Psia] Valor utilizado para a condição esquerda

0      --- [-] Tipo condição direita (0 - Vazão Constante, 1 - Pressão
      Constante)
0      --- [STB/d ou Psia] Valor utilizado para a condição direita

```

### A.3 ArqCell1d.in

Esse arquivo descreve as propriedades armazenadas em uma célula unidimensional.

Listing A.3: Arquivo ArqCell1d.in

```

1      --- [-] Identificador da Célula 1
50     --- [ft] Espessura da Célula 1
1500   --- [ft] Profundidade da Célula 1
1      --- [-] Identificador do Bloco conectado a Célula

2      --- [-] Identificador da Célula 2
30     --- [ft] Espessura da Célula 2
1600   --- [ft] Profundidade da Célula 2
1      --- [-] Identificador do Bloco conectado a Célula

3      --- [-] Identificador da Célula 3
40     --- [ft] Espessura da Célula 3
1500   --- [ft] Profundidade da Célula 3
2      --- [-] Identificador do Bloco conectado a Célula

```

### A.4 ArqGas.in, ArqOil.in e ArqWater.in

Os arquivos para gás, óleo e água possuem o mesmo padrão de escrita, mudando apenas o nome do arquivo.

Listing A.4: Arquivo ArqGas.in, ArqOil.in ou ArqWater.in

```

5 --- Pontos de Pressão PVT
pressão      FVF      Peso Específico      Viscosidade
[psia]      [RB/bbl]      [psi/ft]      [cp]
1000        1.02827      0.43214      0.01
2000        1.02224      0.43342      0.01
3000        1.01621      0.43599      0.01
4000        1.01024      0.43856      0.01
5000        1.00506      0.44113      0.01

```

## A.5 ArqInit1d.in

Esse arquivo possui as configurações para se inicializar a pressão em um reservatório unidimensional.

Listing A.5: Arquivo ArqInit1d.in

```
3000    --- [psi] Pressão de Referência
1500    --- [ft]  Profundidade de Referência
```

## A.6 ArqNumeric.in

Esse arquivo possui as configurações numéricas para a solução do problema.

Listing A.6: Arquivo ArqNumeric.in

```
1      --- [-] Identificador do Solver (1 = Gauss-Seidel);
30     --- [-] Número de Iterações máximas do solver numérico;
0.001  --- [-] Precisão do solver numérico;

0.1    --- [dias] delta t do problema;
10     --- [dias] tempo final de simulação;
50     --- [-] Número de Iterações máximas da linearização ni;
0.001  --- [-] Precisão relativa da linearização ni.
```

## A.7 ArqRes1d.in

Arquivo de configuração da geometria do reservatório.

Listing A.7: Arquivo ArqRes1d.in

```
3      --- [-] Número de Blocos;
3      --- [-] Número de Células;
80     --- [ft] Largura das Células;
1      --- [-] Tipo de Fluido (1- Água, 2- Óleo, 3- Gás)

380    --- [ft] Comprimento da Célula 1
200    --- [ft] Comprimento da Célula 2
250    --- [ft] Comprimento da Célula 3
```

## A.8 ArqWell1d.in

Arquivo com as configurações dos poços do problema.

## Listing A.8: Arquivo ArqWell1d.in

```
2    --- [-] Número de Poços;

2    --- [-] Identificador da Célula que contém o poço;
1    --- [-] Identificador do Poço;
1    --- [-] Tipo de Fluido (1- Água; 2- Óleo; 3- Gás)
-20   --- [STB/D] Vazão do Poço. (Positivo = Injeção // Negativo =
      produção)

