

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

PROJETO DE ENGENHARIA
SIMULADOR DE ESTIMATIVA DE RESERVA DE RESERVATÓRIO
UTILIZANDO EBM,
COM ENFOQUE NOS RESERVATÓRIOS DE GÁS

LEONARDO MESQUITA CAETANO - 2012 - V1
THIAGO PESSANHA DE MACEDO - 2012 - V1
JALILE BARTOLE PEREIRA - 2014 - V2
THAÍSA DE CASTRO ALEIXO - 2014 - V2

MACAÉ - RJ
ABRIL - 2015

Glossário

- ANP: Agência Nacional do Petróleo.
- EBM: Equação de Balanço de Materiais.
- PVT: Análise de Pressão, Volume e Temperatura.

Sumário

1	Introdução	1
1.1	Escopo do problema	1
1.2	Objetivos - V1	2
1.3	Objetivos - V2	2
2	Especificação	3
2.1	Especificação do programa - descrição dos requisitos	3
2.2	Diagrama de casos de uso do programa	3
2.2.1	Diagrama de caso de uso geral	4
2.2.2	Diagrama de caso de uso específico	4
3	Elaboração	5
3.1	Referências Bibliográficas	5
3.2	Descrição do problema físico	5
3.3	Descrição dos mecanismos de produção para reservatório de gás	7
3.3.1	Reservatórios de gás seco	8
3.3.2	Reservatórios de gás condensado	10
3.4	Descrição dos mecanismos de produção para reservatório de óleo	11
3.5	Análise do domínio	12
3.6	Identificação de pacotes	12
3.7	Diagrama de pacotes	13
4	AOO – Análise Orientada a Objeto	14
4.1	Diagramas de classes	14
4.1.1	Dicionário de classes	14
4.2	Diagrama de seqüência – eventos e mensagens	20
4.2.1	Diagrama de seqüência geral	20
4.2.2	Diagrama de seqüência específico	21
4.3	Diagrama de comunicação – colaboração	21
4.4	Diagrama de máquina de estado	22
4.5	Diagrama de atividades	22

5	Implementação	23
5.1	Código fonte	23
5.1.1	CRes	23
5.1.2	CRegLin	24
5.1.3	CResGas	27
5.1.4	CResGasSecoPres	29
5.1.5	CResGasSecoInf	30
5.1.6	CResGasSecoInfPotAquifer	32
5.1.7	CResGasSecoInfSchilthuis	36
5.1.8	CResGasCondNaoRetro	39
5.1.9	CResGasCondRetro	42
5.1.10	CResOil	45
5.1.11	CResOilCapa	48
5.1.12	CResOilInf	50
5.1.13	Main	53
6	Teste	60
6.1	Reservatório de gás	60
6.1.1	Teste 1: volumétrico de gás seco	60
6.1.2	Teste 2: anormalmente pressurizado	64
6.1.3	Teste 3: sob influxo de água	67
6.1.4	Teste 4: sob influxo de água modelo <i>Pot Aquifer</i>	69
6.1.5	Teste 5: sob influxo de água modelo <i>Schilthuis</i>	71
6.1.6	Teste 6: gás condensado não retrógrado	74
6.1.7	Teste 7: gás condensado retrógrado	76
6.2	Reservatório de óleo	79
6.2.1	Teste 8: gás em solução	79
6.2.2	Teste 9: capa de gás	80
6.2.3	Teste 10: sob influxo de água	83
7	Documentação	86
7.1	Objetivo	1
7.2	Fundamentos	1
7.3	Procedimento	1

Capítulo 1

Introdução

No presente trabalho desenvolve-se um projeto de engenharia, cujo código é desenvolvido usando-se a linguagem C++¹ e paradigma de orientação a objetos, que tem como principal objetivo implementar as equações de balanço de materias vistas na disciplina de Engenharia de Reservatório para os diferentes mecanismo de produção de petróleo. Dessa forma, a principal finalidade do programa é fornecer uma estimativa do volume original de óleo ou gás do reservatório. **É dado um enfoque maior nos reservatórios de gás, incluindo, por exemplo, o seu comportamento quando há influxo de água, e alguns modelos para o cálculo deste influxo; além de reservatórios de gás condensado.**

1.1 Escopo do problema

Na indústria do petróleo é fundamental ter uma estimativa da reserva de óleo ou gás de um campo. Além disso, é importante ter uma previsão do comportamento do reservatório antes de começar a produzir e durante toda a sua vida produtiva. Tais estimativas são importantes por diversos fatores, tais como: Estudo de Viabilidade Técnica e Econômica (EVTE), declaração de comercialidade para a ANP, tomada de decisões estratégicas (investir ou abandonar um campo).

Para fazer as estimativas de uma acumulação petrolífera se faz uso da Equação de Balanço de Materiais (EBM), que é um balanço de massas dos fluidos existentes no interior dos poros das rochas-reservatório. Suas principais utilizações práticas são a determinação do volume original de óleo, a determinação do volume original de gás e a previsão do comportamento do reservatório.

¹A linguagem utilizada é uma linguagem orientada a objeto, que utiliza o conceito de classes, o que facilita na criação de programas complexos, como os de engenharia. É uma linguagem de alto nível, fortemente tipada. É amplamente utilizada para montagem de sistemas operacionais como Windows, GNU/Linux, e programas como o Office e OpenOffice.

1.2 Objetivos - V1

Os objetivos da primeira versão do trabalho são:

- Objetivo geral:
 - Desenvolver uma solução para determinar o volume original de fluido no reservatório.
- Objetivos específicos:
 - Selecionar o tipo de fluido presente no reservatório.
 - Selecionar o mecanismo de produção.

1.3 Objetivos - V2

Os objetivos deste trabalho são:

- Objetivo geral:
 - Desenvolver uma solução para determinar o volume original de fluido (principalmente gás) no reservatório.
- Objetivos específicos:
 - Selecionar o tipo de fluido presente no reservatório (gás).
 - Selecionar o mecanismo de produção.
 - Quando houver influxo de água, escolher o modelo que será utilizado para o cálculo deste influxo.

As EBM já foram implementadas no segundo semestre de 2012 pelos alunos Leonardo Mesquita Caetano e Thiago Pessanha de Macedo. O que será feito no presente trabalho nada mais é do que a ampliação deste código, adicionando novas premissas para a determinação do volume original de gás: considerando-se reservatórios de gás condensado (retrógrado e não retrógrado), analisando-se a influência dos aquíferos e detalhando o comportamento de alguns modelos de influxo.

Capítulo 2

Especificação

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

2.1 Especificação do programa - descrição dos requisitos

Desenvolver um software que tem como principal objetivo a determinação do volume original de gás e a determinação do volume original de óleo, a partir de dados do histórico de produção, para diferentes mecanismos de produção.

O software terá interface em modo texto e, além de determinar os volumes originais de gás e óleo, vai salvar os dados analisados do histórico de produção e gerar um gráfico com o auxílio do programa externo Gnuplot.

2.2 Diagrama de casos de uso do programa

Os diagramas de caso de uso são representações visuais dos casos de uso. Estes diagramas são utilizados no intuito de demonstrar os cenários de uso do sistema pelos usuários, os quais, ao verem esses diagramas, terão uma visão geral do sistema. Desse modo, os diagramas demonstram a forma de interação entre o software e o usuário.

2.2.1 Diagrama de caso de uso geral

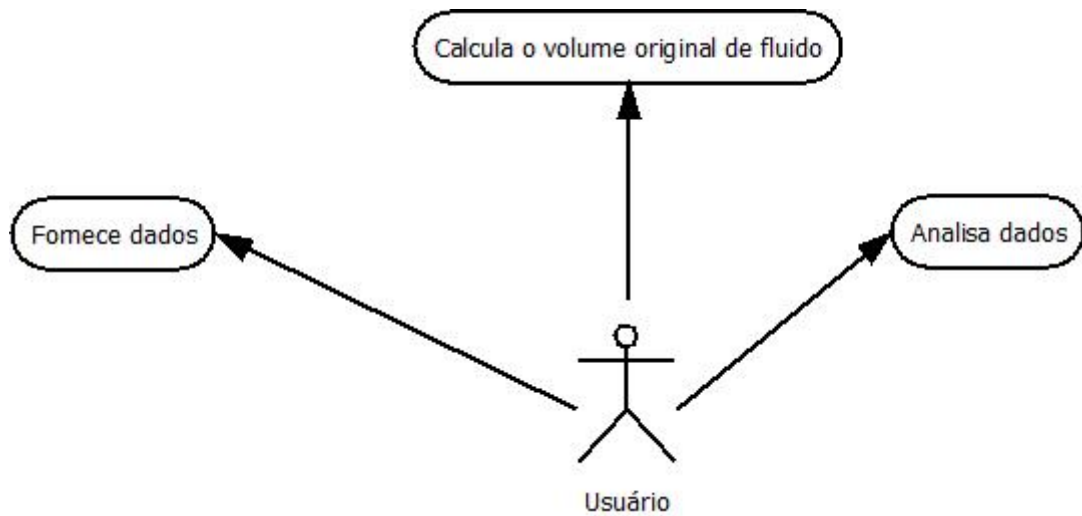


Figura 2.1: Diagrama de caso de uso geral

2.2.2 Diagrama de caso de uso específico

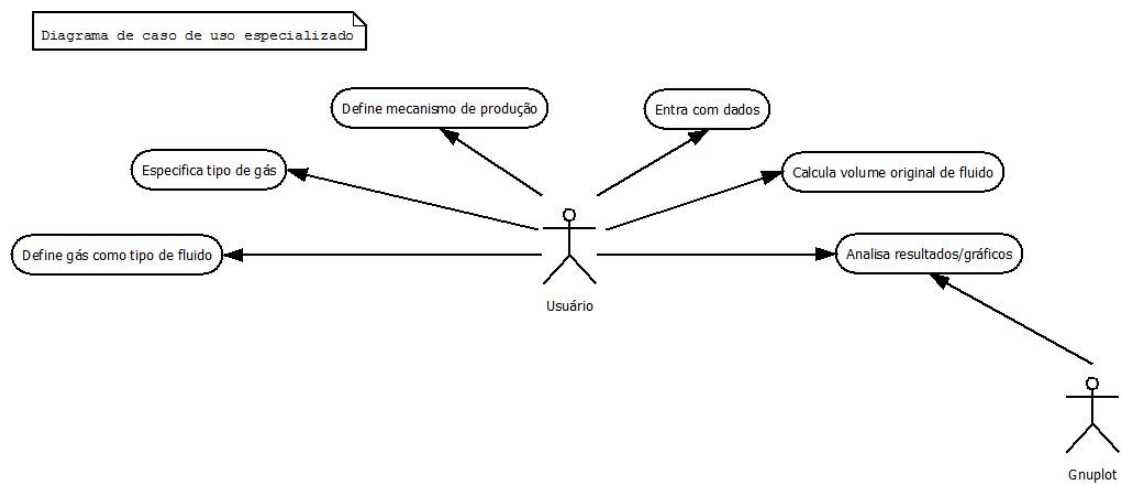


Figura 2.2: Diagrama de caso de uso para reservatório de gás seco sob influxo de água

Capítulo 3

Elaboração

Neste capítulo faremos a análise de domínio, identificação de pacotes e diagrama de pacotes.

3.1 Referências Bibliográficas

- ROSA, ADALBERTO J.; CARVALHO, RENATO S.; XAVIER, JOSÉ AUGUSTO D. Engenharia de Reservatório de Petróleo. Rio de Janeiro: Editora Interciência, 2006.
- AHMED, TAREK. Reservoir Engeneering Handbook. Second Edition. Texas: Gulf Professional Publishing, 2001.
- DAKE, L. P. Fundamentals of reservoir engineering. Rio de Janeiro: Elsevier, 1978.

3.2 Descrição do problema físico

O software calcula as reservas dos reservatórios de óleo e gás, utilizando diferentes métodos de acordo com o tipo de mecanismo de produção. Para isto será utilizada a equação de balanço de materiais, que se baseia no balanço de massas dos fluidos existentes no interior dos poros da rocha-reservatório.

A massa de fluidos existente no reservatório em um determinado instante é a diferença entre a massa original e a massa produzida. Como os volumes dos fluidos produzidos são geralmente medidos em uma determinada condição-padrão de pressão e de temperatura, a equação de balanço de materiais é comumente escrita de tal maneira que, em um instante qualquer, o volume de fluidos existente no reservatório seja a diferença entre o volume inicialmente existente e o produzido, ambos medidos em condição-padrão.

Para reservatórios de gás o programa utiliza a equação de balanço de materiais:

$$\frac{p}{Z} = \frac{1}{V_i - W_e} \left(\frac{p_i V_i}{Z_i} - \frac{T p_0}{T_0} G_p \right) \quad (3.1)$$

Onde:

- p [kgf/cm²] = pressão
- Z [adimensional] = coeficiente ou fator de compressibilidade
- V [m³] = volume ocupado de gás
- W_e [m³] = influxo de água acumulado
- G_p [m³ std] = gás produzido
- Índice i = condições iniciais no reservatório
- Índice 0 = condições padrão

A equação de balanço de materiais utilizada para reservatório de óleo pode ser escrita como:

$$N = \frac{N_p [B_o + (R_p - R_s) B_g] - W_e}{B_o - B_{oi} + (R_{si} - R_s) B_g + m B_{oi} \left(\frac{B_g}{B_{gi}} - 1 \right) + (1 + m) B_{oi} \left(\frac{c_w S_{wi} + c_f}{1 - S_{wi}} \right) \Delta p} \quad (3.2)$$

Onde:

- N [m³std] = volume original de óleo
- N_p [m³ std] = volume de óleo produzido
- B_o [m³/m³ std] = fator volume-formação do óleo
- R_p [m³ std/m³ std] = razão gás/óleo acumulada
- R_s [m³/m³ std] = razão de solubilidade
- B_g [m³/m³ std] = fator volume-formação do gás
- W_e [m³] = influxo de água acumulado
- m [adimensional] = razão entre o volume original de gás e o volume original de óleo em condições de reservatório
- c_w [(kgf/cm²)⁻¹] = compressibilidade da água
- S_{wi} [adimensional] = saturação de água conata
- c_f [(kgf/cm²)⁻¹] = compressibilidade da formação ou de poros

- $\Delta p = p_i - p$ [kgf/cm²] = variação de pressão
- Índice i = condições iniciais no reservatório

Podemos simplificar esta equação, para facilitar a regressão linear posteriormente:

$$F = N(E_o + mE_g + E_{f,w}) + W_e, \quad (3.3)$$

onde:

$$F = N_p[B_o + (R_p - R_s)B_g], \quad (3.4)$$

representa o volume de fluidos produzidos;

$$E_o = B_o + (R_{si} - R_s)B_g - B_{oi}, \quad (3.5)$$

representa a expansão dos fluidos da zona de óleo;

$$E_g = B_{oi} \frac{(B_g - B_{gi})}{B_{gi}}, \quad (3.6)$$

representa a expansão do gás da capa de gás;

$$E_{f,w} = (1 + m)B_{oi}((c_w S_{wi} + c_f)/(1 - S_{wi}))(p_i - p), \quad (3.7)$$

representa os efeitos da expansão da água conata e redução do volume poroso;

$$m = GB_{gi}/NB_{oi}. \quad (3.8)$$

A partir dessa equação geral o programa determina o volume de óleo original do reservatório para cada tipo de mecanismo associado analisando suas características e considerações.

3.3 Descrição dos mecanismos de produção para reservatório de gás

Os mecanismos de produção para reservatórios de gás são:

- **Depleção:** quando o reservatório é colocado para produzir, há um diferencial de pressão entre o fundo do poço e o interior do reservatório, o que causa o início da produção. À medida que o gás vai sendo produzido,

este se expande. Quando o reservatório produz somente por depleção, é chamado de reservatório volumétrico.

- **Influxo de água:** quando há um aquífero contíguo ao reservatório de gás. À medida que ocorre a retirada de gás através da produção, há um influxo de água para o interior do reservatório, que empurra o gás ainda existente no reservatório e dá continuidade à produção.

Os reservatórios de gás a serem considerados serão:

3.3.1 Reservatórios de gás seco

Reservatórios de gás seco são aqueles em que o fluido se encontra no estado gasoso tanto em condições de reservatório, quanto de superfície.