2    --- [-] Identificador da Célula que contém o poço;
2    --- [-] Identificador do Poço;
1    --- [-] Tipo de Fluido (1- Água; 2- Óleo; 3- Gás)
10   --- [STB/D] Vazão do Poço. (Positivo = Injeção // Negativo =
      produção)
```

# Apêndice B

## Sistema de unidades do TFS

A Tabela B.1 mostra o sistema de variáveis adotadas no software TFS. Esse sistema é o padrão OilField internacional utilizado por [Ertekin et al., 2001].

Tabela B.1: Sistema de unidades utilizadas no simulador TFS (Fonte: [Júnior, 2013])

Quantidade	Simulador	SI	Fator de Conversão
Comprimento	ft	m	0,3048
Área	ft <sup>2</sup>	m <sup>2</sup>	0,09290304
Permeabilidade	darcy	$\mu\text{m}^2$	0,9869233
Viscosidade	cp	Pa.s	0,001
FVF do gás	RB/scf	m <sup>3</sup> /std m <sup>3</sup>	5,5519314
FVF do líquido	RB/STB	m <sup>3</sup> /std m <sup>3</sup>	1,0
Razão Gás/Óleo	scf/STB	std m <sup>3</sup> /std m <sup>3</sup>	0,1801175
Pressão	psia	kPa	6,894757
Gradiente de pressão	psi/ft	kPa/m	22,62059
Peso específico	psi/ft	kPa/m	22,62059
Vazão do gás	scf/D	std m <sup>3</sup> /d	0,02863640
Vazão do líquido	STB/D	std m <sup>3</sup> /d	0,1589873
Velocidade volumétrica	RB/(D.ft <sup>2</sup> )	m <sup>3</sup> /(d.m <sup>3</sup> )	1,7103717
Volume do bloco	ft <sup>3</sup>	m <sup>3</sup>	0,02831685
Massa específica	lbm/ft <sup>3</sup>	kg/m <sup>3</sup>	16,01846
Aceleração da gravidade	32,174 ft/s <sup>2</sup>	9,8066352 m/s <sup>2</sup>	0,3048
Compressibilidade	psi <sup>-1</sup>	kPa <sup>-1</sup>	0,1450377
Temperatura Absoluta	°R	K	0,55555556
Permeabilidade relativa	-	-	1,0
Porosidade	-	-	1,0
Saturação	-	-	1,0
Fator de compressibilidade	-	-	1,0
Tempo	dia	dia	1,0
Ângulo	rad	rad	1,0
$\alpha_c$	5,614583	1	-
$\beta_c$	1,127	$86,4 \times 10^{-6}$	-
$\gamma_c$	$0,21584 \times 10^{-3}$	$10^{-3}$	-

\* Multiplica-se o sistema americano pelo fator de conversão para encontrar o sistema internacional.

\*\* STB e scf são medidos a 60°F e 14,696 psia; std m<sup>3</sup> é medido a 15°C e 100kPa. Adaptada de Ertekin *et al.* (2001).



# Apêndice C

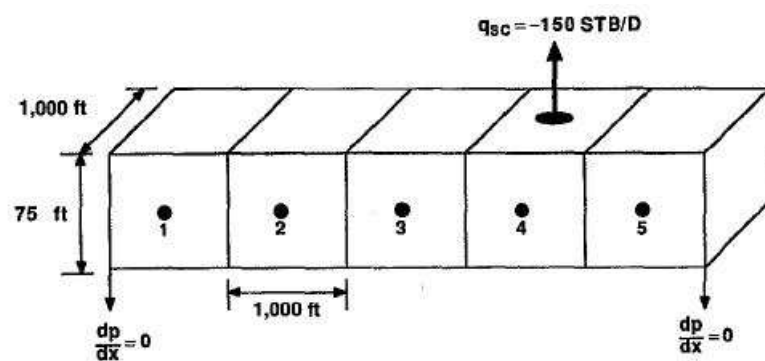
## Tutoriais

Estão descritos nessa seção os seguintes tutoriais de utilização do TFS:

- Reservatório horizontal unidimensional homogêneo e fechado;
- Reservatório anticlinal com duas litologias e alimentação de pressão.

### C.1 Reservatório horizontal unidimensional homogêneo e fechado

A Figura C.1 mostra um esboço do reservatório estudado. É possível observar que as propriedades do óleo são constantes, assim como as de rocha para todos os blocos.



#### Dados do Reservatório

- \*Porosidade: 0.18
- \*Permeabilidade: 0.15 [D]
- \*Pressão Inicial: 6000 [psia]
- \*Compressibilidade:  $3.3 \times 10^{-6}$  [1/psi]

#### Dados do Óleo:

- \*Viscosidade: 10 [cp];
- \*FVF: 1 [RB/STB];

Figura C.1: Reservatório horizontal e homogêneo

Para se modelar esse problema abra o TFS, clique em File->“New Simulation” e preencha as propriedades iniciais do problema como mostrado na Figura C.2. Observa-se que esse exemplo é formado por cinco blocos horizontais.

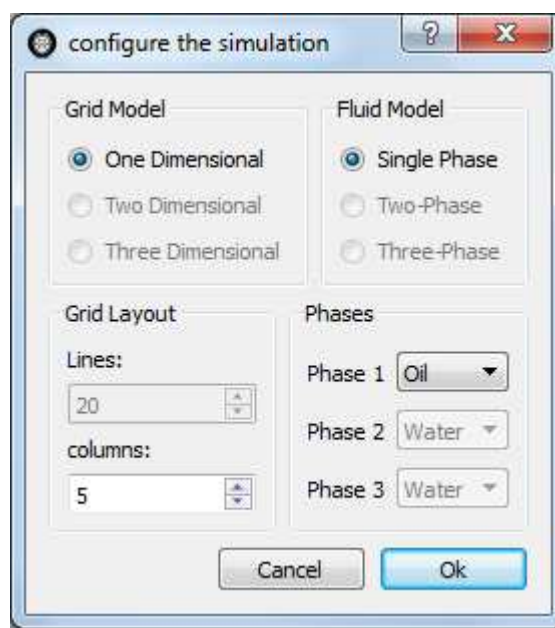


Figura C.2: Propriedades iniciais do exemplo unidimensional homogêneo

Após clicar em Ok é necessário se preencher as propriedade iniciais do problema. Na aba vertical geometria preencha o campo *Width* (Largura) com o valor de 1000. Preencha o valor de 1000 para cada célula criada na aba *length* (Comprimento). A Figura C.3 mostra como deverá ficar essas abas após o preenchimento.

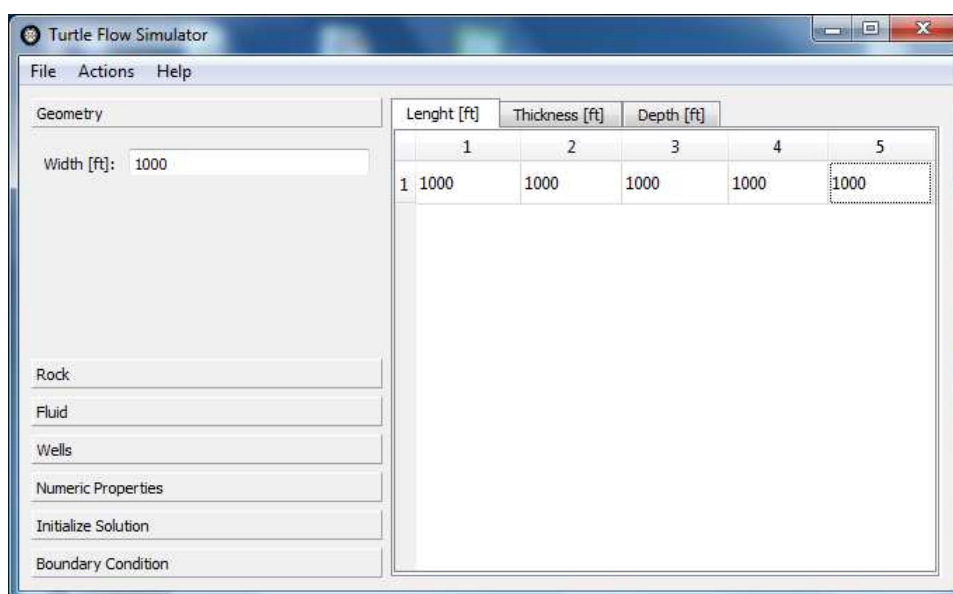


Figura C.3: Configurações para o comprimento do reservatório