Para um reservatório volumétrico de gás seco, que produz somente por depleção, ou seja, por expansão da massa de gás existente no meio poroso, não havendo, portanto, influxo de água proveniente de aquífero, e desconsidera-se as variações de volume poroso e água conata (por serem desprezíveis comparadas à expansão do gás), a equação (3.1) reduz-se a:

$$p/Z = p_i/Z_i - T p_0 G_p / (V_i T_0), \quad (3.9)$$

ou ainda:

$$p/Z = p_i/Z_i - p_i G_p / (Z_i G). \quad (3.10)$$

Para um reservatório de gás seco anormalmente pressurizado, que apresenta grandes gradientes de pressão, a rocha é inconsolidada e altamente compressível, logo não podemos desprezar as variações do volume poroso e de água conata. Também produz por depleção. Portanto, a EBM fica:

$$\frac{p}{Z} \left[1 - \frac{(c_w S_{wi} + c_f) \Delta p}{1 - S_{wi}} \right] = \frac{p_i}{Z_i} - \frac{p_i}{Z_i G} G_p. \quad (3.11)$$

Definindo a compressibilidade efetiva do sistema água formação (c_{ewf}):

$$c_{ewf} = \frac{c_w S_{wi} + c_f}{1 - S_{wi}}, \quad (3.12)$$

simplificamos a equação (3.11) em:

$$\frac{p}{Z} (1 - c_{ewf} \Delta p) = \frac{p_i}{Z_i} - \frac{p_i}{Z_i G} G_p. \quad (3.13)$$

Reservatórios de gás seco sob influxo de água, que são aqueles em que a medida em que ocorre a retirada de massa através da produção dos poços, há influxo de água para o seu interior proveniente de um aquífero contíguo à zona de gás (este mecanismo também ocorre nos reservatórios de óleo). Para reservatórios de gás sob influxo de água considera-se que:

$$V = V_i - W_e. \quad (3.14)$$

Para determinação do influxo de água são utilizados dois modelos. No programa o usuário irá definir o modelo usado: Pot Aquifer ou Schilthuis.

- Modelo Pot Aquifer

Para aquíferos de tamanho próximo ao do reservatório. Em reservatórios com mecanismo de influxo de água, a queda de pressão devida à produção de fluidos causa uma expansão da água do aquífero contíguo à zona de óleo. Essa expansão provoca um fluxo de água para o interior do reservatório (Dake, 1978). De acordo com a definição de compressibilidade, aplicada ao aquífero, pode-se dizer que:

$$W_e = (c_w + c_f)W_i\Delta p, \quad (3.15)$$

onde W_i [m³] é o volume inicial de água no aquífero, Δp [kg/cm²] é a queda de pressão no aquífero, e a compressibilidade total do aquífero é a soma das compressibilidades da água (c_w) e dos poros (c_f), uma vez que o meio poroso se encontra 100% saturado com água. Se o aquífero possui uma geometria radial, o influxo de água pode ser calculado por (Dake, 1978):

$$W_e = (c_w + c_f)\pi(r_e^2 - r_o^2)h\phi\Delta p, \quad (3.16)$$

onde r_e [m] é o raio do aquífero, r_o [m] é o raio do reservatório e h [m] a espessura da formação. Nesse modelo, admite-se que a queda de pressão no aquífero em um determinado instante seja igual à queda de pressão no contato reservatório-aquífero.

- Modelo Schilthuis

Se, por outro lado, o aquífero é bem maior que o reservatório, essas equações se tornam inadequadas para descrever o influxo de água, porque a queda de pressão no limite do reservatório (contato água-óleo) não se transmite instantaneamente através do aquífero. Para aquíferos maiores há um intervalo de

tempo entre a queda de pressão no reservatório e a resposta do aquífero. Um dos modelos utilizados para o cálculo de influxo é o modelo de Schilthuis (1936),

$$dW_e/dt = J(p_i - p), \quad (3.17)$$

onde J [$\text{m}^3/(\text{d} \cdot \text{kgf}/\text{cm}^2)$] é denominada constante de influxo de água, também conhecida como índice de produtividade do aquífero, p_i [kgf/cm^2] é a pressão inicial do sistema e p [kgf/cm^2] é a pressão no contato óleo/água.

3.3.2 Reservatórios de gás condensado

Reservatórios de gás condensado são aqueles que em condições de reservatório o fluido se encontra no estado gasoso, mas na superfície produz uma certa quantidade de líquido.

Reservatórios de gás condensado não-retrógrado, que são aqueles em que os fluidos se encontram no estado gasoso dentro do reservatório, mas em condições de superfície produzem volumes equivalentes de água e gás.

Para reservatórios de gás condensado não-retrógrado o programa faz alguns cálculos e aplica-se a equação de balanço de materiais (3.9) acima citada. A razão disso, é que para esse tipo de reservatório precisa-se achar um novo valor para G_p [$\text{m}^3 \text{ std}$] e para isso considera que:

Precisa-se achar o volume de gás equivalente para cada 1 m^3 de condensado produzido:

$$(GE)_c = 23686d_c/M_c, \quad (3.18)$$

onde $(GE)_c$ [$\text{m}^3 \text{ std}/\text{m}^3 \text{ std}$] é o volume de gás equivalente a 1 m^3 produzido, d_c [adimensional] é a densidade do condensado e M_c [adimensional] é a massa molecular do condensado. Sendo que:

$$M_c = 44.29d_c/(1.03 - d_c). \quad (3.19)$$

Precisa-se saber também o volume de vapor de água produzido para cada 1 m^3 de condensado produzido e esse valor será

$$(GE)_w = 1316 [\text{m}^3 \text{ std}/\text{m}^3 \text{ std}]. \quad (3.20)$$

Chegando-se então no seguinte valor para $(G_p)_t$

$$(G_p)_t = G_p + (GE)_{ct} + (GE)_{wt}, \quad (3.21)$$

onde $(G_p)_t$ [m³ std] é o novo valor para G_p a ser usado na equação de balanço de materiais (3.9), G_p [m³ std] é o volume de gás produzido que continua no estado gasoso na superfície, $(GE)_{ct}$ [m³ std] é o volume acumulado de condensado produzido e $(GE)_{wt}$ é o volume acumulado de água produzida.

Reservatórios de gás condensado retrógrado, que são aqueles em que há formação de líquido ainda dentro do reservatório, à medida que ocorre a depleção. Emprega-se o fator de compressibilidade duas fases Z_{2f} [adimensional] no lugar do fator de compressibilidade Z . O valor de Z_{2f} pode ser obtido por análises PVT ou por correlações empíricas.

3.4 Descrição dos mecanismos de produção para reservatório de óleo

Os reservatórios de óleo a serem considerados são:

- Reservatórios com gás em solução, cujo mecanismo de produção se baseia na expansão do gás inicialmente dissolvido no óleo e que foi liberado devido à redução de pressão, consequência da produção de fluidos. Para este reservatório, considerando-se a compressibilidade desprezível obtemos a equação de balanço de materiais:

$$F = NE_o; \quad (3.22)$$

- Reservatórios com capa de gás, cujo mecanismo de produção se baseia na expansão da capa de gás devido à diminuição de pressão. Para um reservatório com mecanismo de capa de gás, temos:

$$F = N(E_o + mE_g); \quad (3.23)$$

- Reservatórios com influxo de água, onde a queda de pressão devida à produção de fluidos causa uma expansão da água do aquífero contíguo à zona de óleo. Para um reservatório com mecanismo de influxo de água:

$$F = NE_o + W_e. \quad (3.24)$$

Estas equações obtidas são apenas para a linearização da EBM, para facilitar a regressão linear. Seus termos já foram explicados no início deste capítulo.

3.5 Análise do domínio

- Área: Engenharia de Petróleo

A área relacionada ao desenvolvimento do programa é a Engenharia de petróleo. É a área da engenharia que trata de todos os ramos relacionados à produção de hidrocarbonetos, que podem ser óleo ou gás natural. As atividades são divididas geralmente em duas grandes áreas: upstream, refere-se às atividades de exploração e produção e downstream, que refere-se às atividades de refino e distribuição.

- Sub-área: Engenharia de Reservatório

A Engenharia de Reservatório constitui uma sub-área de extrema importância na engenharia de petróleo, e está associada a descrição do comportamento do óleo, do gás e da água no interior dos meios porosos que compõem as acumulações petrolíferas na sub-superfície, essencial para o desenvolvimento de campos de petróleo. Os engenheiros, geólogos e geofísicos de petróleo, assim como outros profissionais que atuam na área de engenharia de reservatórios, utilizam informações sobre as propriedades e características das rochas e dos fluidos contido nas formações portadoras de petróleo, bem como seu comportamento passado (no caso de parte do fluido já ter sido produzida), para inferir o comportamento futuro desses reservatórios.

- Área: Modelagem Numérica Computacional

A Modelagem Numérica Computacional é a área que trata do desenvolvimento de soluções para problemas científicos de Engenharia, analisando os fenômenos, desenvolvendo modelos matemáticos para sua descrição, e elaborando algoritmos e códigos computacionais para obtenção daquelas soluções.

3.6 Identificação de pacotes

- Pacote Reservatório:

Contém todos os conceitos associados à reservatório incluindo modelos de equação do balanço de materiais, modelos para determinação do influxo de água, e os tipos de mecanismos associados aos reservatórios como mecanismo de influxo de água, capa de gás, gás em solução, etc.

- Pacote Métodos Numéricos:

Esse pacote inclui pacotes para realização de operações numéricas como o pacote regressão linear, que será utilizado para determinação de funções lineares a partir de dados de produção do reservatório para os diferentes mecanismos de produção, e o pacote ntegral, que será utilizado nos modelos de influxo de água.

- Pacote Gráfico:

Esse pacote é responsável por plotar os gráficos.

3.7 Diagrama de pacotes

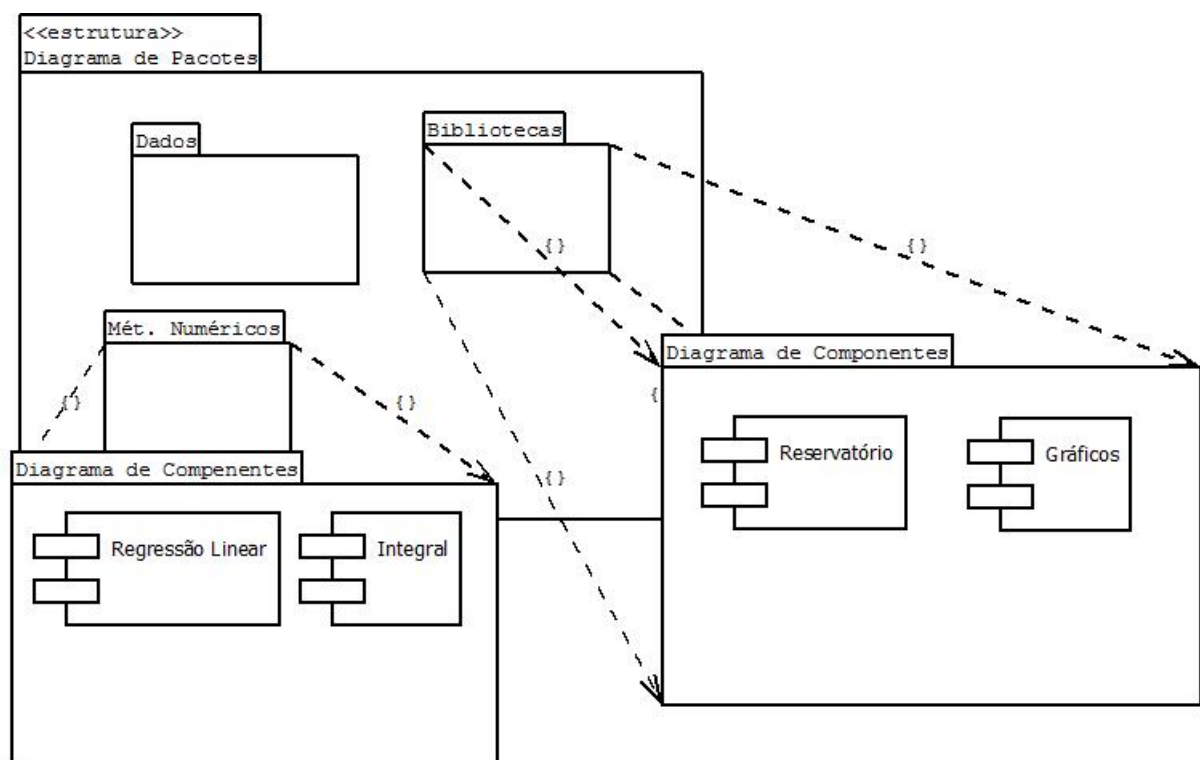


Figura 3.1: Diagrama de pacotes

Capítulo 4

AOO – Análise Orientada a Objeto

4.1 Diagramas de classes

O diagrama de classes é apresentado na Figura 4.1. Uma descrição das classes, seus atributos e métodos é apresentada a seguir.

4.1.1 Dicionário de classes

- Classe CRegLin: representa a classe que fará a regressão linear dos pontos gerados pelas classe hedeiras da classe CRes.
 - atributo a: coeficiente linear que vai caracterizar uma grandeza física de um dos mecanismos de produção.
 - atributo b: coeficiente angular que vai caracterizar uma grandeza física de um dos mecanismos de produção.
 - atributo sx: representa o somatório das variáveis x.
 - atributo sy: representa o somatório das variáveis y.
 - atributo sxx: representa o somatório do quadrado das variáveis x.
 - atributo syy: representa o somatório do quadrado das variáveis y.
 - método Reg: representa os calculos da regressão.

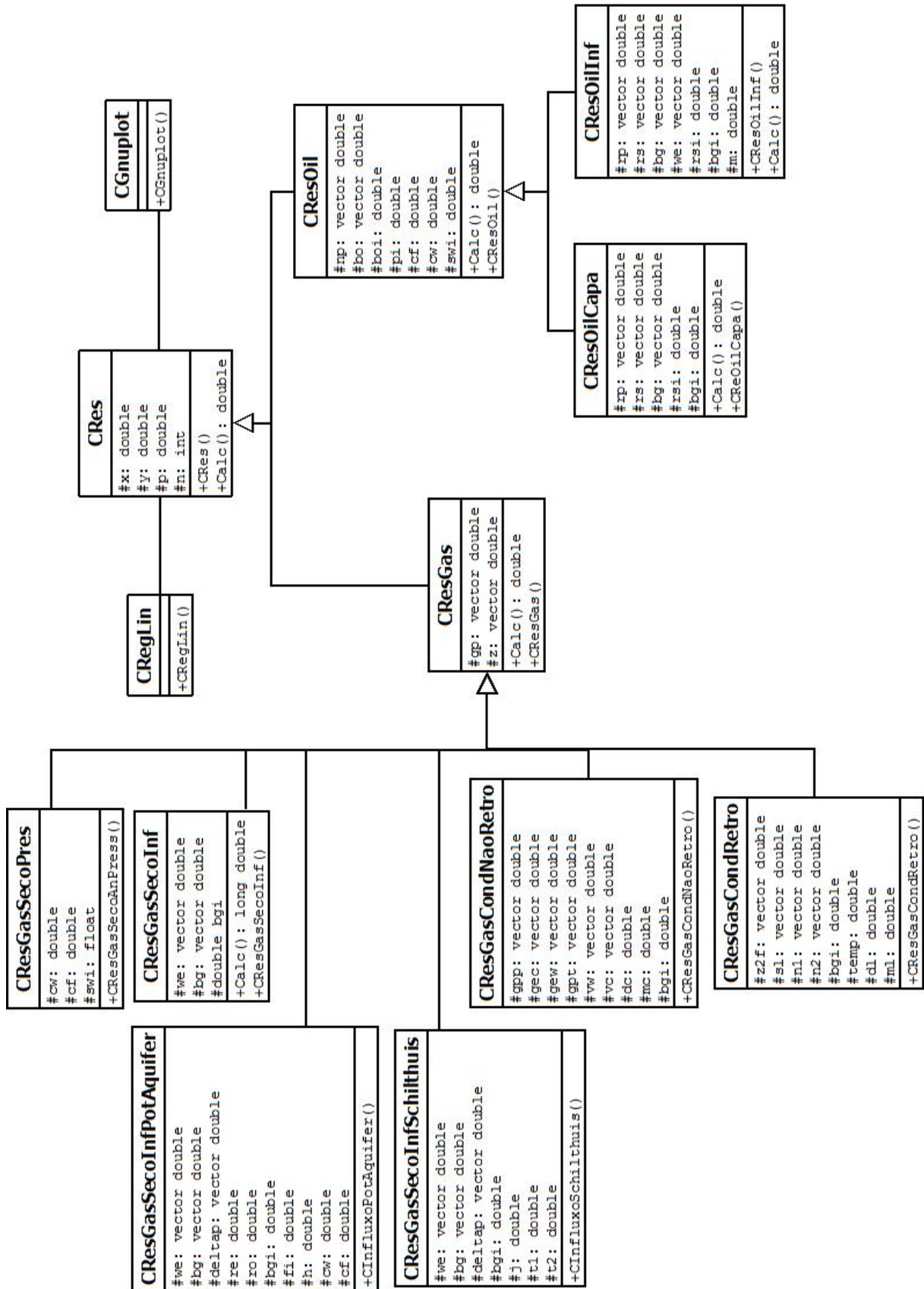


Figura 4.1: Diagrama de classes

- Classe CRes: representa os conceitos básicos da engenharia de reservatório.

- atributo x: representa a variável das abscissas para cada mecanismo de produção.
- atributo y: representa a variável das ordenadas para cada mecanismo de produção.
- atributo n: determina o número de variáveis dos atributos x e y.
- atributo p [kgf/cm²]: representa a pressão no reservatório para cada histórico de produção.
- método Calc:
- método CRes: representa o construtor da classe.
- Classe CResGas: representa os conceitos fundamentais dos mecanismos de produção de gás.
 - atributo gp [m³ std]: este armazena o volume de gás produzido.
 - atributo z [adimensional]: representa o fator de compressibilidade do gás.
 - método Calc: calcula as variáveis x e y da classe CRes.
 - método CResGas: representa o construtor da classe.
- Classe CResOil: representa os conceitos fundamentais dos mecanismos de produção de óleo.
 - atributo np [m³ std]: representa o volume de óleo produzido para cada histórico de produção.
 - atributo bo [m³/m³ std]: representa o fator volume de formação do óleo para cada histórico de produção.
 - atributo boi [m³/m³ std]: representa o fator volume de formação inicial do óleo para o primeiro histórico de produção.
 - atributo pi [kgf/cm²]: representa a pressão inicial no primeiro histórico de produção.
 - atributo cf [(kgf/cm²)⁻¹]: representa a compressibilidade da rocha reservatório.
 - atributo cw [(kgf/cm²)⁻¹]: representa a compressibilidade do fluido do reservatório.
 - atributo swi: respresenta a saturação de água inicial do reservatório.
 - método Calc: calcula as variáveis x e y da classe CRes.
 - método CResOil: representa o construtor da classe.

- Classe CResGasSecoPres: representa os conceitos fundamentais do mecanismo de produção de gás anormalmente pressurizado.
 - atributo cw $[(\text{kgf}/\text{cm}^2)^{-1}]$: representa a compressibilidade da água.
 - atributo cf $[(\text{kgf}/\text{cm}^2)^{-1}]$: representa compressibilidade da formação.
 - atributo swi: representa a saturação de água inicial para cada historico de produção.
 - método CResGasPres: representa o construtor da classe.
- Classe CResGasSecoInf: representa os conceitos fundamentais do mecanismo de gás com influxo de água.
 - atributo we $[\text{m}^3]$: respresenta o influxo acumulado do aquífero.
 - atributo bg $[\text{m}^3/\text{m}^3 \text{ std}]$: representa o fator volume de formação para cada histórico de produção.
 - atributo bgi $[\text{m}^3/\text{m}^3 \text{ std}]$: representa o fator volume de formação inicial para cada histórico de produção.
 - método CResGasSecoInf: representa o construtor da classe.
- Classe CResGasSecoInfPotAquifer: representa os conceitos do mecanismo de gás com influxo de água, utilizando o método Pot Aquifer.
 - atributo we $[\text{m}^3]$: respresenta o influxo acumulado do aquífero.
 - atributo bg $[\text{m}^3/\text{m}^3 \text{ std}]$: representa o fator volume de formação para cada histórico de produção.
 - atributo bgi $[\text{m}^3/\text{m}^3 \text{ std}]$: representa o fator volume de formação inicial para cada histórico de produção.
 - atributo deltap $[\text{kgf}/\text{cm}^2]$: representa o diferencial de pressão.
 - atributo re $[\text{m}]$: representa o raio do aquífero.
 - atributo ro $[\text{m}]$: representa o raio do reservatórioatributo we: respresenta o influxo acumulado do aquífero.
 - atributo fi: representa a porosidade do aquífero.
 - atributo h $[\text{m}]$: representa a espessura do aquífero.
 - atributo cw $[(\text{kgf}/\text{cm}^2)^{-1}]$: representa a compressibilidade da água.
 - atributo cf $[(\text{kgf}/\text{cm}^2)^{-1}]$: representa a compressibilidade da formação.
 - método CResGasSecoInfPotAquifer: representa o construtor da classe.

- **Classe CResGasSecoInfSchilthuis:** representa os conceitos do mecanismo de gás com influxo de água, utilizando o método de Schilthuis.
 - atributo w_e [m^3]: respresenta o influxo acumulado do aquífero.
 - atributo bg [m^3/m^3 std]: representa o fator volume de formação para cada histórico de produção.
 - atributo bgi [m^3/m^3 std]: representa o fator volume de formação inicial para cada histórico de produção.
 - atributo Δp [kgf/cm^2]: representa o diferencial de pressão.
 - atributo j [$m^3/(d*kgf/cm^2)$]: representa a constante de influxo.
 - atributos t_1 e t_2 [d]: representam os limites fixos de integração.
 - método CResGasSecoInfSchilthuis: representa o construtor da classe.
- **Classe CResGasCondNaoRetro:** representa os conceitos do mecanismo de gás condensado não-retrógrado.
 - atributo gpp [m^3 std]: representa o volume de gás produzido parcial.
 - atributo gec [m^3 std]: representa o volume de gás condensado equivalente.
 - atributo gew [m^3 std]: representa o volume de vapor d'água equivalente.
 - atributo gpt [m^3 std]: representa o volume total de gás produzido.
 - atributo vw [m^3]: representa o volume de água produzida.
 - atributo vc [m^3]: representa o volume de gás condensado produzido.
 - atributo dc : representa a densidade do gás.
 - atributo mc : representa a massa de condensado.
 - atributo bgi [m^3/m^3 std]: representa o fator volume de formação inicial para cada histórico de produção.
 - método CResGasCondNaoRetro: representa o construtor da classe.
- **Classe CResGasCondRetro:** representa os conceitos do mecanismo de gás condensado retrógrado.
 - atributo bgi [m^3/m^3 std]: representa o fator volume de formação inicial para cada histórico de produção.
 - atributo sl : representa a saturação de líquido no reservatório.
 - atributo n_1 : número de mols de líquido.

- **atributo n2:** número de mols de líquido.
 - **atributo temp [K]:** temperatura do gás
 - **atributo dl:** representa a densidade do líquido.
 - **atributo ml:** representa a massa do líquido.
 - **atributo z2f:** representa o fator de compressibilidade de duas fases.
 - **método CResGasCondRetro:** representa o construtor da classe.
- **Classe COilCapa:** representa os conceitos do reservatório que tem como mecanismo de produção a expansão da capa de gás.
 - atributo bg: representa o fator volume de formação do gás para cada histórico de produção.
 - atributo bgi: representa o fator volume de formação inicial do gás para o primeiro histórico de produção.
 - atributo rp: representa a razão gás/óleo acumulada em condições de temperatura e pressão do reservatório.
 - atributo rs: representa razão de solubilidade do gás no óleo.
 - atributo rsi: representa razão de solubilidade inicial do gás no óleo.
 - método COilCapa: representa o construtor da classe.
- **Classe COilInf:** representa os conceitos do reservatório que tem como mecanismo de produção o influxo de água do aquífero para zona de óleo.
 - atributo bg: representa o fator volume de formação do gás para cada histórico de produção.
 - atributo bgi: representa o fator volume de formação inicial do gás para o primeiro histórico de produção.
 - atributo rp: representa a razão gás/óleo acumulada em condições de temperatura e pressão do reservatório.
 - atributo rs: representa razão de solubilidade do gás no óleo.
 - atributo rsi: representa razão de solubilidade inicial do gás no óleo.
 - atributo we: respresenta o influxo acumulado do aquífero.
 - atributo m: representa a razão volume de gás na capa de gás pelo volume de óleo na zona de óleo.
 - método COilInf: representa o construtor da classe.

4.2 Diagrama de seqüência – eventos e mensagens

O diagrama de seqüência enfatiza a troca de eventos e mensagens e sua ordem temporal. Contém informações sobre o fluxo de controle do programa. 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 seqüência geral

Veja o diagrama de seqüência na Figura 4.2.

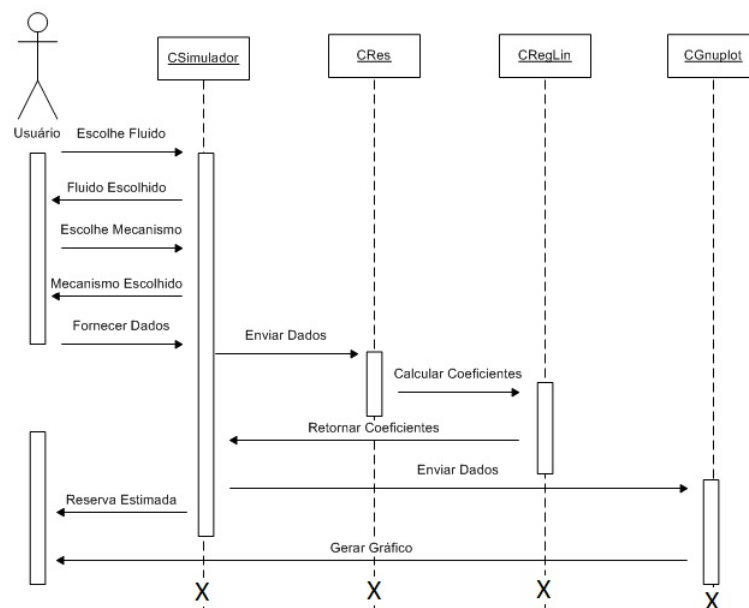


Figura 4.2: Diagrama de seqüência

4.2.2 Diagrama de sequência específico

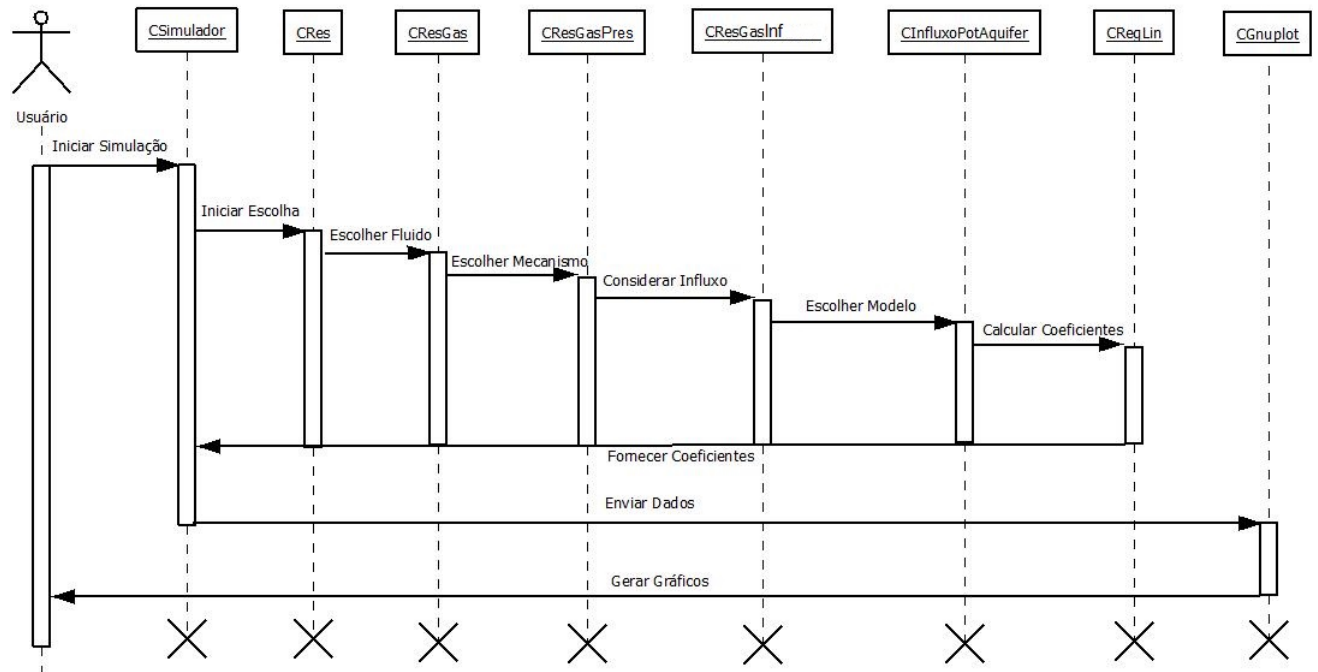


Figura 4.3: Diagrama de sequência específico - Reservatório

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.

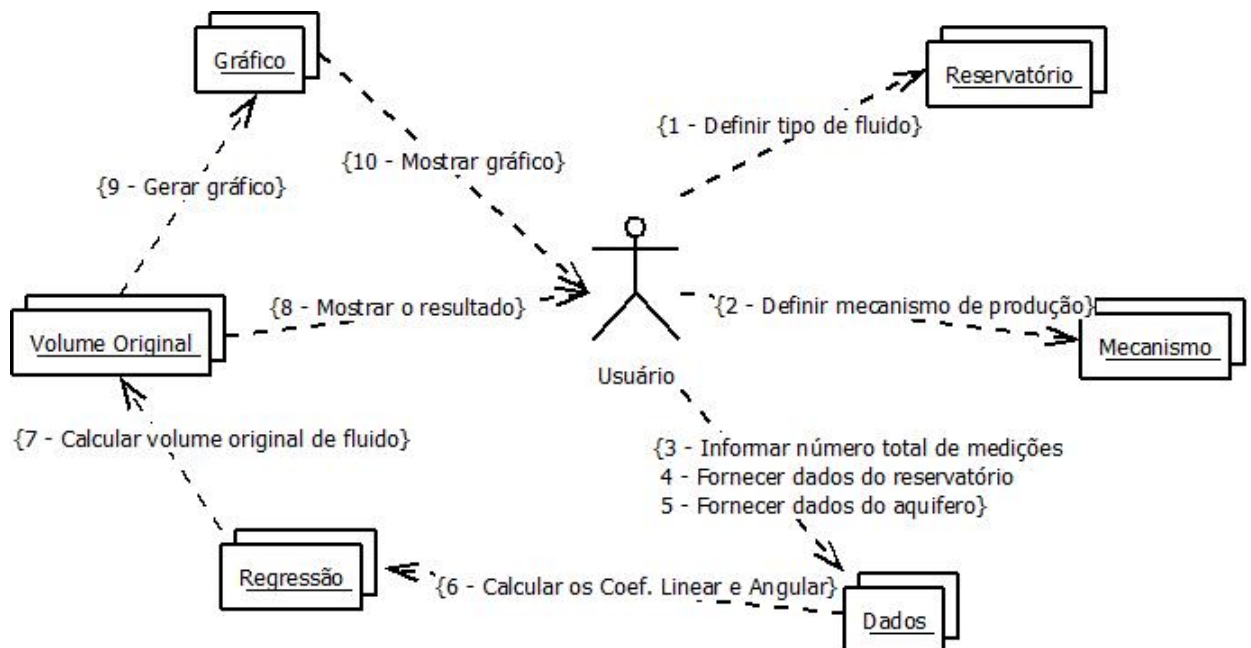


Figura 4.4: Diagrama de comunicação

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.

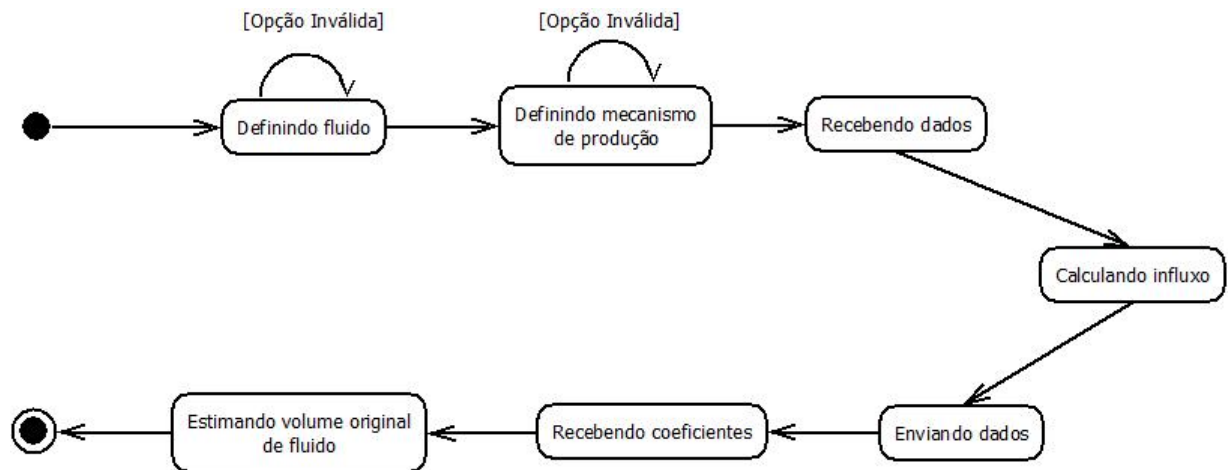


Figura 4.5: Diagrama de Máquina de Estado

4.5 Diagrama de atividades

Veja na Figura o diagrama de atividades correspondente a uma atividade específica do diagrama de máquina de estado.