Assim como feito para o comprimento do reservatório clique na aba horizontal *Thickness* (Espessura) e preencha o valor de 75 para todas as células. A Figura C.4 mostra a

aba *Thickness* completamente preenchida.

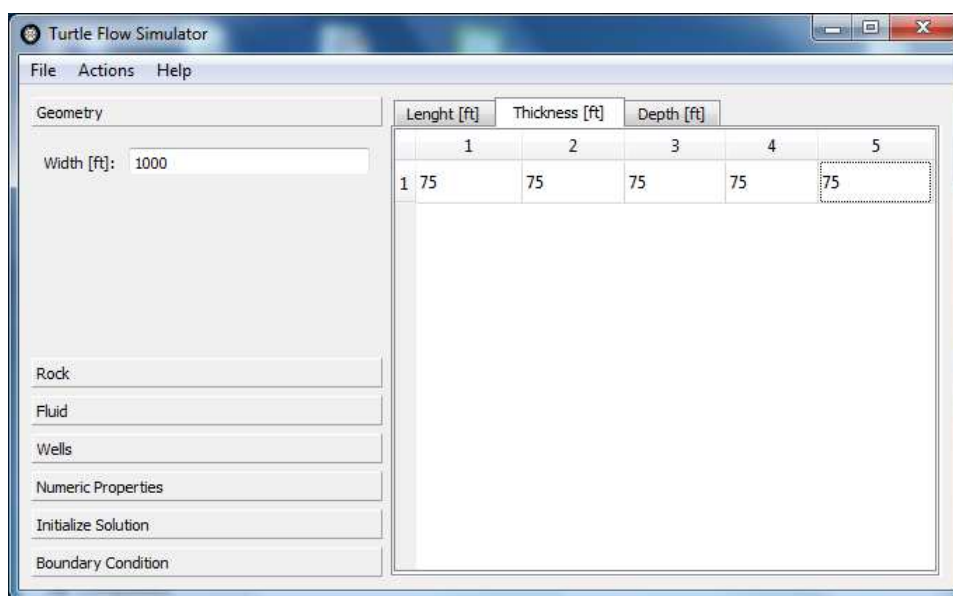


Figura C.4: Configurações para a espessura das células do reservatório

Por último preencha na aba *Depth* (Profundidade) um valor de 0 para todas as células.<sup>1</sup> A Figura C.5 mostra a aba *Depth* preenchida.

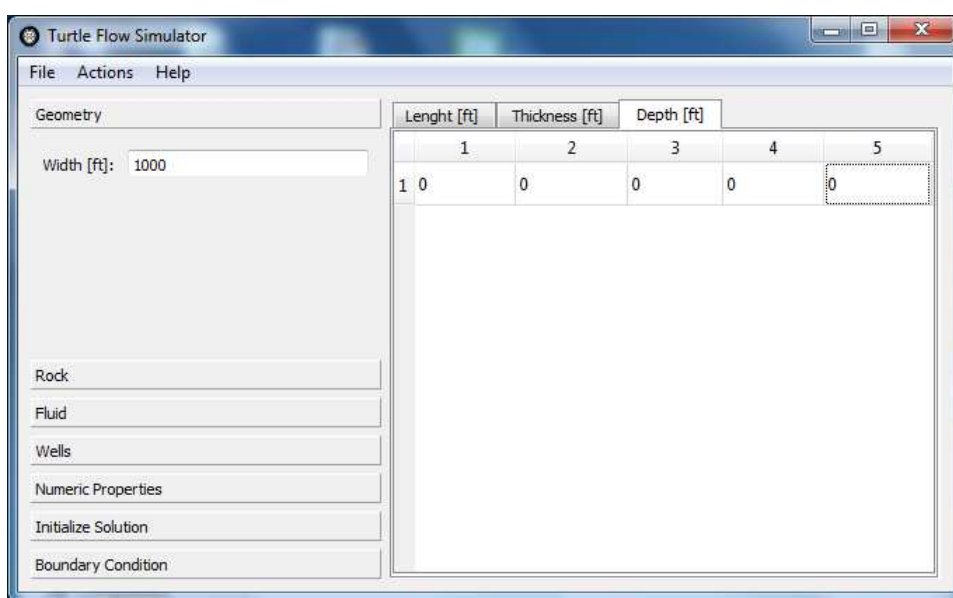


Figura C.5: Aba *Depth* preenchida

A próxima etapa é o preenchimento das propriedades de rocha do reservatório. Para isso clique na aba vertical *Rock* (Rocha) e entre um número de blocos igual a 1. Preencha os valores do bloco 1 com os dados mostrado na Figura C.1. Após preenchido os valores

<sup>1</sup>Como o TFS trata a profundidade relativa entre os blocos, qualquer valor digitado no campo *Depth* traria os mesmos resultados, importando apenas a variação horizontal desses valores.

do bloco 1, associe todas as células à esse bloco, para isso coloque o valor 1 em todas as células a direita. A Figura C.6 mostra a aba *Rock* preenchida.

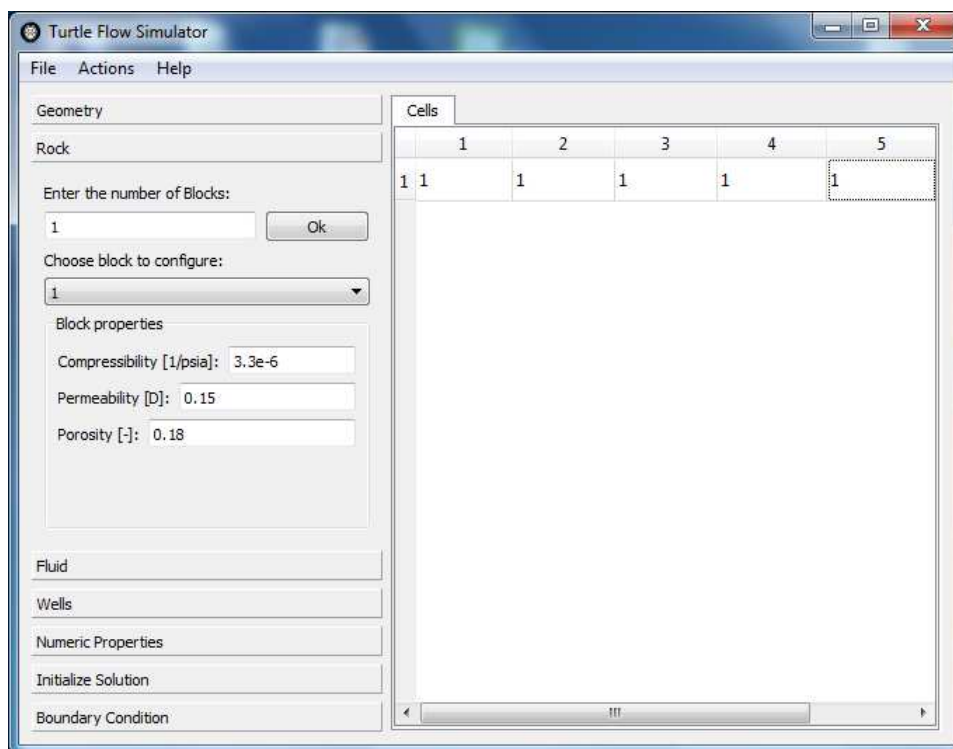


Figura C.6: Inserção das propriedades de rocha no problema

Em sequência é necessário se preencher as propriedades de fluido do problema. Clique na aba *Fluid* e escolha 2 pontos do teste PVT<sup>2</sup>. Coloque as propriedades de fluido mostradas na Figura C.1 entre os pontos 1000 e 10000 psia de pressão. Observa-se que como o caso simulado é horizontal qualquer valor de densidade escolhido produzirá os mesmos resultados. A Figura C.7 mostra os dados de fluido devidamente preenchidos.