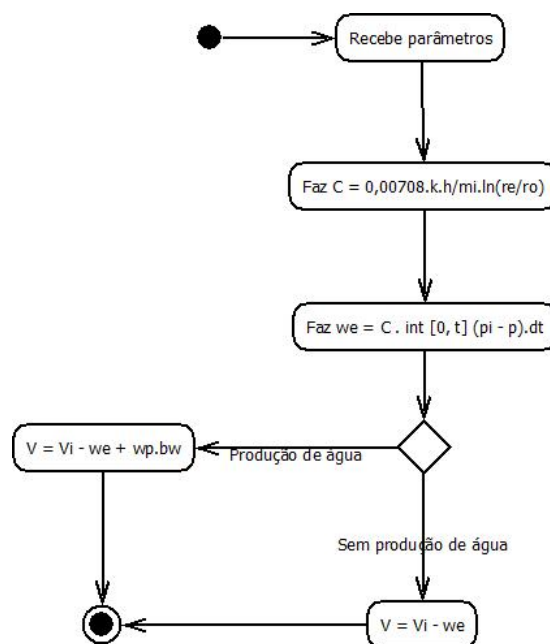


Figura 4.6: Diagrama de atividades, detalhando um método para o cálculo do influxo

Capítulo 5

Implementação

Apresenta-se neste capítulo os códigos fontes em C++.

5.1 Código fonte

5.1.1 CRes

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

Listing 5.1: Arquivo de cabeçalho da classe CRes

```
//Classe Base
//Classes Derivadas: CResGas e CResOil
#ifndef CRES_H
#define CRES_H

#include <vector>

class CRes
{
    ///Protected permite as classes derivadas usar os atributos declarados
    aqui
protected:
    ///declarando vetor
    std::vector<double> x; //representa valores do eixo x que serao
        plotados no grafico
    std::vector<double> y; //representa valores do eixo y que serao
        plotados no grafico
    std::vector<double> p; //representa os valores de pressao
    int n;

    ///Public permite que todas as classes possam acessar o próximos métodos
public:
    ///Declara o método virtual para que se possa ser feito o polimorfismo e
        assim redefinir o mesmo método nas classes derivadas
```

```

virtual void Calc() = 0;
///metodo para pegar os valores de pressão
double P(int i) const {return p[i];}
///metodo para pegar os valores do vetor de pressões
std::vector<double> P() const {return p;}
///metodo para pegar o valores de x
double X(int i) const {return x[i];}
///metodo para pegar o valores do vetor das variáveis x
std::vector<double> X() const {return x;}
///metodo para pegar o valores de y
double Y(int i) const {return y[i];}
///metodo para pegar o valores do vetor das variáveis y
std::vector<double> Y() const {return y;}
///metodo que pega o valor de n
int N() const {return n;}

///construtor default da classe
CRes() {}
};

#endif

```

5.1.2 CRegLin

Apresenta-se na listagem 6.2o arquivo com código da classe CRegLin.

Listing 5.2: Arquivo de cabeçalho da classe CRegLin

```

#ifndef CREGLIN_H
#define CREGLIN_H

#include "CRes.h"

class CRegLin
{
    ///Protected permite as classes derivadas usar os atributos declarados aqui
protected:
    ///declara o atributo "a" que corresponde ao coeficiente linear da reta
    double a;
    ///Declara o atributo "b" que corresponde ao coeficiente angular da reta
    double b;

    double media, maximo, minimo;
    ///Public permite que todas as classes possam acessar o próximos métodos
public:
    ///Declara a regressao Linear feita pelo Metodo dos Mínimos Quadrados

```

```

        void Reg(CRes*);
    ///Declara o metodo para pegar o valor de "a"
        double A()          const {return a;}
    ///Declara o metodo para pegar o valor de "b"
        double B()          const {return b;}

    //    double Media() const {return media;}

    ///Construtor default que inicializa a e b com valores iguais a zero
    CRegLin(): a(0.0), b(0.0) {}

    /*.....*/
    /*
    void Est(Cres*);
    double Media() const {return media;}
    double Maximo () const {return maximo;}
    double Minimo() const {return minimo;}
    */
};

#endif

```

Apresenta-se na listagem 5.3o arquivo de implementação da classe CRegLin.

Listing 5.3: Arquivo de implementação da classe CRegLin

```

#include "CRegLin.h"

#include <iostream>
///Metodo que fara a regressao linear e recebe um ponteiro que aponta
    para um atributo do tipo CRes
void CRegLin::Reg(CRes* parametro)
{
    ///declara o atributo que armazena o somatorio de x e defini-o igual a
    zero
        double sx = 0.0;
    ///declara o atributo que armazena o somatorio de y e defini-o igual a
    zero
        double sy = 0.0;
    ///declara o atributo que armazena o somatorio do produto de x*x e
    defini-o igual a zero
        double sxx = 0.0;
    ///declara o atributo que armazena o somatorio do produto de y*y e
    defini-o igual a zero
        double sxy = 0.0;
    ///define n iguaal ao tamanho do atributo parametro
        int n = parametro->N();

```

```

        //media=0.0;
    ///For utilizado para fazer o somatorio
        for (int i=0; i<n; i++)
        {
    ///faz o somatório de x sobrecarregando-o
            sx += parametro->X(i);
    ///faz o somatório de y sobrecarregando-o
            sy += parametro->Y(i);
    ///faz o somatório do produto x*x sobrecarregando-o
            sxx += parametro->X(i)*parametro->X(i);
    ///faz o somatório do produto y*y sobrecarregando-o
            sxy += parametro->X(i)*parametro->Y(i);

            //media+=parametro->P(i)/parametro->N();
        }
    ///Coeficiente Linear
        a = (sy-(n*sx*sxy-sx*sx*sy)/(n*sxx-sx*sx))/n;
    ///Coeficiente Angular
        b = (n*sxy-sx*sy)/(n*sxx-sx*sx);
    }

    /*
    void CRegLin::Est(CRes* parametro2)
    {
        double soma=0.0;
        media=0.0;
        minimo=0.0;
        maximo=0.0;
        for (int i=0; i<parametro2->N(); i++)
        {
            soma+=parametro2->P(i);
        }
        media=soma/parametro2->N();

        for (int i=0; i<parametro2->N(); i++)
        {
            if (maximo<parametro2->P(i))
            { maximo=parametro2->p(i);}
        }

        minimo=maximo;
        for (int i=0; i<parametro2->N(); i++)
        {
            if (minimo>parametro2->P(i))
            { minimo=parametro2->p(i);}
        }
    }
}

```

*/

5.1.3 CResGas

Apresenta-se na listagem o arquivo com código da classe CResGas.

Listing 5.4: Arquivo de cabeçalho da classe CResGas

```

///Classe Derivada de CRes
///Possui os dados iniciais de Reservatorio de Gas
///Classes Derivadas: CGasSecoPres, CResGasSecoInf,
    CResGasSecoInfPotAQUIfer, CResGasSecoInfSchilthuis,
    CResGasCondNaoRetro e CResGasCondRetro

#ifndef CRESGAS_H
#define CRESGAS_H

#include <iostream>
#include <cmath>
#include <vector>
#include "CRes.h"

///Heranca Simples
class CResGas: public CRes
{
    ///Protected permite as classes derivadas usar os atributos declarados
    aqui
protected:
    ///Declara o vetor que armazena o volume de gas produzido
    std::vector<double> gp;
    ///Declara o vetor que armazena o fator de compressibilidade do gas
    std::vector<double> z;
    ///Torna publico os proximos metodos
public:
    ///Declara o metodo calculo (polimorfismo)
    virtual void Calc();
    ///Declara o metodo para pegar os valores de gas produzido
    double Gp(int i) const {return gp[i];}
    ///Declara o metodo para pegar os valores do fator de compressibilidade
    double Z(int i) const {return z[i];}

    ///Construtor default da classe
    CResGas();
};

#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResGas.

Listing 5.5: Arquivo de implementação da classe CResGas

```

#include <iostream>
#include "CResGas.h"

using namespace std;

///Define o metodo de calculo da classe
void CResGas::Calc()
{
///Atribui os valores calculados nos vetores X e Y para este tipo de reservatorio
    for (int i=0; i<n; i++)
    {
        x[i] = gp[i];
        y[i] = p[i]/z[i];
    }
}

///Define o construtor
CResGas::CResGas()
{
    n=0;

    cout << "\nNumero de medicoes:"; //
        Numero de dados(medicoes) que temos
    cin >> n;
    cin.get();
    x.resize(n); //
        Dimensiona como n o numero de valores
    y.resize(n); //para os
        vetores X, Y, GP, P e Z
    gp.resize(n);
    p.resize(n);
    z.resize(n);

    for (int i = 0; i < n; i++)
        //Corre de 0 a n, pegando os valores de P, GP e Z
    {
        cout << "\nPressao (" << i+1 << ") (kgf/cm2):";
        cin >> p[i];
        cin.get();
        cout << "\nGas Produzido (" << i+1 << ") (m3 std):";
        cin >> gp[i];
        cin.get();
        cout << "\nZ (" << i+1 << "):";
        cin >> z[i];
        cin.get();
    }
}

```



```

    }
}

```

5.1.4 CResGasSecoPres

Apresenta-se na listagem o arquivo com código da classe CResGasSecoPres.

Listing 5.6: Arquivo de cabeçalho da classe CResGasSecoPres

```

///Classe herdeira de CResGas para Reservatorio de Gas seco anormalmente
pressurizado

#ifndef CRESGASSECOPRES_H
#define CRESGASSECOPRES_H

#include <cmath>
#include <iostream>
#include "CResGas.h"

///Heranca Simples
class CResGasSecoPres: public CResGas
{
///Protected permite as classes derivadas usar os atributos declarados
aqui
protected:
    double cw, cf, swi;

///Torna publico os proximos metodos
public:
///Declara o metodo calculo (polimorfismo)
    void Calc();
///Declara o metodo para pegar o valor da saturacao inicial de agua
    float Swi() const {return swi;}
///Declara o metodo para pegar o valor da compressibilidade da agua
    double Cw() const {return cw;}
///Declara o metodo para pegar o valor da compressibilidade da formacao
    double Cf() const {return cf;}

///Construtor default da classe
    CResGasSecoPres();
};
#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResGasSecoPres.

Listing 5.7: Arquivo de implementação da classe CResGasSecoPres

```

#include <iostream>
#include "CResGasSecoPres.h"

```

```

using namespace std;

///Define o metodo de calculo da classe
void CResGasSecoPres::Calc()
{
    double cewf = (cw*swi + cf)/(1 - swi);
                                ///Define as variaveis da
                                Equacao de Balanco de Materiais (EBM): Cewf
    double pi = p[0];
///atribui os valores dos vetores X e Y para linearizacao da EBM do
mecanismo de producao presente
    for ( int i=0; i<n; i++ )
    {
        x[i] = gp[i];
        y[i] = (p[i]/z[i])*(1.0-cewf*(pi-p[i]));
    }
}

///Define o construtor
CResGasSecoPres::CResGasSecoPres()
{
///pede e pega os valores de swi, cw e cf
    cout << "Saturacao_Inicial_de_Agua:_";
    cin >> swi;
    cin.get();
    cout << "Compressibilidade_da_agua_((kgf/cm2)^-1):_";
    cin >> cw;
    cin.get();
    cout << "Compressibilidade_da_formacao_((kgf/cm2)^-1):_";
    cin >> cf;
    cin.get();
}

```

5.1.5 CResGasSecoInf

Apresenta-se na listagem o arquivo com código da classe CResGasSecoInf.

Listing 5.8: Arquivo de cabeçalho da classe CResGasSecoInf

```

///Classe herdeira de CResGas para Reservatorio de Gas Seco Sob Influxo
de Agua

#ifndef CRESGASSECOINF_H
#define CRESGASSECOINF_H

#include <cmath>

```

```

#include <iostream>
#include "CResGas.h"

//Heranca Simples
class CResGasSecoInf: public CResGas
{
    //Protected permite as classes derivadas usar os atributos declarados
    aqui
protected:
    //Declara os vetores que armazenam os valores de fator volume-formacao
    do gas e influxo de agua
    std::vector<double> bg, we;
    //Declara a variável que representa o fator volume-formacao inicial do
    gas
    double bgi;

    //Torna publico os proximos metodos
public:
    //Declara o metodo calculo (polimorfismo)
    void Calc();
    //Declara o metodo para pegar os valores do fator volume-formacao do
    gas
    double Bg(int i) const {return bg[i];}
    //Declara o metodo para pegar os valores de influxo de agua
    double We(int i) const {return we[i];}
    //Declara o metodo para pegar o valor inicial do fator volume-formacao
    double Bgi() const {return bgi;}

    //Construtor default da classe
    CResGasSecoInf();

};
#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResGasSecoInf.

Listing 5.9: Arquivo de implementação da classe CResGasSecoInf

```

#include <iostream>
#include "CResGasSecoInf.h"

using namespace std;

//Define o metodo de calculo da classe
void CResGasSecoInf::Calc()
{
    //atribui os valores dos vetores X e Y para linearizacao da EBM do
    mecanismo de producao presente
    for (int i=0; i<n; i++)

```

```

        {
            x[i] = we[i]/(bg[i] - bgi);
            y[i] = gp[i]*bg[i]/(bg[i] - bgi);
        }
    }

CResGasSecoInf::CResGasSecoInf()
{
    bg.resize(n);

    //
    Dimensiona como n o numero de valores
    we.resize(n);

    //para
    os vetores bg e we
    cout << "Bgi_(m3/m3std):_";
    cin >> bgi;
    cin.get();
    for (int i=0; i<n; i++)

    //Corre de 0 a n
    , pegando os valores de bg e we
    {
        cout << "Bg_(" << i+1 << ")(m3/m3std):_";
        cin >> bg[i];
        cin.get();
        cout << "Influxo_Acumulado_We_(" << i+1 << ")(m3):_";
        cin >> we[i];
        cin.get();
    }
}

```

5.1.6 CResGasSecoInfPotAquifer

Apresenta-se na listagem o arquivo com código da classe CResGasSecoInfPotAquifer.

Listing 5.10: Arquivo de cabeçalho da classe CResGasSecoInfPotAquifer

///Classe herdeira de CResGas para Reservatorio com o modelo de influxo Pot Aquifer

```

#ifndef CRESGASSECOINFPOTAQUIFER_H
#define CRESGASSECOINFPOTAQUIFER_H
#include <cmath>
#include <vector>
#include <iostream>
#include "CResGas.h"

```

```

///Heranca Simples
class CResGasSecoInfPotAquifer: public CResGas
{
    ///Protected permite as classes derivadas usar os atributos
    declarados aqui
protected:
    ///Declara a variavel que representa o valor do influxo e do
    fator volume formacao do gas
    std::vector<double> we, bg;
    ///Declara a variavel que representa o fator volume formacao
    inicial do gas
    double bgi;
    ///Declara a variavel que representa a porosidade do aquifero
    double fi;
    ///Declara a variavel que representa a altura do aquifero
    double h;
    ///Declara a variavel que representa o raio do aquifero
    double re;
    ///Declara a variavel que representa o raio do reservatorio
    double ro;
    ///Declara a variavel que representa a compressibilidade da
    água
    double cw;
    ///Declara a variavel que representa a compressibilidade dos
    poros
    double cf;
    ///Declara a variavel que representa a diferenca de pressao
    delta p
    std::vector<double> deltap;

    ///Torna publico os proximos metodos
public:
    ///Declara o metodo calculo (polimorfismo)
    void Calc();

    ///Declara o metodo para pegar os valores da porosidade do
    aquifero
    double Fi () const {return fi;}

    ///Declara o metodo para pegar os valores da porosidade do
    aquifero
    double Bgi () const {return bgi;}

    ///Declara o metodo para pegar o valor da altura do aquifero
    double H () const {return h;}

    ///Declara o metodo para pegar os valores do raio do aquifero

```

```

double Re () const {return re;}

///Declara o metodo para pegar os valores do raio do reservatorio
double Ro () const {return ro;}

///Declara o metodo para pegar os valores da compressibilidade da
agua
double Cw () const {return cw;}

///Declara o metodo para pegar os valores da compressibilidade
dos poros
double Cf () const {return cf;}

///Declara o metodo para pegar os valores do deltap
double Deltap (int i) const {return deltap[i];}

///Declara o metodo para pegar os valores do influxo
double We (int i) const {return we[i];}

///Declara o metodo para pegar os valores do fator volume
formacao do gas
double Bg (int i) const {return bg[i];}

///Construtor default da classe
CResGasSecoInfPotAquifer();

};

#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResGasSecoInfPotAquifer.

Listing 5.11: Arquivo de implementação da classe CResGasSecoInfPotAquifer

```

#include <iostream>
#include "CResGasSecoInfPotAquifer.h"

using namespace std;

///Define o metodo de calculo da classe
void CResGasSecoInfPotAquifer::Calc()
{
///atribui os valores dos vetores X e Y para linearizacao da EBM do
mecanismo de producao presente

    deltap.resize(n); ///Dimensiona como n o
    numero de valores para os vetores deltap e we
    we.resize(n);

```

```

        double wi = 3.14159*(re*re - ro*ro)*h*fi;           //Calcula o
            valor do influxo de água inicial

    for (int i=0; i<n; i++)
    {
        deltap[i] = p[0] - p[i];                             //
            Calcula variação de pressão
        we[i] = (cw+cf)*wi*deltap[i];                         //
            Calcula o influxo de água acumulado

    }

    for (int i=0; i<n; i++)
    {
        x[i] = we[i]/(bg[i] - bgi);
        y[i] = gp[i]*bg[i]/(bg[i] - bgi);
    }

}

///Define o construtor
CResGasSecoInfPotAquifer::CResGasSecoInfPotAquifer()
{

    cout << "Porosidade_: ";
        cin >> fi;
        cin.get();

    cout << "bgi_(m3/m3_std): ";
        cin >> bgi;
        cin.get();

    cout << "Espessura_do_aquifero_-h_(m): ";
        cin >> h;
        cin.get();

    cout << "Raio_do_aquifero_-re_(m): ";
        cin >> re;
        cin.get();

    cout << "Raio_do_reservatorio_-ro_(m): ";
        cin >> ro;
        cin.get();

    cout << "Compressibilidade_da_agua_-Cw_((kgf/cm2)^-1): ";
        cin >> cw;

```

```

        cin.get();

        cout << "Compressibilidade da formacao - Cf ((kgf/cm2)^-1): ";
        cin >> cf;
        cin.get();

        bg.resize(n);

        for (int i=0; i<n; i++)
        {
            cout << "\nbg(" << i+1 << ") (m3/m3std): ";
            cin >> bg[i];
            cin.get();
        }
    }
}

```

5.1.7 CResGasSecoInfSchilthuis

Apresenta-se na listagem o arquivo com código da classe CResGasSecoSchilthuis.

Listing 5.12: Arquivo de cabeçalho da classe CResGasSecoInfSchilthuis

```

///Classe herdeira de CResGas para Reservatorio com o modelo de influxo
Schilthuis

#ifndef CRESGASSECOINFSCHILTHUIS_H
#define CRESGASSECOINFSCHILTHUIS_H
#include <cmath>
#include <iostream>
#include <vector>
#include "CResGas.h"

///Heranca Simples
class CResGasSecoInfSchilthuis: public CResGas
{
    ///Protected permite as classes derivadas usar os atributos
    declarados aqui
protected:
    ///Declara a variavel que representa o valor do influxo e do
    fator volume formacao do gas
    std::vector<double> we, bg;
    ///Declara a variavel que representa a diferenca de pressao
    delta p

```



```
std::vector<double> deltap;  
//Declara a variavel que representa o fator volume formacao  
    inicial do gas  
double bgi;  
//Declara a variavel que representa a constante de influxo  
double j;  
//Declara a variavel que representa os valores de t1 e t2 =  
    limites fixos de integracao  
double t1, t2;  
  
    ///Torna publico os proximos metodos  
public:  
  
    ///Declara o metodo calculo (polimorfismo)  
    void Calc();  
  
    ///Declara o metodo para pegar o valor do fator volume formacao  
    inicial do gas  
    double Bgi () const {return bgi;}  
  
    ///Declara o metodo para pegar o valor da constante de influxo  
    double J () const {return j;}  
  
    ///Declara o metodo para pegar o valor do limite fixo de  
    integracao t1  
    double T1 () const {return t1;}  
  
    ///Declara o metodo para pegar o valor do limite fixo de  
    integracao t2  
    double T2 () const {return t2;}  
  
    ///Declara o metodo para pegar os valores do deltap  
    double Deltap (int i) const {return deltap[i];}  
  
    ///Declara o metodo para pegar os valores do influxo  
    double We (int i) const {return we[i];}  
  
    ///Declara o metodo para pegar os valores do fator volume  
    formacao do gas  
    double Bg (int i) const {return bg[i];}  
  
    ///Construtor default da classe
```

```

CResGasSecoInfSchilthuis();

};

#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResGasSecoInfSchilthuis.

Listing 5.13: Arquivo de implementação da classe CResGasSecoInfSchilthuis

```

#include <iostream>
#include "CResGasSecoInfSchilthuis.h"

using namespace std;

///Define o metodo de calculo da classe
void CResGasSecoInfSchilthuis::Calc()
{
    deltap.resize(n); //Dimensiona como
    n o numero de valores para os vetores deltap e we
    we.resize(n);

    for (int i=0; i<n; i++)
    {
        deltap[i] = p[0] - p[i];
        we[i] = j*deltap[i]*(t2-t1);
    }

    for (int i=0; i<n; i++)
    {
        x[i] = we[i]/(bg[i] - bgi);
        y[i] = gp[i]*bg[i]/(bg[i] - bgi);
    }
}

///Define o construtor
CResGasSecoInfSchilthuis::CResGasSecoInfSchilthuis()
{

    cout << "t1_␣(d):_␣";
    cin>>t1;
    cin.get();

    cout << "t2_␣(d):_␣";
    cin>>t2;
    cin.get();

    cout << "Constante_␣de_␣influxo_␣-_␣J_␣(m3/(d*kgf/cm2)_␣:_␣";
    cin>>j;

```

```

        cin.get();

        cout << "bgi_(m3/m3_std):_";
        cin >> bgi;
        cin.get();

    bg.resize(n);

    for (int i=0; i<n; i++)
    {
        cout << "\nbg_(" << i+1 << ")_(m3/m3_std):_";
        cin >> bg[i];
        cin.get();
    }

}

```

5.1.8 CResGasCondNaoRetro

Apresenta-se na listagem o arquivo com código da classe CResGasCondNaoRetro.

Listing 5.14: Arquivo de cabeçalho da classe CResGasCondNaoRetro

```

///Classe herdeira de CResGas para Reservatorio de Gas Condensado nao
Retrogrado

#ifndef CRESGASCONDNAORETRO_H
#define CRESGASCONDNAORETRO_H

#include <cmath>
#include <iostream>
#include <vector>
#include "CResGas.h"

///Heranca Simples
class CResGasCondNaoRetro: public CResGas
{
    ///Protected permite as classes derivadas usar os atributos declarados
    aqui
protected:
    ///Declara os vetores que armazenaram os valores de volume de gas
    produzido parcial, volume de gas equivalente, volume de vapor de agua
    equivalente
    ///a agua produzida, volume total de gas produzido, volume de agua
    produzida, volume de gas condensado produzido
    std::vector<double> gpp, gec, gew, gpt, vw, vc;

```

```

///Declara a variável que representa o fator volume de formacao inicial
do gas, densidade do condensado produzido, massa molecular do
condensado produzido
    double bgi, dc, mc;

///Torna publico os proximos metodos
public:
///Declara o metodo calculo (polimorfismo)
void Calc();
///Declara o metodo para pegar os valores do volume de gas produzido
parcial
double Gpp(int i) const {return gpp[i];}
///Declara o metodo para pegar os valores do volume de gas equivalente
double Gec(int i) const {return gec[i];}
///Declara o metodo para pegar os valores do volume de vapor de agua
equivalente a agua produzida
double Gew(int i) const {return gew[i];}
///Declara o metodo para pegar os valores do volume total de gas
produzido
double Gpt(int i) const {return gpt[i];}
///Declara o metodo para pegar os valores de agua produzida
double Vw(int i) const {return vw[i];}
///Declara o metodo para pegar os valores de gas condensado produzido
double Vc(int i) const {return vc[i];}
///Declara o metodo para pegar o valor inicial do fator volume de
formacao
double Bgi() const {return bgi;}
///Declara o metodo para pegar a densidade do condensado produzido
double Dc() const {return dc;}
///Declara o metodo para pegar a massa molecular do condensado produzido
double Mc() const {return mc;}

///Construtor default da classe
CResGasCondNaoRetro();

};
#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResGasCondNaoRetro.

Listing 5.15: Arquivo de implementação da classe CResGasCondNaoRetro

```

#include <iostream>
#include "CResGasCondNaoRetro.h"

using namespace std;

///Define o metodo de calculo da classe
void CResGasCondNaoRetro::Calc()
{

```

```

        // Redefine os tamanhos dos vetores para o numero de medicoes
        escolhida pelo usuario
        gpp.resize(n);
        gec.resize(n);
        gew.resize(n);
        gpt.resize(n);

        //percorre o vetor atribuido em cada posicao da medicao os
        valores calculados
        for (int i=0; i<n; i++)
        {
            gpp[i] = gp[i];

            gec[i] = ((23686 * dc)/mc)*vc[i];

            gew[i] = 1316 * vw[i];

            gpt[i] = gpp[i] + gec[i] + gew[i];

            // recalcula gp assumindo os novos valores de gpt para ser
            plotado no grafico
            x[i] = gpt[i];
            y[i] = p[i]/z[i];
        }
    }

    ///Define o construtor
    CResGasCondNaoRetro::CResGasCondNaoRetro()
    {

        cout << "\nDc: ";
        cin >> dc;
        cin.get();

        cout << "\nMc: ";
        cin >> mc;
        cin.get();

        cout << "\nBgi(m3/m3std): ";
        cin >> bgi;
        cin.get();

        vc.resize(n);
        vw.resize(n);

        for (int i=0; i<n; i++)

```

```

        {
            cout << "\nVc_(" << m3 << ")_(" << i+1 << "):_";
            cin >> vc[i];
            cin.get();

            cout << "\nVw_(" << m3 << ")_(" << i+1 << "):_";
            cin >> vw[i];
            cin.get();

        }
    }
}

```

5.1.9 CResGasCondRetro

Apresenta-se na listagem o arquivo com código da classe CResGasCondRetro.

Listing 5.16: Arquivo de cabeçalho da classe CResGasCondRetro

```

///Classe herdeira de CResGas para Reservatorio de Gas Condensado
Retrogrado

#ifndef CRESGASCONDRETRO_H
#define CRESGASCONDRETRO_H

#include <cmath>
#include <iostream>
#include <vector>
#include "CResGas.h"

///Heranca Simples
class CResGasCondRetro: public CResGas
{
    ///Protected permite as classes derivadas usar os atributos declarados
    aqui
protected:
    ///Declara os vetores que armazenam os valores para z de duas fases,
    saturacao de liquido condensado dentro do reservatorio, numero de
    mols de gas
    ///por volume unitario de fluido no reservatorio e numero de mols de
    liquido por volume unitario de fluido no reservatorio
    std::vector<double> z2f, sl, n1, n2;
    ///Declara a variável que representa o fator volume de formacao inicial
    do gas, temperatura do gas no reservatorio
    ///densidade do condensado no reservatorio, massa molecular do condensado
    no reservatorio
    double bgi, temp, dl, ml;

    ///Torna publico os proximos metodos

```

```

public:
    ///Declara o metodo calculo (polimorfismo)
    void Calc();
    ///Declara o metodo para pegar os valores de z de duas fases
    double Z2f(int i) const {return z2f[i];}
    ///Declara o metodo para pegar os valores de saturacao de liquido
        condensado dentro do reservatorio
    double Sl(int i) const {return sl[i];}
    ///Declara o metodo para pegar os valores do numero de mols de gas por
        volume unitario de fluido no reservatorio
    double N1(int i) const {return n1[i];}
    ///Declara o metodo para pegar os valores do numero de mols de liquido
        por volume unitario de fluido no reservatorio
    double N2(int i) const {return n2[i];}
    ///Declara o metodo para pegar o valor inicial do fator volume de
        formacao
    double Bgi() const {return bgi;}
    ///Declara o metodo para pegar o valor da temperatura no reservatorio
    double Temp() const {return temp;}
    ///Declara o metodo para pegar a densidade do condensado no reservatorio
    double Dl() const {return dl;}
    ///Declara o metodo para pegar a massa molecular do condensado no
        reservatorio
    double Ml() const {return ml;}

    ///Construtor default da classe
    CResGasCondRetro();

};
#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResGasSecoCondRetro.

Listing 5.17: Arquivo de implementação da classe CResGasCondRetro

```

#include <iostream>
#include "CResGasCondRetro.h"

using namespace std;

///Define o metodo de calculo da classe
void CResGasCondRetro::Calc()
{
    // Redefine os tamanhos dos vetores para o numero de medicoes
        escolhida pelo usuario
        z2f.resize(n);

    //percorre o vetor atribuido em cada posicao da medicao os
        valores calculados

```

```

        for (int i=0; i<n; i++)
        {
            n1[i] = (p[i]*(1-sl[i]))/(z[i]*0.08478*temp);

            n2[i] = (1000* dl * sl[i])/ml;

            z2f[i] = p[i]/(0.08478 * temp *(n1[i]+n2[i]));

            // assume os novos valores de z2f para ser plotado no grafico
            x[i] = gp[i];
            y[i] = p[i]/z2f[i];
        }
    }

    ///Define o construtor
    CResGasCondRetro::CResGasCondRetro()
    {

        cout << "\nBgi_␣(m3/m3std):␣";
            cin >> bgi;
            cin.get();

        cout << "\nTemperatura_␣(K):␣";
            cin >> temp;
            cin.get();

        cout << "\nDl:␣";
            cin >> dl;
            cin.get();

        cout << "\nMl:␣";
            cin >> ml;
            cin.get();

            sl.resize(n);
            n1.resize(n);
            n2.resize(n);

        for (int i=0; i<n; i++)
        {
            cout << "\nSl_␣(" << i+1 << "):␣";
                cin >> sl[i];
                cin.get();

        }
    }

```



```
}
```

5.1.10 CResOil

Apresenta-se na listagem o arquivo com código da classe CResOil.

Listing 5.18: Arquivo de cabeçalho da classe CResOil

```
///Classe Derivada de CRes
///Possui os dados iniciais de Reservatorio de Óleo
///Classes Derivadas: COilCapa e COilInf

#ifndef CRESOIL_H
#define CRESOIL_H

#include <iostream>
#include <cmath>
#include <vector>
#include "CRes.h"

///Heranca Simples
class CResOil: public CRes
{
///Protected permite as classes derivadas usar os atributos declarados aqui
protected:
///Declara o vetor que armazena o volume de oleo produzido
    std::vector<double> np;
///Declara o vetor que armazena o volume do fator volume de formacao
    std::vector<double> bo;
///declara os atributos que armazenaram os valores correspondentes a copressibilidade da agua e da formacao, saturacao inicial de agua, pressao inicial e fator volume de formacao inicial do oleo, respectivamente
    double cw, cf, swi, pi, boi;

///Torna publico os proximos metodos
public:
///Declara o metodo calculo (polimorfismo)
    virtual void Calc();
///Declara o metodo para pegar os valores de oleo produzido
    double Np(int i) const {return np[i];}
///Declara o metodo para pegar os valores dos fatores de formacao do oleo
    double Bo(int i) const {return bo[i];}
///Declara o metodo para pegar o valor da saturacao inicial de agua
    double Swi() const {return swi;}
///Declara o metodo para pegar o valor da compressibilidade da agua
```

```
double Cw() const {return cw;}
///Declara o metodo para pegar o valor da compressibilidade da formacao
double Cf() const {return cf;}
///Declara o metodo para pegar o valor da pressao incial
double Pi() const {return pi;}
///Declara o metodo para pegar o valor do fator volume de formacao
    inicial do oleo
double Boi() const {return boi;}

///Construtor default da classe
CResOil();
};

#endif
```

Apresenta-se na listagem o arquivo de implementação da classe CResOil.

Listing 5.19: Arquivo de implementação da classe CResOil

```
#include <iostream>
#include "CResOil.h"

using namespace std;

///Define o metodo de calculo da classe
void CResOil::Calc()
{
    double f;
    double eo;
    double ewf;
    for (int i=0; i<n; i++)
    {
        f = np[i] * bo[i];
        eo = bo[i] - boi;
        ewf = boi * (pi - p[i]) * ( (cw*swi + cf) / (1-swi) );
///Atribui os valores calculados nos vetores X e Y para este tipo de
    reservatorio
        x[i] = eo + ewf;
        y[i] = f;
    }
}

///Define o construtor
CResOil::CResOil()
{
    cout << "\nNumero de medicoes:";
    //Numero de dados(
    medicoes) que temos
```

```

cin >> n;
cin.get();
x.resize(n);

    //Dimensiona como n o numero de valores
y.resize(n);

    //para os vetores X, Y, NP, P e Bo
np.resize(n);
p.resize(n);
bo.resize(n);
cout << "Pressao_Inicial_(kgf/cm2):_";
cin >> pi;
cin.get();
cout << "Boi_(m3/m3_std):_";

                                                                    //Pega o valor de
    B0 inicial
cin >> boi;
cin.get();
cout << "Saturacao_Inicial_de_Agua:_";

                                                                    //Pega o valor de Sw
    inicial
cin >> swi;
cin.get();
cout << "Compressibilidade_da_agua_((kgf/cm2)^-1):_";