<sup>2</sup>O TFS faz interpolações na tabela PVT para conseguir as propriedades de fluido em todos os pontos de pressão da simulação, dessa forma é sempre necessário preencher a tabela PVT no *range* de pressão esperados para o problema.

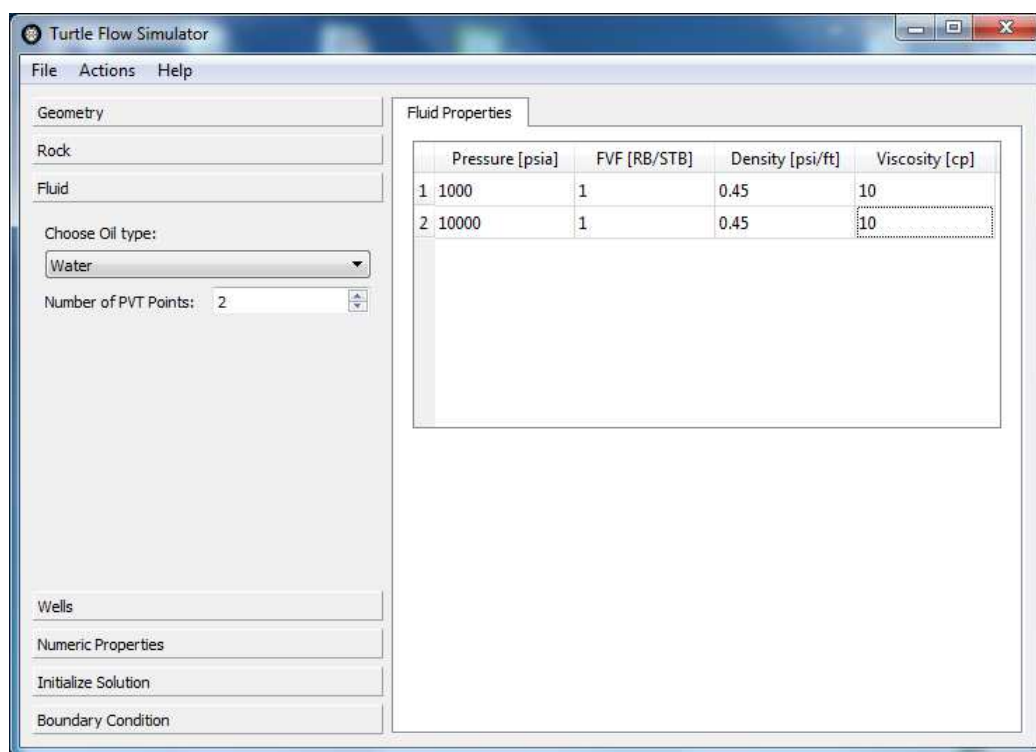
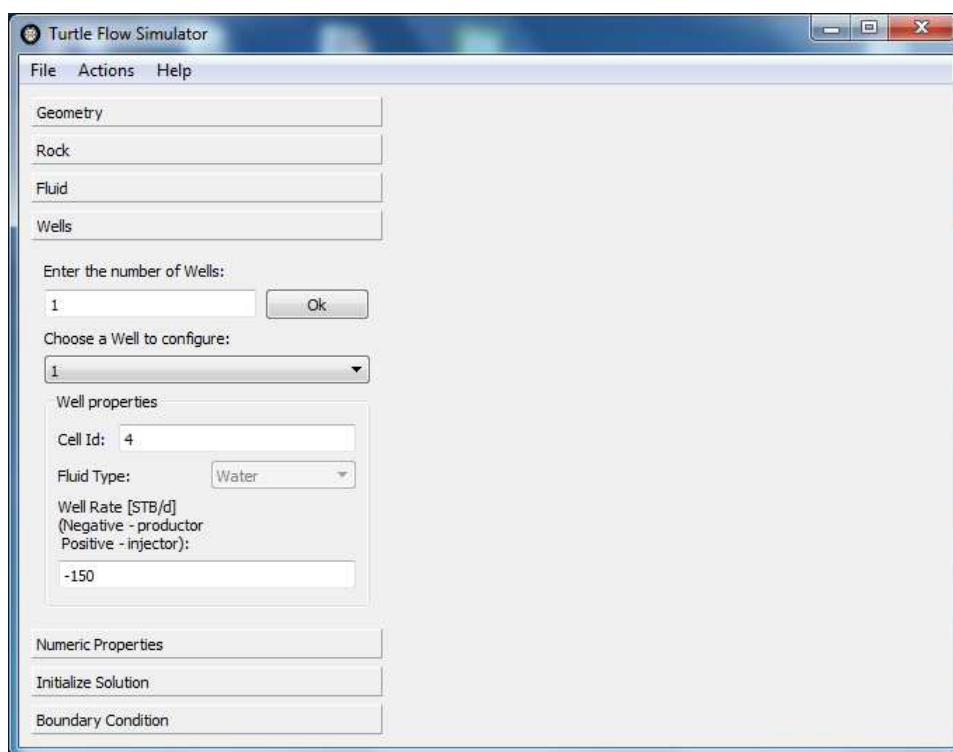


Figura C.7: Dados de fluido para o reservatório homogêneo

Para se preencher as propriedades de poço clique na aba *wells* (poço) e entre com 1 poço para o problema. Coloque o poço na célula 4 e coloque a vazão de -150 para esse poço.<sup>3</sup> A Figura C.8 mostra a aba *Wells* preenchida.

<sup>3</sup>Como notação a vazão negativa indica que o problema possui um poço produtor, enquanto valores positivos indicam a presença de poços injetores.

Figura C.8: Aba *Wells* devidamente preenchida

O próximo passo é a definição das propriedades numéricas do problema. Clique na aba *Numerical Properties* (Propriedades Numéricas) e preencha os valores dessa propriedade conforme mostrado na Figura C.9. Nesse caso será simulado a produção desse reservatório durante dois anos, sendo que o passo de tempo da simulação será de 1 dia.

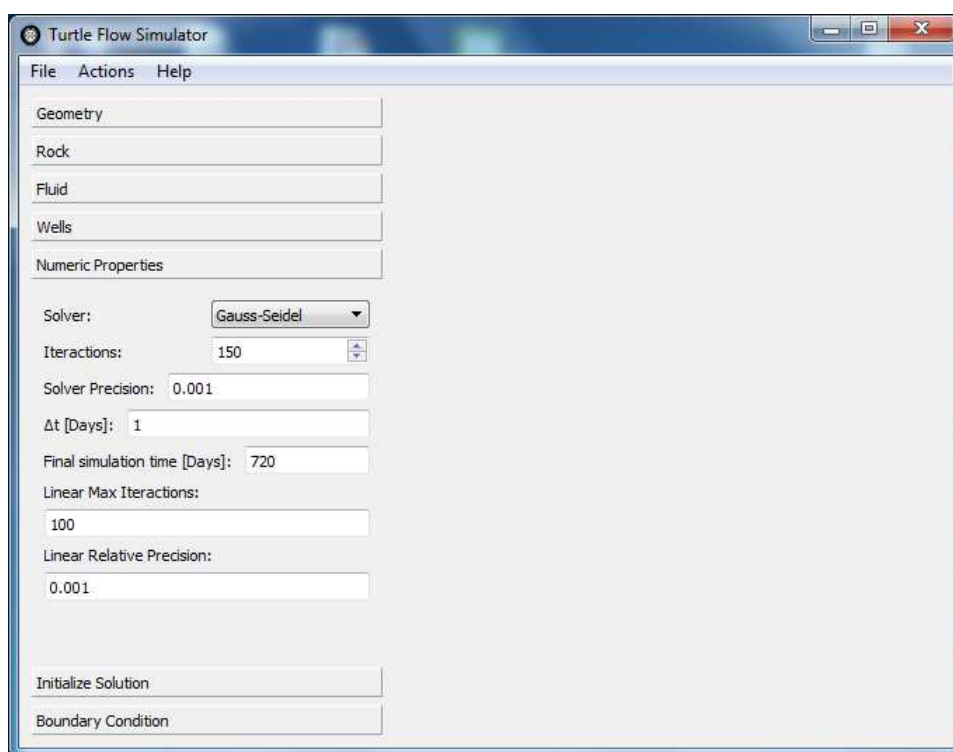


Figura C.9: Propriedades numéricas do problema

Para se definir a solução inicial do problema, clique na aba Initialize Solution e preencha os valores do problema como mostrado na Figura C.10. Observa-se que o valor de profundidade de referência deve ser o mesmo que o preenchido na profundidade do reservatório (Figura C.5) para um reservatório horizontal.

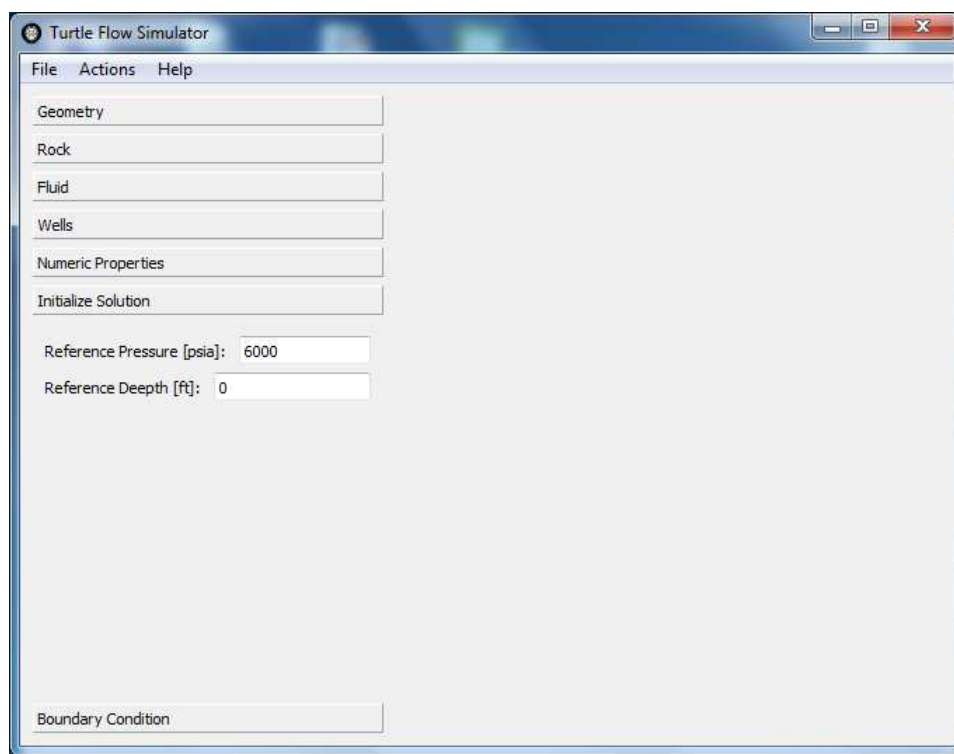


Figura C.10: Solução inicial do problema horizontal