                                                                    //Pega o valor de cw
cin >> cw;
cin.get();
cout << "Compressibilidade_da_formacao_((kgf/cm2)^-1):_"
    ;                                                                    //Pega o valor de cf
cin >> cf;
cin.get();
for (int i = 0; i<n; i++)

                                                                    //Corre de 0 a
    n, pegando os valores de P, NP e Bo
{
    cout << "Pressao_(" << i+1 << ")(kgf/cm2):_";
    cin >> p[i];
    cin.get();
    cout << "Oleo_Produzido_(" << i+1 << ")(m3_std):_";
    cin >> np[i];
    cin.get();
    cout << "Bo_(" << i+1 << ")(m3/m3_std):_";
    cin >> bo[i];
    cin.get();
}
}

```

5.1.11 CResOilCapa

Apresenta-se na listagem o arquivo com código da classe CResOilCapa.

Listing 5.20: Arquivo de cabeçalho da classe CResOilCapa

```
///Classe herdeira de CResOil para Mecanismo de Capa de Gas

#ifndef CRESOILCAPA_H
#define CRESOILCAPA_H

#include <cmath>
#include <iostream>
#include "CResOil.h"

///Heranca Simples
class CResOilCapa: public CResOil
{
///Protected permite as classes derivadas usar os atributos declarados aqui
protected:
///Declara os vetores que armazenaram os valores de rp, rs, bg
    std::vector<double> rp, rs, bg;
///declara os atributos que armazenaram os valores correspondentes a rsi e bgi
    double rsi, bgi;

///Torna publico os proximos metodos
public:
///Declara o metodo calculo (polimorfismo)
    void Calc();
///Declara o metodo para pegar os valores de rp
    double Rp(int i) const {return rp[i];}
///Declara o metodo para pegar os valores de rs
    double Rs(int i) const {return rs[i];}
///Declara o metodo para pegar os valores de fator volume de formacao do gas
    double Bg(int i) const {return bg[i];}
///Declara o metodo para pegar os valores de rasao de solubilidade do gas no oleo
    double Rsi() const {return rsi;}
///Declara o metodo para pegar os valores de fator de formacao do gas
    double Bgi() const {return bgi;}

///Construtor default da classe
    CResOilCapa();
};
#endif
```

Apresenta-se na listagem o arquivo de implementação da classe CResOilCapa.

Listing 5.21: Arquivo de implementação da classe CResOilCapa

```

#include <iostream>
#include "CResOilCapa.h"

using namespace std;

///Defini o metodo de calculo da classe
void CResOilCapa::Calc()
{
    double eg;

    ///Define as variaveis da Equacao de Balanco de Materiais (EBM):
    double eo;

    ///eg (expansao do gas), eo (expansao do oleo), f
    double f;
///atribui os valores dos vetores X e Y para linearizacao da EBM do mecanismo de producao presente
    for ( int i=0; i<n; i++ )
    {
        eg = (boi*(bg[i]/bgi-1));
        eo = (bo[i]-boi+(rsi-rs[i])*bg[i]);
        f = np[i]*(bo[i]+(rp[i]-rs[i])*bg[i]);
        x[i] = eg/eo;
        y[i] = f/eo;
    }
}

///Define o construtor
CResOilCapa::CResOilCapa()
{
    rp.resize(n);

    Dimensiona como n o numero de valores
    rs.resize(n);

    para os vetores rs, rp, bg
    bg.resize(n);
    cout << "Bgi_(m3/m3_std):_";
    cin >> bgi;
    cin.get();
    cout << "Rsi_(m3/m3_std):_";
    cin >> rsi;
    cin.get();
    cout << "." << endl;
    cin.get();

```

```

///percorre os vetores rp, rs, bg e we pedindo e pegando seus valores
correspondentes
    for (int i = 0; i<n; i++)
    {
        cout << "Rp_" << i+1 << ")(m3_std/m3_std):_";
        cin >> rp[i];
        cin.get();
        cout << "Rs_" << i+1 << ")(m3/m3_std):_";
        cin >> rs[i];
        cin.get();
        cout << "Bg_" << i+1 << ")(m3/m3_std):_";
        cin >> bg[i];
        cin.get();
    }
}

```

5.1.12 CResOilInf

Apresenta-se na listagem o arquivo com código da classe CResOilInf.

Listing 5.22: Arquivo de cabeçalho da classe CResOilInf

```

///Classe herdeira de CResGas para Reservatorio de Oleo Sob Influxo de
Agua

#ifndef CRESOILINF_H
#define CRESOILINF_H

#include <cmath>
#include <iostream>
#include "CResOil.h"

///Herança Simples
class CResOilInf: public CResOil
{
    ///Protected permite as classes derivadas usar os atributos declarados
    aqui
protected:
    ///Declara os vetores que armazenaram os valores de rp, rs, bg e we
        std::vector<double> we;
        std::vector<double> rp, rs, bg;
    ///declara os atributos que armazenaram os valores correspondentes a m,
    bgi, rsi
        double m, bgi, rsi;

    ///Torna puplico os proximos metodos
public:

```

```

///Declara o metodo calculo (polimorfismo)
void Calc();
///Declara o metodo para pegar os valores de m
double M() const {return m;}
///Declara o metodo para pegar os valores de fator volume de formacao do
gas inicial
double Bgi() const {return bgi;}
///Declara o metodo para pegar os valores de rasao de solubilidade
inicial
double Rsi() const {return rsi;}
///Declara o metodo para pegar os valores do volume do influxo dagua
double We(int i) const {return we[i];}
///Declara o metodo para pegar os valores de rp
double Rp(int i) const {return rp[i];}
///Declara o metodo para pegar os valores de rasao de solubilidade
double Rs(int i) const {return rs[i];}
///Declara o metodo para pegar os valores de bg
double Bg(int i) const {return bg[i];}

///Construtor default da classe
CResOilInf();
};
#endif

```

Apresenta-se na listagem o arquivo de implementação da classe CResOilInf.

Listing 5.23: Arquivo de implementação da classe CResOilInf

```

#include <iostream>
#include "CResOilInf.h"

using namespace std;

///Defini o metodo de calculo da classe
void CResOilInf::Calc()
{
    double eg;

    ///Define as variaveis da Equacao de Balanco de Materiais
    (EBM):
    double eo;

    ///eg (expansao do gas), eo (expansao do oleo), f
    double f;
///atribui os valores dos vetores X e Y para linearizacao da EBM do
mecanismo de producao presente
    for (int i=0; i<n; i++)
    {
        eg = (boi*(bg[i]/bgi-1));
        eo = (bo[i]-boi+(rsi-rs[i])*bg[i]);
    }
}

```

```

        f = np[i]*(bo[i]+(rp[i]-rs[i])*bg[i]);
        x[i] = we[i] / (eo + m*eg);
        y[i] = f/ (eo + m*eg);
    }
}

///Define o construtor
CResOilInf::CResOilInf()
{
    rp.resize(n);
    Dimensiona como n o numero de valores
    rs.resize(n);
    para os vetores rs, rp, bg, we
    bg.resize(n);
    we.resize(n);
    cout << "Bgi_(m3/m3_std):_";
    cin >> bgi;

    //Pega o valor de Bg inicial
    cin.get();
    cout << "Rsi_(m3_std/m3_std):_";
    cin >> rsi;

    //Pega o valor de rs inicial
    cin.get();
    cout << "m_(m3/m3):_";
    cin >> m;

    //Pega o valor de m inicial
    cin.get();
    cout << "." << endl;
    cin.get();
///percorre os vetores rp, rs, bg e we pedindo e pegando seus valores
correspondentes
    for (int i=0; i<n; i++)
    {
        cout << "Rp_(" << i+1 << ")(m3_std/m3_std):_";
        cin >> rp[i];
        cin.get();
        cout << "Rs_(" << i+1 << ")(m3/m3_std):_";
        cin >> rs[i];
        cin.get();
        cout << "Bg_(" << i+1 << ")(m3/m3_std):_";
        cin >> bg[i];
        cin.get();
        cout << "Influxo_Acumulado_We_(" << i+1 << ")(m3):_";
    }
}

```



```

        cin >> we[i];
        cin.get();
    }
};

```

5.1.13 Main

Listing 5.24: Arquivo de implementação da função main()

```

#include <iostream>
#include "CRes.h"
#include "CResGas.h"
#include "CResGasSecoPres.h"
#include "CResGasSecoInf.h"
#include "CResGasSecoInfPotAQUIFER.h"
#include "CResGasSecoInfSchilthuis.h"
#include "CResGasCondNaoRetro.h"
#include "CResGasCondRetro.h"
#include "CResOil.h"
#include "CResOilCapa.h"
#include "CResOilInf.h"
#include "CRegLin.h"
#include "CGnuplot.h"

using namespace std;

int main()
{
    ///booleano para escolher tipo de fluido
    int escolha1;
    ///inicia-se o booleano como falso
    bool valido = false;
    ///loop para pegar resposta valida
    while (!valido)
    {

        cout << "Tipo de Fluido:\n(1) Gas\n(2) Oleo" << endl;
        cin >> escolha1;
        cin.get();
///escolha valida
        if (escolha1 == 1 || escolha1 == 2)
            valido = true;
        else
///escolha invalida
            cout << "Escolha invalida." << endl;
    }
}

```

```

///booleano para escolher tipo de mecanismo
int escolha2;
///inicia-se o booleano como falso
valido = false;
///loop para pegar resposta valida
while (!valido)
{
    cout << (escolha1 == 1 ? "\nTipo de mecanismo:\n(1)Volumetrico de Gas Seco\n(2)Anormalmente Pressurizado\n(3)Sob Influxo de Agua\n(4)Sob Influxo de agua Modelo Pot Aquifer\n(5)Sob Influxo de agua Modelo Schilthuis\n(6)Gas Condensado Nao Retrogrado\n(7)Gas Condensado Retrogrado" : "\nTipo de mecanismo:\n(1)Gas em solucao\n(2)Capa de Gas\n(3)Sob Influxo de Agua") << endl;
    cin >> escolha2;
    cin.get();
///escolha valida
    if (escolha2 == 1 || escolha2 == 2 || escolha2 == 3 || escolha2 == 4 || escolha2 == 5 || escolha2 == 6 || escolha2 == 7)
        valido = true;
    else
///escolha invalida
        cout << "\nEscolha invalida." << endl;
}

///cirar ponteiro res do tipo CRes
CRes* res;
std::string eixo_x;

std::string eixo_y;

if ( escolha1 == 1 )
{
    if (escolha2 == 1)
    {
///reservatorio volumetrico de gas seco
        eixo_x = "Gp(m3_std)";
        eixo_y = "P/Z(kgf/cm2)";
        res = new CResGas();
    }

    else if ( escolha2 == 2)
    {
///reservatorio de gas anormalmente pressurizado
        eixo_x = "Gp(m3_std)";
        eixo_y = "(P/Z)*(1-Cewf*dP)(kgf/cm2)";
    }
}

```

```

        res = new CResGasSecoPres();
    }

    else if ( escolha2 == 3)
    {
        ///reservatorio de gas sob influxo de agua
        eixo_x = "We/(Bg-Bgi)";
        eixo_y = "GpBg/(Bg-Bgi)";
        res = new CResGasSecoInf();
    }
}

if ( escolha1 == 1)
{
    if ( escolha2 == 4)
    {
        ///reservatorio de gas sob influxo de agua modelo Pot Aquifer
        eixo_x = "We/(Bg-Bgi)";
        eixo_y = "GpBg/(Bg-Bgi)";
        res = new CResGasSecoInfPotAquifer();
    }
}

if ( escolha1 ==1)
{
    if ( escolha2 == 5)
    {
        ///reservatorio de gas sob influxo de agua modelo
        Schilthuis
        eixo_x = "We/(Bg-Bgi)";
        eixo_y = "GpBg/(Bg-Bgi)";
        res = new CResGasSecoInfSchilthuis();
    }
}

if ( escolha1 ==1)
{
    if ( escolha2 == 6)
    {
        ///reservatorio de gas condensado nao retrogrado
        eixo_x = "Gpt(m3_std)";
        eixo_y = "(P/Z)(kgf/cm2)";
        res = new CResGasCondNaoRetro();
    }
}

```

```

        }
    }

    if ( escolha1 == 1)
    {
        if ( escolha2 == 7)
        {
            //reservatorio de gas condensado retrogrado
            eixo_x = "Gp(m3_std)";
            eixo_y = "(P/Z2f)(kgf/cm2)";
            res = new CResGasCondRetro();
        }
    }

    if ( escolha1 == 2 )
    {
        if (escolha2 == 1)
        {
            //reservatorio de oleo com gas em solucao
            eixo_x = "Eo+Ef,w(m3/m3_std)";
            eixo_y = "F(m3)";
            res = new CResOil;
        }
        else if ( escolha2 == 2)
        {
            //reservatorio de oleo com capa de gas
            eixo_x = "Eg/Eo(adm)";
            eixo_y = "F/Eo(m3_std)";
            res = new CResOilCapa;
        }
        else if ( escolha2 == 3)
        {
            //reservatorio de oleo sob influxo de agua
            eixo_x = "We/Eo(m3_std)";
            eixo_y = "F/Eo(m3_std)";
            res = new CResOilInf;
        }
    }

    cout << "\n" << eixo_x << "\t\t\t\t" << eixo_y << endl;

    //utilizar ponteiro res para chamar metodo Calc()
    res->Calc();

    //loop variando de 0 a n (numero de termos)

```

```

for (int i=0; i<res->N(); i++)
{
    ///mostrar valores dos vetores calculados
    cout << res->X(i) << "\t\t\t\t" << res->Y(i) << endl;
}

///
-----

///criar vetor coef do tipo CRegLin e obrigar iniciar o construtor da classe
CRegLin* coef = new CRegLin();
///usar ponteiro coef para chamar metodo Reg(res)
coef->Reg(res);

///calcular estimativa de reserva
if ( escolha1 == 1 )
{
    if (escolha2 == 1)
    ///Calculo de estimativa de reserva para reservatorio de gas volumetricamente seco
    {
        cout << "\n\nRegressao_Linear" << endl;
        cout << "G(m3/std)_=-a/b_=" << -coef->A()/coef->B() << "\n\n\n" << endl;
    }
    else if ( escolha2 == 2)
    {
///Calculo de estimativa de reserva para reservatorio de gas anormalmente pressurizado
        cout << "\n\nRegressao_Linear" << endl;
        cout << "G(m3/std)_=-a/b_=" << -coef->A()/coef->B() << "\n\n\n" << endl;
    }
    else if ( escolha2 == 3)
    {
///Calculo de estimativa de reserva para reservatorio de gas sob influxo de agua
        cout << "\n\nRegressao_Linear" << endl;
        cout << "G(m3/std)_=a_=" << coef->A() << "\n\n\n" << endl;
    }
    else if ( escolha2 == 4)
    {
///Calculo de estimativa de reserva para reservatorio de gas sob influxo de agua modelo Pot AQuifer
        cout << "\n\nRegressao_Linear" << endl;
        cout << "G(m3/std)_=a_=" << coef->A() << "\n\n\n" << endl;
    }
}

```

```

    }
    else if ( escolha2 == 5)
    {
///Calculo de estimativa de reserva para reservatorio de gas sob influxo
de agua modelo Schilthuis
        cout << "\n\nRegressao_Linear" << endl;
        cout << "G(m3std)=a=" << coef->A() << "\n\n" << endl;
    }
    else if ( escolha2 == 6)
    {
///Calculo de estimativa de reserva para reservatorio de gas condensado
nao retrogrado
        cout << "\n\nRegressao_Linear" << endl;
        cout << "G(m3std)=a/b=" << -coef->A()/coef->B() << "\n\n"
            << endl;
    }
    else if ( escolha2 == 7)
    {
///Calculo de estimativa de reserva para reservatorio de gas condensado
retrogrado
        cout << "\n\nRegressao_Linear" << endl;
        cout << "G(m3std)=a/b=" << -coef->A()/coef->B() << "\n\n"
            << endl;
    }

}
if ( escolha1 == 2 )
{
    if (escolha2 == 1)
    {
///Calculo de estimativa de reserva para reservatorio de oleo com gas em
solucao
        cout << "\n\nRegressao_Linear" << endl;
        cout << "N(m3std)=b=" << coef->B() << "\n\n" << endl;
    }
    else if ( escolha2 == 2)
    {
///Calculo de estimativa de reserva para reservatorio de oleo com capa
de gas
        cout << "\n\nRegressao_Linear" << endl;
        cout << "N(m3std)=a=" << coef->A() << "\n\n" << endl;
    }
    else if ( escolha2 == 3)
    {
///Calculo de estimativa de reserva para reservatorio de oleo sob
influxo de agua
        cout << "\n\nRegressao_Linear" << endl;
        cout << "N(m3std)=a=" << coef->A() << "\n\n" << endl;
    }
}

```

```
    }  
}  
  
    ///criar um objeto gnu do tipo Gnuplot  
    CGnuplot gnu;  
    ///reta formatada em linhas (lines, steps, points)  
    gnu.set_style("lines");  
  
    ///nomenclatura do eixo x recebe o nome dado acima  
    gnu.set_xlabel(eixo_x);  
    ///nomenclatura do eixo y recebe o nome dado acima  
    gnu.set_ylabel(eixo_y);  
    ///range do eixo x  
    gnu.set_xrange(res->X().front(), res->X().back());  
    ///plota grafico a partir dos coeficientes linear e angular  
    gnu.plot_slope(coef->B(), coef->A());  
  
    gnu.plot_x(res->P());  
  
    cin.get();  
    return 0;  
  
    system ("pause");  
}
```

Capítulo 6

Teste

Apresenta-se neste capítulo os testes realizados com o programa, com suas respectivas saídas de tela e os gráficos gerados pelo Gnuplot.

6.1 Reservatório de gás

6.1.1 Teste 1: volumétrico de gás seco

Neste caso será testado o simulador para um reservatório volumétrico de gás seco. Para isto utilizaremos os dados de um exemplo para calcular o volume de gás original.

No primeiro momento, escolhe-se “Gas” como o tipo de fluido do reservatório e “Volumetrico de Gas Seco” como tipo de mecanismo. O número de medições feitas foi onze. Os dados de pressão, gás produzido e coeficiente de compressibilidade (Z) seguem na tabela abaixo.

Tabela 6.1: Dados para Teste de Reservatório Volumétrico de Gás Seco

n	Pressão (kgf/cm ²)	Gp (m ³ std)	Z
1	805	0	1.496
2	712	81e6	1.397
3	651	152e6	1.330
4	603	220e6	1.280
5	556	287e6	1.230
6	519	341e6	1.192
7	482	411e6	1.154
8	449	455e6	1.122
9	410	516e6	1.084
10	352	611e6	1.033
11	293	696e6	0.998

Apresenta-se na listagem o arquivo com a saída de tela do teste 1.

Listing 6.1: Arquivo de saída de tela do teste 1

```

Tipo de Fluido:
(1) Gas
(2) Oleo
1

Tipo de mecanismo:
(1) Volumetrico de Gas Seco
(2) Anormalmente Pressurizado
(3) Sob Influxo de Agua
(4) Sob Influxo de agua Modelo Pot Aquifer
(5) Sob Influxo de agua Modelo Schilthuis
(6) Gas Condensado Nao Retrogrado
(7) Gas Condensado Retrogrado
1

Numero de medicoes:11

Pressao (1) (kgf/cm2):805

Gas Produzido (1)(m3 std):0

Z(1):1.496

Pressao (2) (kgf/cm2):712

Gas Produzido (2)(m3 std):81e6

```

Z(2):1.397

Pressao (3) (kgf/cm2):651

Gas Produzido (3)(m3 std):152e6

Z(3):1.330

Pressao (4) (kgf/cm2):603

Gas Produzido (4)(m3 std):220e6

Z(4):1.280

Pressao (5) (kgf/cm2):556

Gas Produzido (5)(m3 std):287e6

Z(5):1.230

Pressao (6) (kgf/cm2):519

Gas Produzido (6)(m3 std):341e6

Z(6):1.192

Pressao (7) (kgf/cm2):482

Gas Produzido (7)(m3 std):411e6

Z(7):1.154

Pressao (8) (kgf/cm2):449

Gas Produzido (8)(m3 std):455e6

Z(8):1.122

Pressao (9) (kgf/cm2):410

Gas Produzido (9)(m3 std):516e6

Z(9):1.084

Pressao (10) (kgf/cm2):352

Gas Produzido (10)(m3 std):611e6

Z(10):1.033

Pressao (11) (kgf/cm2):293

Gas Produzido (11)(m3 std):696e6

Z(11):0.998

Gp(m3 std)	P/Z(kgf/cm2)
0	538.102
8.1e+07	509.664
1.52e+08	489.474
2.2e+08	471.094
2.87e+08	452.033
3.41e+08	435.403
4.11e+08	417.678
4.55e+08	400.178
5.16e+08	378.229
6.11e+08	340.755
6.96e+08	293.587

Regressao Linear

G (m3 std) = -a/b = 1.63736e+09

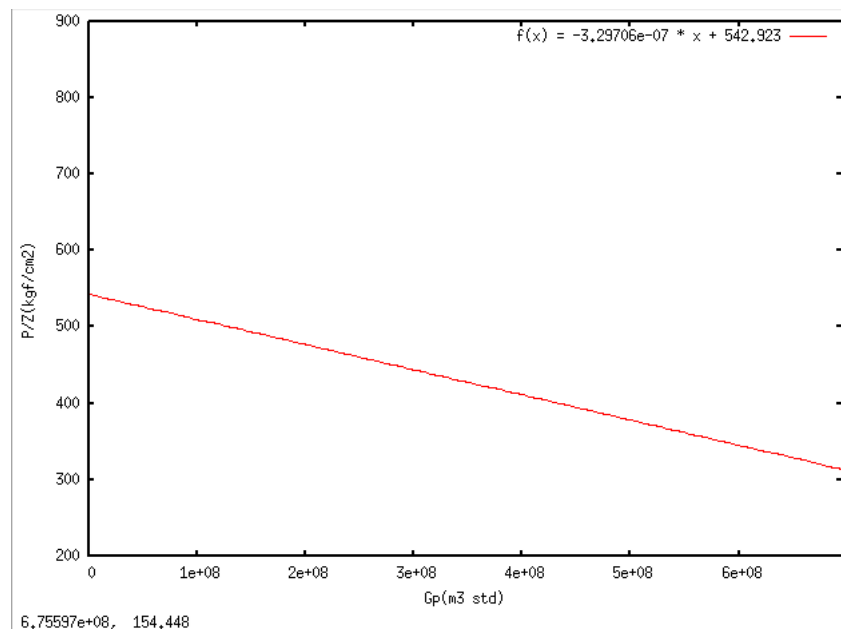


Figura 6.1: Gráfico Reservatório Volumétrico de Gás Seco

6.1.2 Teste 2: anormalmente pressurizado

Neste caso será testado o simulador para um reservatório de gás seco anormalmente pressurizado. Para isto utilizaremos os dados de um exemplo para calcular o volume original de gás.

No primeiro momento, escolhe-se “Gas” como o tipo de fluido do reservatório e “Anormalmente Pressurizado” como tipo de mecanismo. O número de medições feitas foi onze. Os dados de pressão, gás produzido e coeficiente de compressibilidade (Z) seguem na tabela abaixo.

Tabela 6.2: Dados para Teste de Reservatório de Gás Seco Anormalmente Pressurizado

n	Pressão (kgf/cm ²)	Gp (m ³ std)	Z
1	805	0	1.496
2	712	81e6	1.397
3	651	152e6	1.330
4	603	220e6	1.280
5	556	287e6	1.230
6	519	341e6	1.192
7	482	411e6	1.154
8	449	455e6	1.122
9	410	516e6	1.084
10	352	611e6	1.033
11	293	696e6	0.998

Outros Dados:

- Saturação inicial de água - Swi: 0.22
- Compressibilidade da água - Cw: 4.3e-5 (kgf/cm²)-¹
- Compressibilidade da formação - Cf: 27.7e-5 (kgf/cm²)-¹

Apresenta-se na listagem o arquivo com a saída de tela do teste 2.

Listing 6.2: Arquivo de saída de tela do teste 2

Tipo de Fluido:

```
(1) Gas
(2) Oleo
1
```

Tipo de mecanismo:

```
(1) Volumetrico de Gas Seco
(2) Anormalmente Pressurizado
(3) Sob Influxo de Agua
(4) Sob Influxo de agua Modelo Pot Aquifer
```

(5) Sob Influxo de agua Modelo Schilthuis
(6) Gas Condensado Nao Retrogrado
(7) Gas Condensado Retrogrado
2

Numero de medicoes:11

Pressao (1) (kgf/cm2):805

Gas Produzido (1)(m3 std):0

Z(1):1.496

Pressao (2) (kgf/cm2):712

Gas Produzido (2)(m3 std):81e6

Z(2):1.397

Pressao (3) (kgf/cm2):651

Gas Produzido (3)(m3 std):152e6

Z(3):1.330

Pressao (4) (kgf/cm2):603

Gas Produzido (4)(m3 std):220e6

Z(4):1.280

Pressao (5) (kgf/cm2):556

Gas Produzido (5)(m3 std):287e6

Z(5):1.230

Pressao (6) (kgf/cm2):519

Gas Produzido (6)(m3 std):341e6

Z(6):1.192

Pressao (7) (kgf/cm2):482

Gas Produzido (7)(m3 std):411e6

Z(7):1.154

Pressao (8) (kgf/cm2):449

Gas Produzido (8)(m3 std):455e6

Z(8):1.122

Pressao (9) (kgf/cm2):410

Gas Produzido (9)(m3 std):516e6

Z(9):1.084

Pressao (10) (kgf/cm2):352

Gas Produzido (10)(m3 std):611e6

Z(10):1.033

Pressao (11) (kgf/cm2):293

Gas Produzido (11)(m3 std):696e6

Z(11):0.998

Saturacao Inicial de Agua: 0.22

Compressibilidade da agua ((kgf/cm2)⁻¹): 4.3e-5

Compressibilidade da formacao ((kgf/cm2)⁻¹): 27.7e-5

Gp(m3 std)	(P/Z)*(1-Cewf*dP)(kgf/cm2)
0	538.102
8.1e+07	492.256
1.52e+08	461.79
2.2e+08	436.145
2.87e+08	410.696
3.41e+08	389.67
4.11e+08	368.131
4.55e+08	347.858
5.16e+08	323.361
6.11e+08	284.065
6.96e+08	238.382

Regressao Linear

G (m3 std) = -a/b = 1.29833e+09

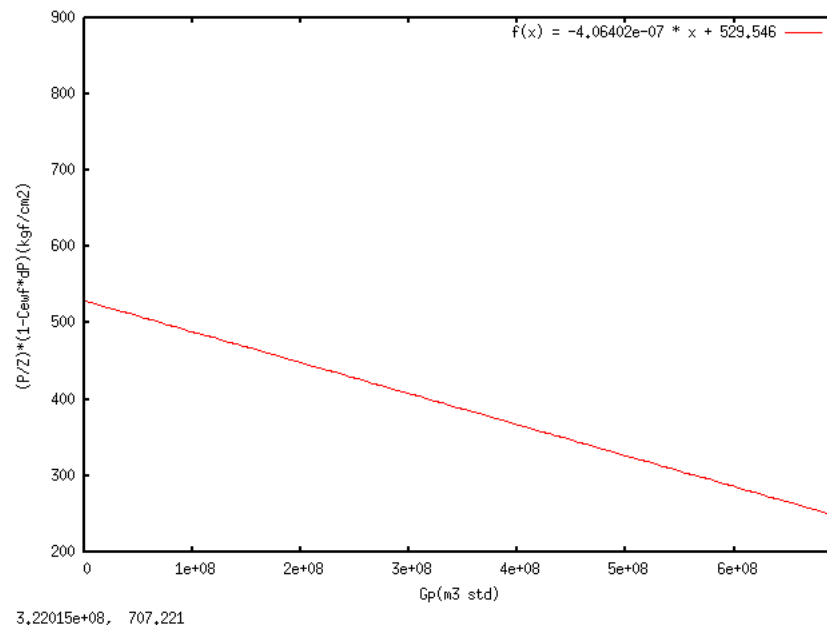


Figura 6.2: Gráfico Gás Anormalmente Pressurizado

6.1.3 Teste 3: sob influxo de água

Neste caso será testado o simulador para um reservatório de gás sob influxo de água, ou seja, alimentado por um aquífero. Para isto utilizaremos os dados de um exemplo para calcular o volume original de gás.

No primeiro momento, escolhe-se “Gás” como o tipo de fluido do reservatório e “Sob Influxo de Água” como tipo de mecanismo. O número de medições feitas foi três. Os dados de pressão, gás produzido, coeficiente de compressibilidade (Z), fator volume-formação do gás (Bg) e influxo de água acumulado (We) seguem na tabela abaixo.

Tabela 6.3: Dados para Teste de Reservatório de Gás Sob Influxo de Água

n	Pressão (kgf/cm ²)	Gp (m ³ std)	Z	n	Bg (m ³ /m ³ std)	We (m ³)
1	205.70	2.237e6	0.86895911	1	0.0057004	120
2	177.57	6.258e6	0.85944533	2	0.0065311	4294
3	149.44	12.799e6	0.85673336	3	0.0077360	27703

Outros Dados:

- Fator volume-formação inicial do gás - Bgi: 0.0052622 m³/m³ std

Apresenta-se na listagem o arquivo com a saída de tela do teste 3.

Listing 6.3: Arquivo de saída de tela do teste 3

```
Tipo de Fluido:
(1) Gas
```

(2) Oleo

1

Tipo de mecanismo:

- (1) Volumetrico de Gas Seco
- (2) Anormalmente Pressurizado
- (3) Sob Influxo de Agua
- (4) Sob Influxo de agua Modelo Pot Aquifer
- (5) Sob Influxo de agua Modelo Schilthuis
- (6) Gas Condensado Nao Retrogrado
- (7) Gas Condensado Retrogrado

3

Numero de medicoes:3

Pressao (1) (kgf/cm2):205.70

Gas Produzido (1)(m3 std):2.237e6

Z(1):0.86895911

Pressao (2) (kgf/cm2):177.57

Gas Produzido (2)(m3 std):6.258e6

Z(2):0.85944533

Pressao (3) (kgf/cm2):149.44

Gas Produzido (3)(m3 std):12.799e6

Z(3):0.85673336

Bgi (m3/m3std): 0.0052622

Bg (1)(m3/m3 std): 0.0057004

Influxo Acumulado We (1)(m3): 120

Bg (2)(m3/m3 std): 0.0065311

Influxo Acumulado We (2)(m3): 4294

Bg (3)(m3/m3 std): 0.0077360

Influxo Acumulado We (3)(m3): 27703

We/(Bg-Bgi)

GpBg/(Bg-Bgi)

273848

2.91004e+07

3.38403e+06

3.22103e+07

1.11986e+07

4.00247e+07

Regressao Linear

G (m3 std) = a = 2.88265e+07

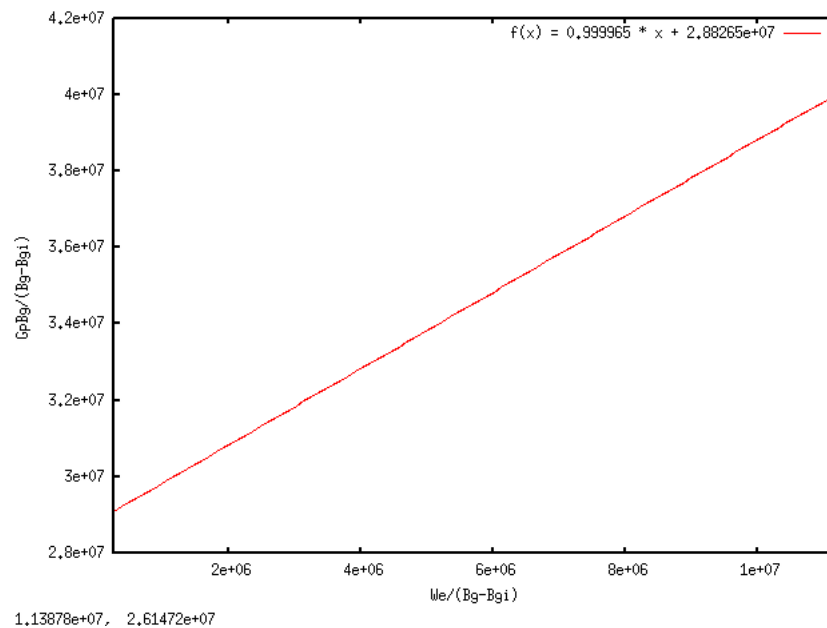


Figura 6.3: Gráfico Influxo de água

6.1.4 Teste 4: sob influxo de água modelo *Pot Aquifer*

Neste caso será testado o simulador para um reservatório de gás sob influxo de água, mas o cálculo do influxo será feito através do modelo Pot Aquifer, e não um valor dado como anteriormente. Para isto utilizaremos os dados de um exemplo para calcular o volume original de gás.

No primeiro momento, escolhe-se “Gas” como o tipo de fluido do reservatório e “Sob Influxo de agua Modelo Pot Aquifer” como tipo de mecanismo. O número de medições feitas foi três. Os dados de pressão, gás produzido, coeficiente de compressibilidade (Z), fator volume-formação do gás (Bg) seguem na tabela abaixo.

Tabela 6.4: Dados para Teste de Reservatorio Sob Influxo de Água Modelo Pot Aquifer

n	Pressão (kgf/cm ²)	Gp (m ³ std)	Z	n	Bg (m ³ /m ³ std)	We (m ³)
1	205.70	2.237e6	0.8689	1	0.0057004	Calculado
2	177.57	6.258e6	0.8594	2	0.0065311	Calculado
3	149.44	12.799e6	0.8567	3	0.0077360	Calculado

Outros Dados:

- $Re = 6096$ m
- $Ro = 762$ m

- $h = 18.3 \text{ m}$
- $f_i = 0.2$
- $B_{gi} = 0.0052622 \text{ m}^3/\text{m}^3 \text{ std}$
- $C_w = 42.7\text{e-}6 \text{ (kgf/cm}^2\text{)-}1$
- $C_f = 56.9\text{e-}6 \text{ (kgf/cm}^2\text{)-}1$

Apresenta-se na listagem o arquivo com a saída de tela do teste 4.

Listing 6.4: Arquivo de saída de tela do teste 4