Por último é necessário se inserir as condições de contorno do problema, para isso clique na aba *Boundary Condition* (Condições de Contorno) e selecione reservatório fechado para ambos os lados do reservatório. Após isso clique no botão *Create Files* para se gerar os arquivos necessários para se iniciar a simulação. A Figura C.11 mostra as condições de contorno preenchidas e a mensagem de alerta que os arquivos de configuração foram devidamente gerados.

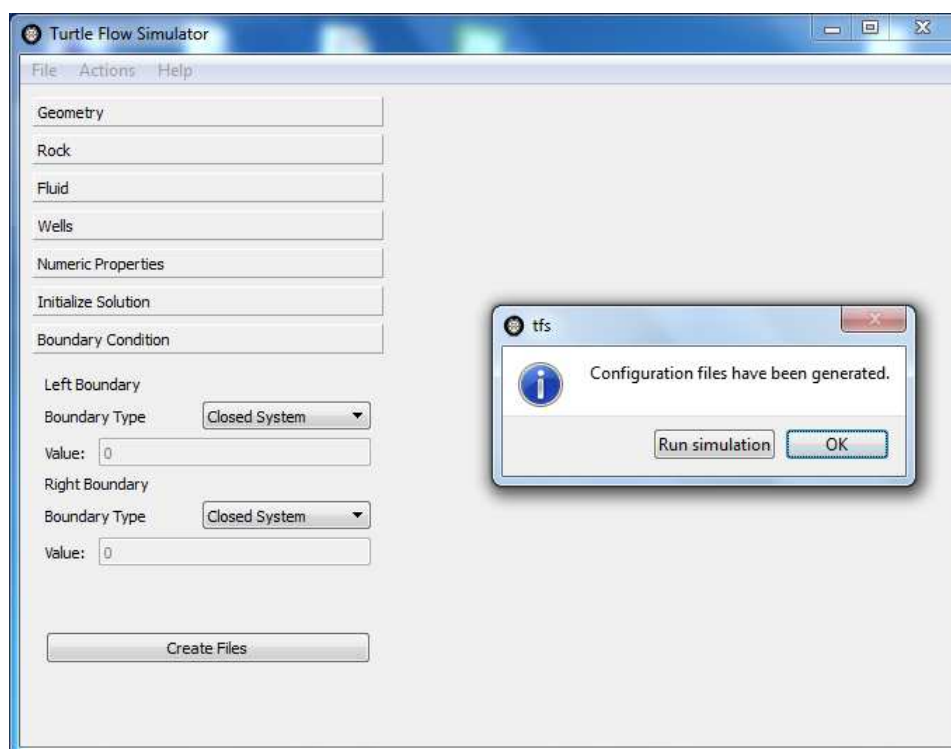


Figura C.11: Preenchimento das condições de contorno e geração dos arquivos de simulação

Para se iniciar a simulação numérica clique em *Actions*->“*Run Simulation*”, após dar Ok em uma caixa de mensagem indicando o início da simulação, irá aparecer uma janela mostrando o andamento da simulação, conforme mostrado na Figura C.12. A primeira coluna de dados mostra o tempo atual da simulação, a segunda coluna mostra o número de iterações de linearização para cada passo de tempo e a terceira coluna mostra o erro da iteração linear. É possível perceber que o erro se manteve sempre abaixo do erro inserido na Figura C.9, indicando convergência numérica do modelo.



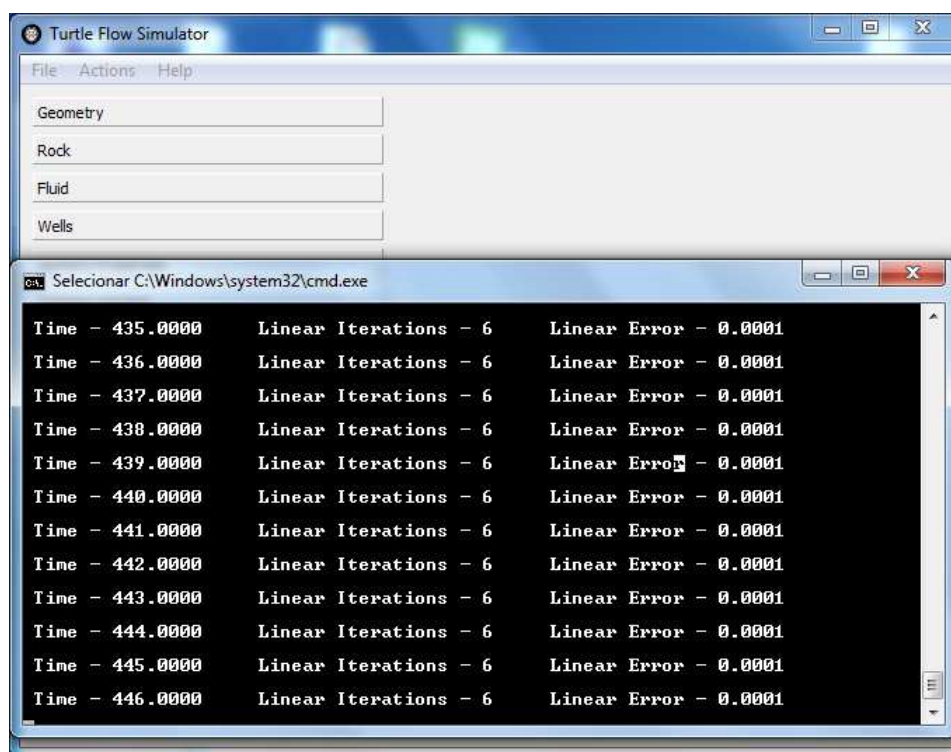


Figura C.12: Janela mostrando o andamento da simulação

Após terminar a simulação numérica é possível se plotar gráficos com os resultados da simulação. Para se plotar a produção acumulada do poço clique em *Actions->“Plot Results”* como mostrado na Figura C.13, escolha a opção *Well Production* (Produção do Poço) e coloque o número referente ao poço 1. Após isso abrirá uma janela com o gráfico desejado (Figura C.14).

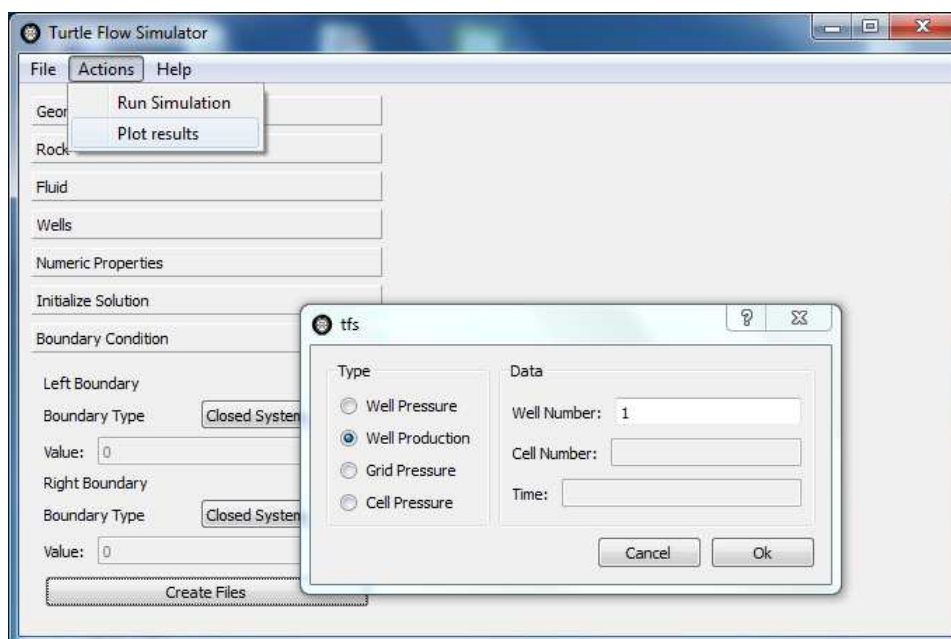


Figura C.13: Plotar produção acumulada de um poço

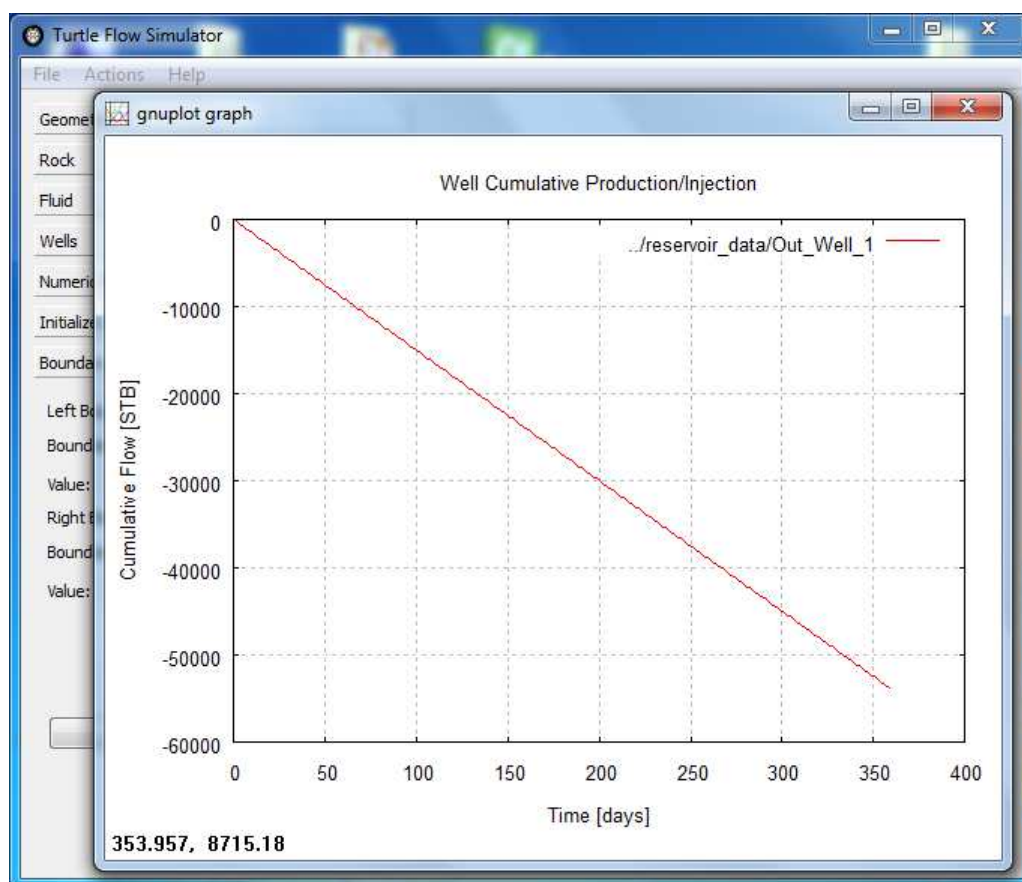


Figura C.14: Gráfico de produção acumulada para um poço especificado

Um gráfico interessante a ser plotado é o perfil de pressão do reservatório em um determinado tempo. Para isso escolha a opção “*Grid Pressure*” (Pressão do Grid) e entre, por exemplo, um tempo de 360 dias após o início da produção (Figura C.15).

A Figura C.16 mostra o perfil de pressão do reservatório após 1 ano de produção. É possível ver a redução de pressão na célula 4, devido a presença do poço produtor.

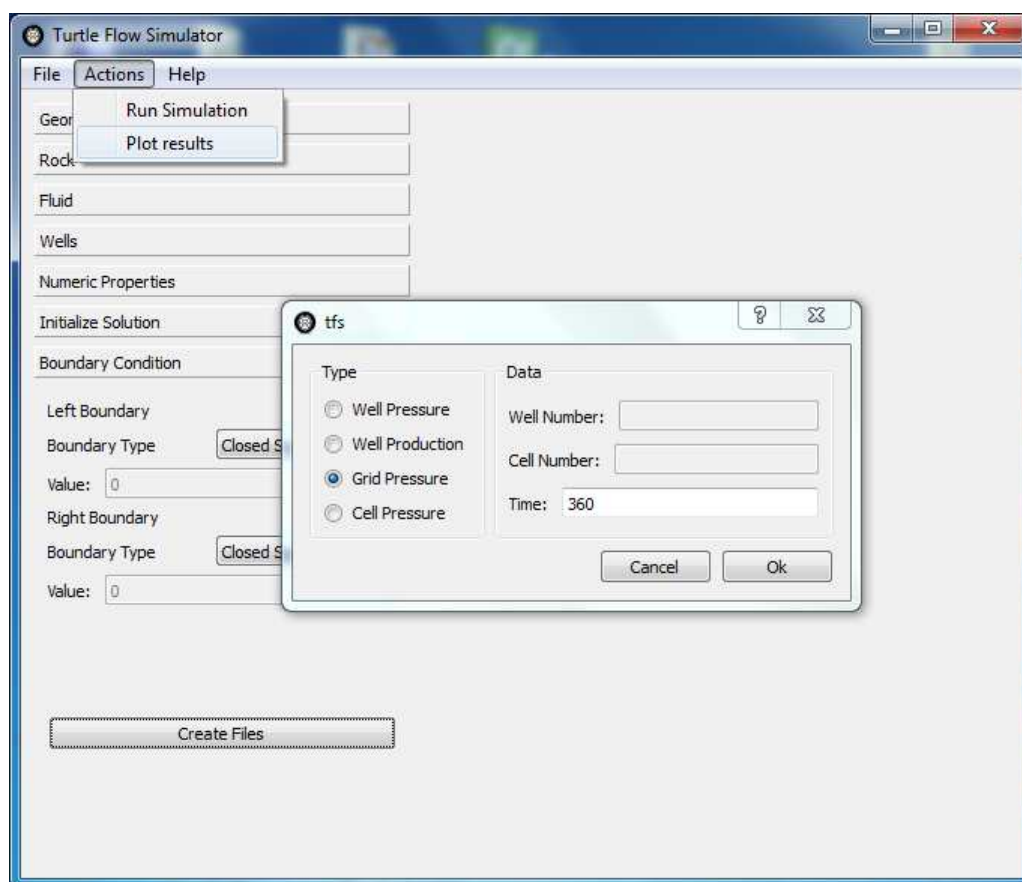


Figura C.15: Plotar o perfil de pressão do reservatório para um tempo especificado



Figura C.16: Perfil de pressão do reservatório após 360 dias após a abertura do poço

## C.2 Reservatório anticlinal com duas litologias e alimentação de pressão

O tutorial 2 mostra um reservatório em formato anticlinal com duas litologias diferentes e alimentação de pressão em um dos lados do anticlinal, como pode ser visto na Figura C.17. Nesse exemplo será mostrado apenas as diferenças entre as configurações de um reservatório homogêneo e heterogêneo, sendo necessário a leitura do Tutorial 1 para o completo entendimento.

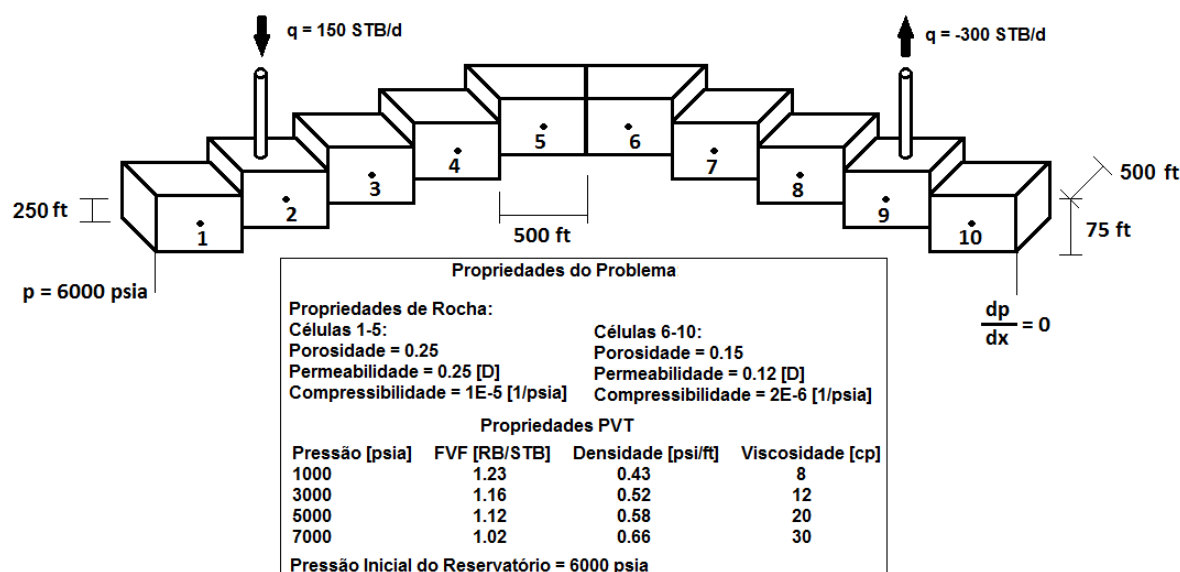


Figura C.17: Reservatório anticlinal com duas litologias e alimentação de pressão

Como pode ser visto na Figura C.17 os blocos possuem a mesma dimensão, variando apenas a profundidade relativa entre os mesmos. Como não foi dada a profundidade do reservatório, apenas a variação de profundidade entre um bloco e outro, será determinada que a profundidade do bloco 1 será a referência. A Figura C.18 mostra a guia *Depth* corretamente preenchida. Preencha o resto da geometria do problema como descrito no Tutorial 1.

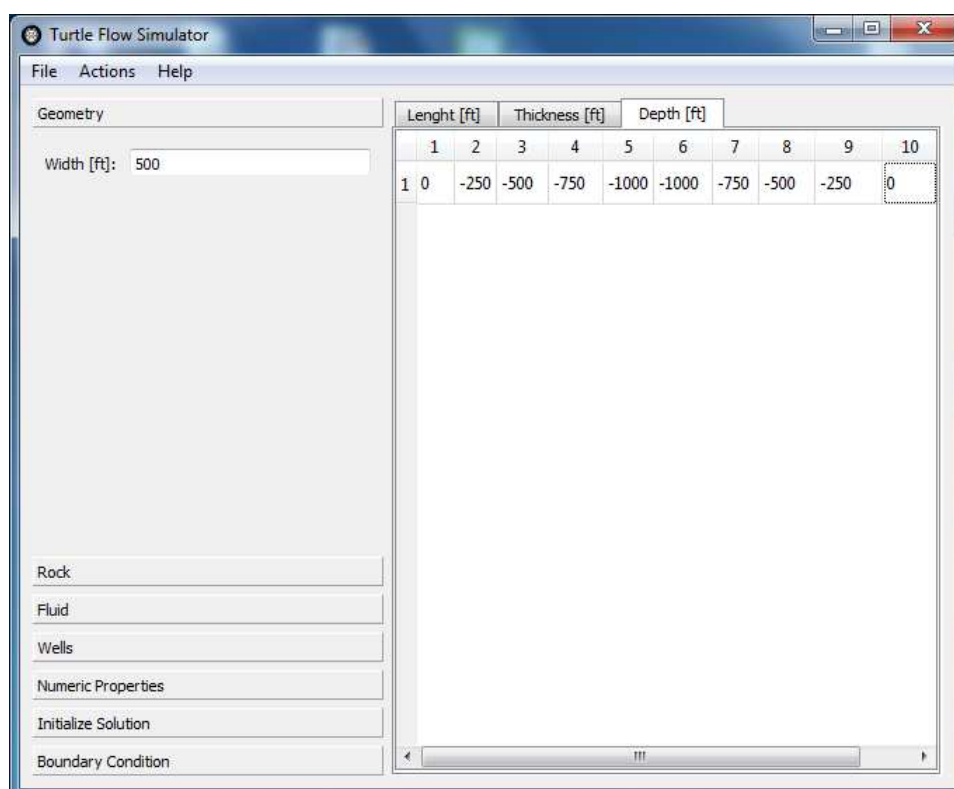


Figura C.18: Guia *Depth* corretamente preenchida para o reservatório mostrado na Figura C.17

Esse problema é composto de duas rochas com propriedades físicas diferentes, assim é necessário se inserir dois blocos diferentes no problema. Conforme mostrado na Figura C.17 as células 1-5 estão ligadas à rocha 1, e as células 6-10 estão ligadas à rocha 2. A Figura C.19 mostra a guia rocha corretamente preenchida e conectada as respectivas células.

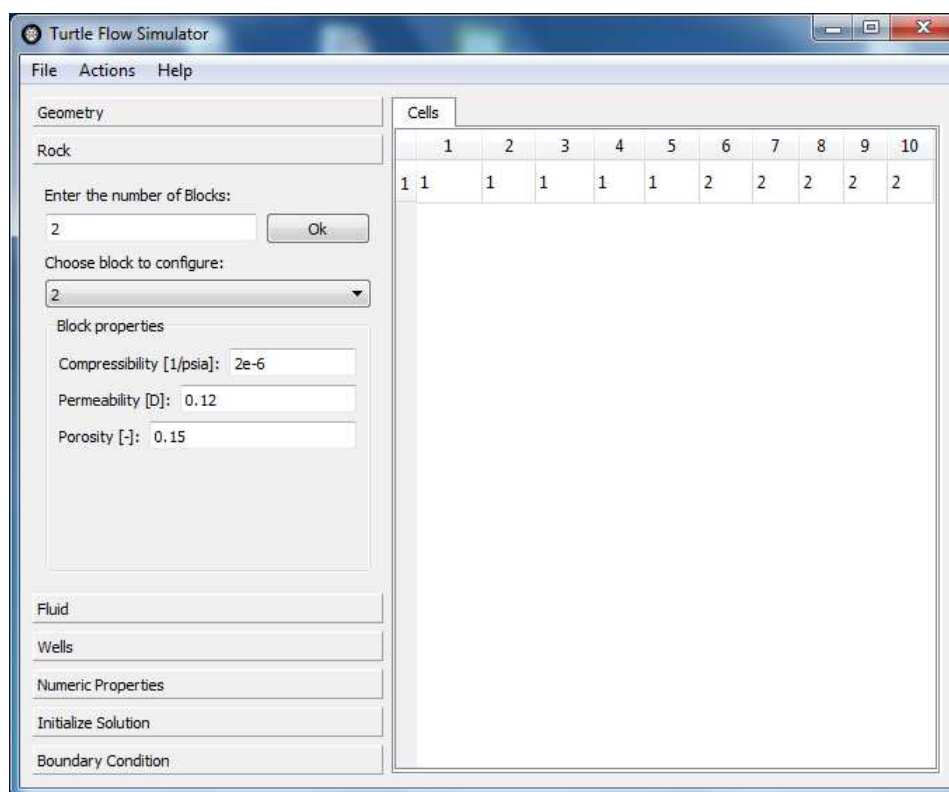


Figura C.19: Preenchimento da guia *Rock*

Preencha a guia *Fluid* e a guia *Wells* conforme demonstrado no Tutorial 1. Como esse problema é um pouco mais complexo que o discutido no Tutorial 1, é interessante se aumentar a precisão do solver, e se diminuir o passo de tempo. A Figura C.20 mostra as propriedades numéricas para esse caso.

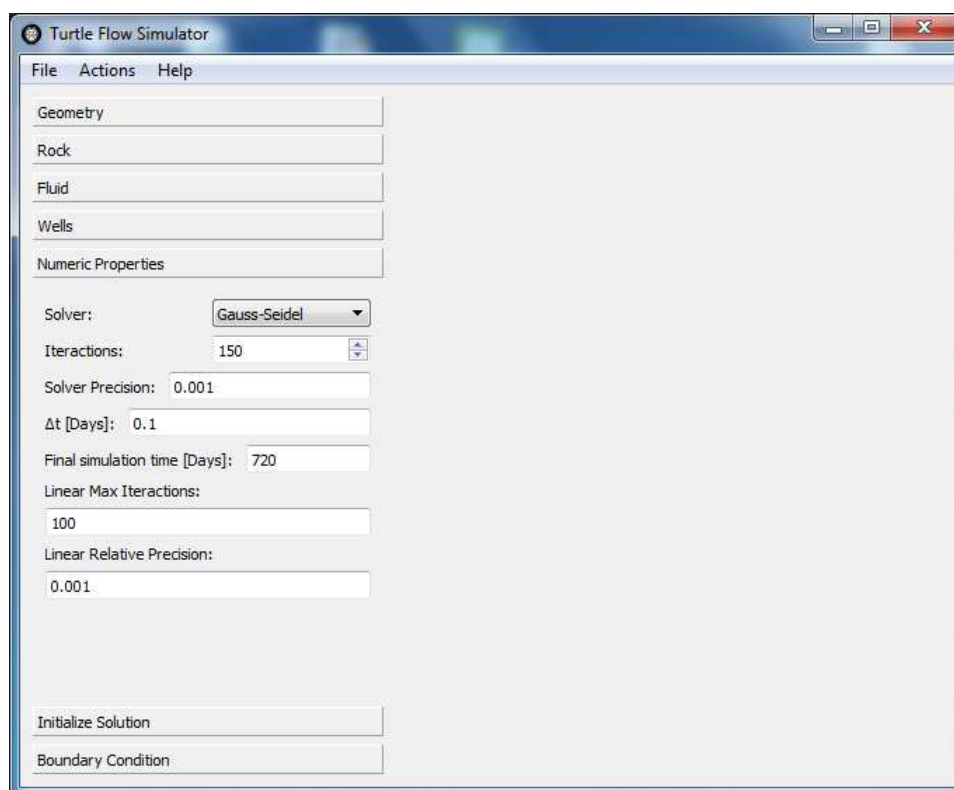


Figura C.20: Propriedades numéricas para o reservatório anticlinal

Preencha a guia *Initialize Solution* com a pressão de 6000 psia na profundidade de 0 ft (profundidade de referência do reservatório). Na guia *Boundary Condition* coloque a condição de contorno esquerda como sendo de pressão especificada (6000 psia), e a condição direita como reservatório selado, conforme mostrado na Figura C.21.

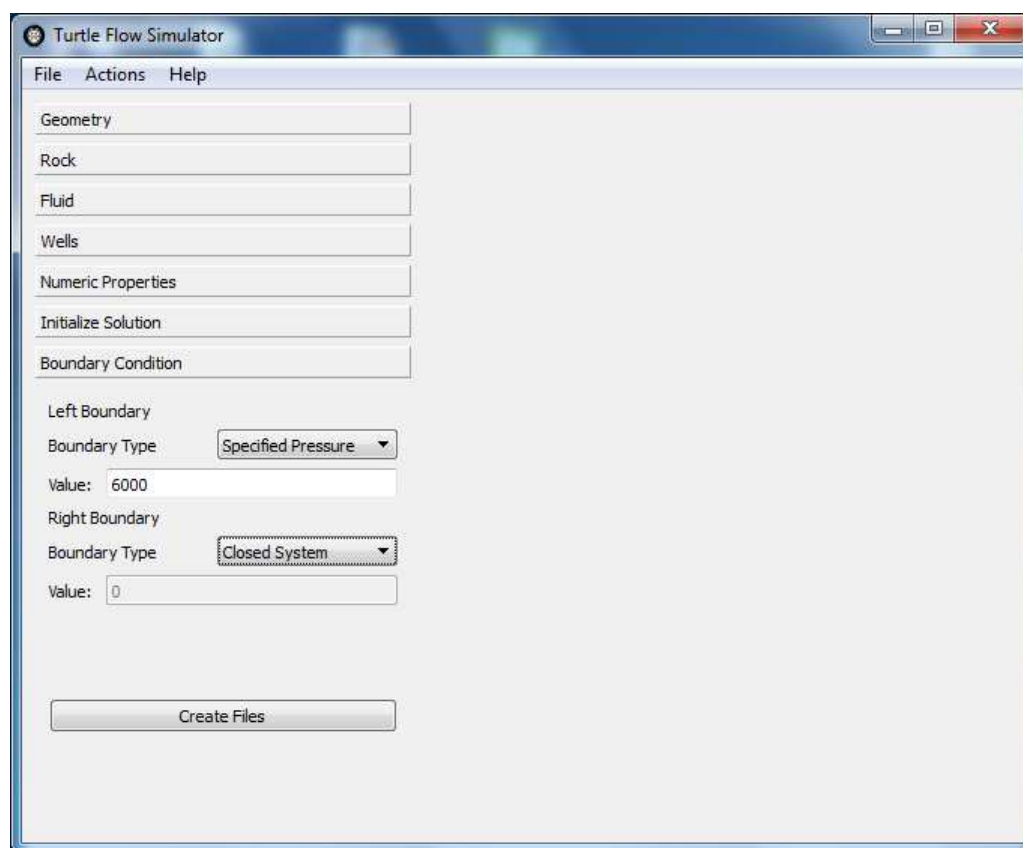


Figura C.21: Condições de contorno para o reservatório anticlinal

Crie os arquivos de simulação e em sequência execute a simulação conforme discutido no Tutorial 1. A Figura C.22 mostra o perfil de pressão do reservatório após 2 anos de produção. É possível ver a queda de pressão na célula 9 devido a presença do poço produtor, e a manutenção da pressão inicial na parte esquerda do reservatório devido a condição de contorno.





Figura C.22: Perfil de pressão para o reservatório anticlinal após 2 anos de produção

Esse tutorial ilustrou a simulação de um reservatório em formato de anticlinal com propriedades heterogêneas.