```
Tipo de Fluido:
(1) Gas
(2) Oleo
1

Tipo de mecanismo:
(1) Volumetrico de Gas Seco
(2) Anormalmente Pressurizado
(3) Sob Influxo de Agua
(4) Sob Influxo de agua Modelo Pot Aquifer
(5) Sob Influxo de agua Modelo Schilthuis
(6) Gas Condensado Nao Retrogrado
(7) Gas Condensado Retrogrado
4

Numero de medicoes:3

Pressao (1) (kgf/cm2):205.70

Gas Produzido (1)(m3 std):2.237e6

Z(1):0.8689

Pressao (2) (kgf/cm2):177.57

Gas Produzido (2)(m3 std):6.258e6

Z(2):0.8594

Pressao (3) (kgf/cm2):149.44

Gas Produzido (3)(m3 std):12.799e6

Z(3):0.8567
Porosidade : 0.2
bgi (m3/m3 std) : 0.0052622
```

Espessura do aquifero - h (m) : 18.3
 Raio do aquifero - r_e (m): 6096
 Raio do reservatorio - r_o (m): 762
 Compressibilidade da agua - C_w ((kgf/cm²)⁻¹): 42.7e-6
 Compressibilidade da formacao - C_f ((kgf/cm²)⁻¹): 56.9e-6

b_g (1) (m³/m³ std) : 0.0057004

b_g (2) (m³/m³ std) : 0.0065311

b_g (3) (m³/m³ std) : 0.0077360

$W_e/(B_g - B_{gi})$	$G_p B_g/(B_g - B_{gi})$
0	2.91004e+07
9.28716e+08	3.22103e+07
9.52743e+08	4.00247e+07

Regressao Linear

G (m³ std) = a = 2.90029e+07

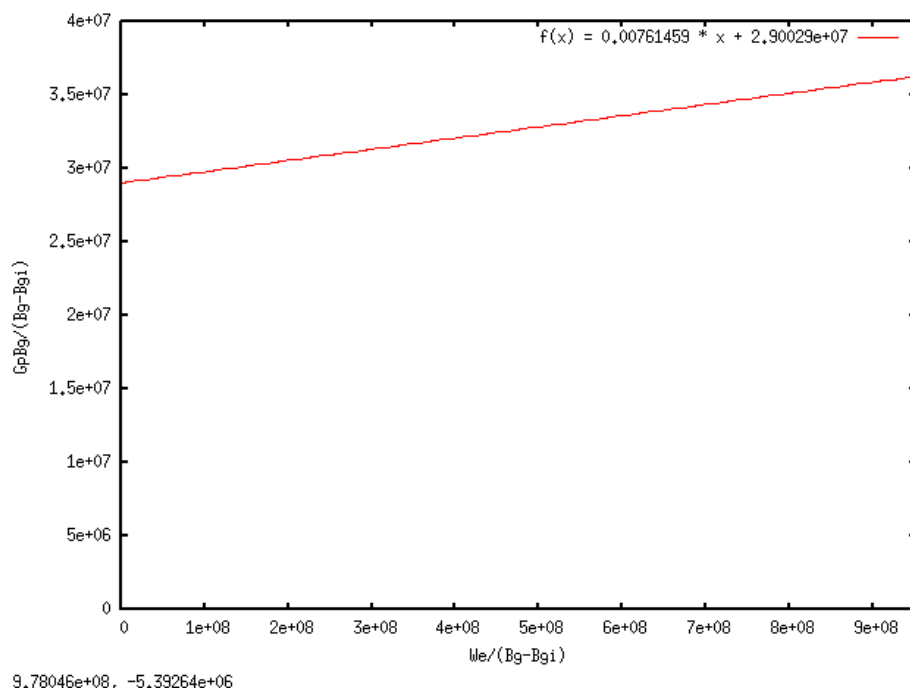


Figura 6.4: Gráfico Modelo Pot Aquifer

6.1.5 Teste 5: sob influxo de água modelo *Schilthuis*

Neste caso será testado o simulador para um reservatório de gás sob influxo de água, mas o cálculo do influxo será feito através do modelo Schilthuis, e não um valor dado como anteriormente. Para isto utilizaremos os dados de

um um exemplo para calcular o volume original de gás.

No primeiro momento, escolhe-se “Gas” como o tipo de fluido do reservatório e “Sob Influxo de agua Modelo Schilthuis” como tipo de mecanismo. O número de medições feitas foi três. Os dados de pressão, gás produzido, coeficiente de compressibilidade (Z), fator volume-formação do gás (Bg) seguem na tabela abaixo.

Tabela 6.5: Dados para Teste de Reservatorio Sob Influxo de Água Modelo Schilthuis

n	Pressão (kgf/cm ²)	Gp (m ³ std)	Z	n	Bg (m ³ /m ³ std)	We (m ³)
1	205.70	2.237e6	0.8689	1	0.0057004	Calculado
2	177.57	6.258e6	0.8594	2	0.0065311	Calculado
3	149.44	12.799e6	0.8567	3	0.0077360	Calculado

Outros Dados:

- Fator volume-formação inicial do gás - Bgi: 0.0052622 m³/m³ std
- j = 70
- t1 = 3 dias; t2 = 5 dias

Apresenta-se na listagem o arquivo com a saída de tela do teste 5.

Listing 6.5: Arquivo de saída de tela do teste 5

```
Tipo de Fluido:
(1) Gas
(2) Oleo
1

Tipo de mecanismo:
(1) Volumetrico de Gas Seco
(2) Anormalmente Pressurizado
(3) Sob Influxo de Agua
(4) Sob Influxo de agua Modelo Pot Aquifer
(5) Sob Influxo de agua Modelo Schilthuis
(6) Gas Condensado Nao Retrogrado
(7) Gas Condensado Retrogrado
5

Numero de medicoes:3

Pressao (1) (kgf/cm2):205.70

Gas Produzido (1)(m3 std):2.237e6
```

Z(1):0.8689

Pressao (2) (kgf/cm2):177.57

Gas Produzido (2)(m3 std):6.258e6

Z(2):0.8594

Pressao (3) (kgf/cm2):149.44

Gas Produzido (3)(m3 std):12.799e6

Z(3):0.8567

t1 (d): 3

t2 (d): 5

Constante de influxo - J (m3/(d*kgf/cm2) : 70

bgi (m3/m3 std) : 0.0052622

bg (1) (m3/m3 std) : 0.0057004

bg (2) (m3/m3 std) : 0.0065311

bg (3) (m3/m3 std) : 0.0077360

We/(Bg-Bgi)

GpBg/(Bg-Bgi)

0

2.91004e+07

3.10363e+06

3.22103e+07

3.18393e+06

4.00247e+07

Regressao Linear

G (m3 std) = a = 2.90029e+07

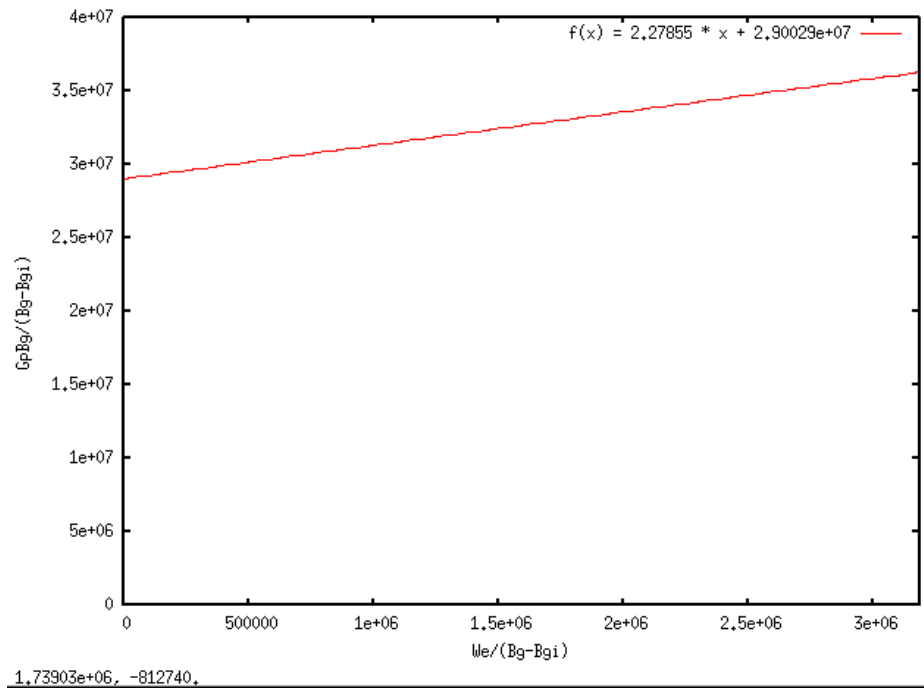


Figura 6.5: Gráfico Modelo Schilthuis

6.1.6 Teste 6: gás condensado não retrógrado

Neste caso será testado o simulador para um reservatório de gás condensado não retrógrado. Para isto utilizaremos os dados de um exemplo para calcular o volume original de gás.

No primeiro momento, escolhe-se “Gas” como o tipo de fluido do reservatório e “Gas Condensado Nao Retrogrado” como tipo de mecanismo. O número de medições feitas foi 3. Os dados de pressão, gás produzido, coeficiente de compressibilidade (Z), fator volume-formação do gás (Bg), volume de condensado (Vc) e volume de água (Vw) seguem na tabela abaixo.

Tabela 6.6: Dados para Teste de Reservatório de Gás Condensado Não Retrógrado

n	P (kgf/cm ²)	Gp (m ³ std)	Z	n	Bg (m ³ /m ³ std)	Vc (m ³)	Vw (m ³)
1	205.70	2.237e6	0.8689	1	0.0057004	15.9	1.6
2	177.57	6.258e6	0.8594	2	0.0065311	16.7	1.9
3	149.44	12.799e6	0.8567	3	0.0077360	17.3	2.2

Outros Dados:

- $B_{gi} = 0.0052622 \text{ m}^3/\text{m}^3 \text{ std}$
- Densidade do condensado (Dc) - 0.78
- Massa do condensado (Mc) - 138

Apresenta-se na listagem o arquivo com a saída de tela do teste 6.

Listing 6.6: Arquivo de saída de tela do teste 6

```
Tipo de Fluido:
(1) Gas
(2) Oleo
1

Tipo de mecanismo:
(1) Volumetrico de Gas Seco
(2) Anormalmente Pressurizado
(3) Sob Influxo de Agua
(4) Sob Influxo de agua Modelo Pot Aquifer
(5) Sob Influxo de agua Modelo Schilthuis
(6) Gas Condensado Nao Retrogrado
(7) Gas Condensado Retrogrado
6

Numero de medicoes:3

Pressao (1) (kgf/cm2):205.70

Gas Produzido (1)(m3 std):2.237e6

Z(1):0.8689

Pressao (2) (kgf/cm2):177.57

Gas Produzido (2)(m3 std):6.258e6

Z(2):0.8594

Pressao (3) (kgf/cm2):149.44

Gas Produzido (3)(m3 std):12.799e6

Z(3):0.8567

Dc: 0.78

Mc: 138

Bgi (m3/m3std): 0.0052622

Vc (m3) (1): 15.9

Vw (m3) (1): 1.6
```

V_c (m³) (2): 16.7

V_w (m³) (2): 1.9

V_c (m³) (3): 17.3

V_w (m³) (3): 2.2

G _{pt} (m ³ std)	(P/Z) (kgf/cm ²)
2.24123e+06	236.736
6.26274e+06	206.621
1.28042e+07	174.437

Regressao Linear

G (m³ std) = -a/b = 4.25875e+07

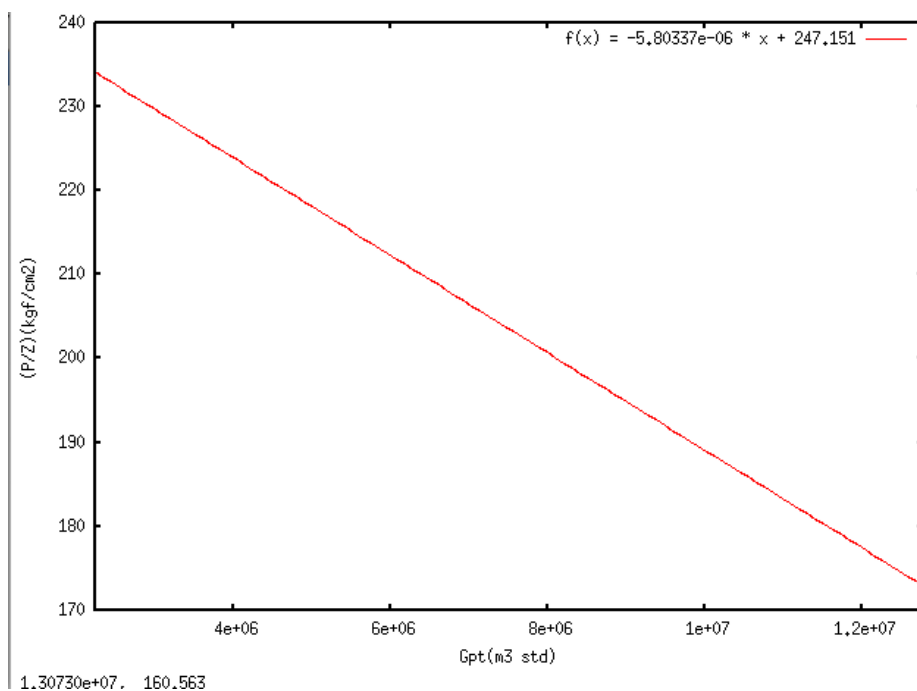


Figura 6.6: Gráfico Gás Condensado Não Retrógrado

6.1.7 Teste 7: gás condensado retrógrado

Neste caso será testado o simulador para um reservatório de gás condensado retrógrado. Para isto utilizaremos os dados de um exemplo para calcular o volume original de gás.

No primeiro momento, escolhe-se “Gas” como o tipo de fluido do reservatório e “Gas Condensado Retrogrado” como tipo de mecanismo. O número de medições feitas foi 3. Os dados de pressão, gás produzido, coeficiente de com-

pressibilidade (Z), fator volume-formação do gás (Bg) e saturação de líquido (Sl) seguem na tabela abaixo.

Tabela 6.7: Dados para Teste de Reservatório de Gás Condensado Retrógrado

n	Pressão (kgf/cm ²)	Gp (m ³ std)	Z	n	Bg (m ³ /m ³ std)	Sl
1	205.70	2.237e6	0.8689	1	0.0057004	0.0260
2	177.57	6.258e6	0.8594	2	0.0065311	0.0030
3	149.44	12.799e6	0.8567	3	0.0077360	0.0308

Outros Dados:

- $B_{gi} = 0.0052622 \text{ m}^3/\text{m}^3 \text{ std}$
- Temperatura = 366 K
- Densidade do líquido = 0.65
- Massa do líquido = 70

Apresenta-se na listagem o arquivo com a saída de tela do teste 7.

Listing 6.7: Arquivo de saída de tela do teste 7

```
Tipo de Fluido:
(1) Gas
(2) Oleo
1

Tipo de mecanismo:
(1) Volumetrico de Gas Seco
(2) Anormalmente Pressurizado
(3) Sob Influxo de Agua
(4) Sob Influxo de agua Modelo Pot Aquifer
(5) Sob Influxo de agua Modelo Schilthuis
(6) Gas Condensado Nao Retrogrado
(7) Gas Condensado Retrogrado
7

Numero de medicoes:3

Pressao (1) (kgf/cm2):205.70

Gas Produzido (1)(m3 std):2.237e6

Z(1):0.8689

Pressao (2) (kgf/cm2):177.57
```

Gas Produzido (2)(m3 std):6.258e6

Z(2):0.8594

Pressao (3) (kgf/cm2):149.44

Gas Produzido (3)(m3 std):12.799e6

Z(3):0.8567

Bgi (m3/m3std): 0.0052622

Temperatura (K): 366

D1: 0.65

M1: 70

S1 (1): 0.0260

S1 (2): 0.0300

S1 (3): 0.0308

Gp(m3 std)	(P/Z2f)(kgf/cm2)
2.237e+06	238.072
6.258e+06	209.066
1.2799e+07	177.939

Regressao Linear

G (m3 std) = -a/b = 4.42846e+07

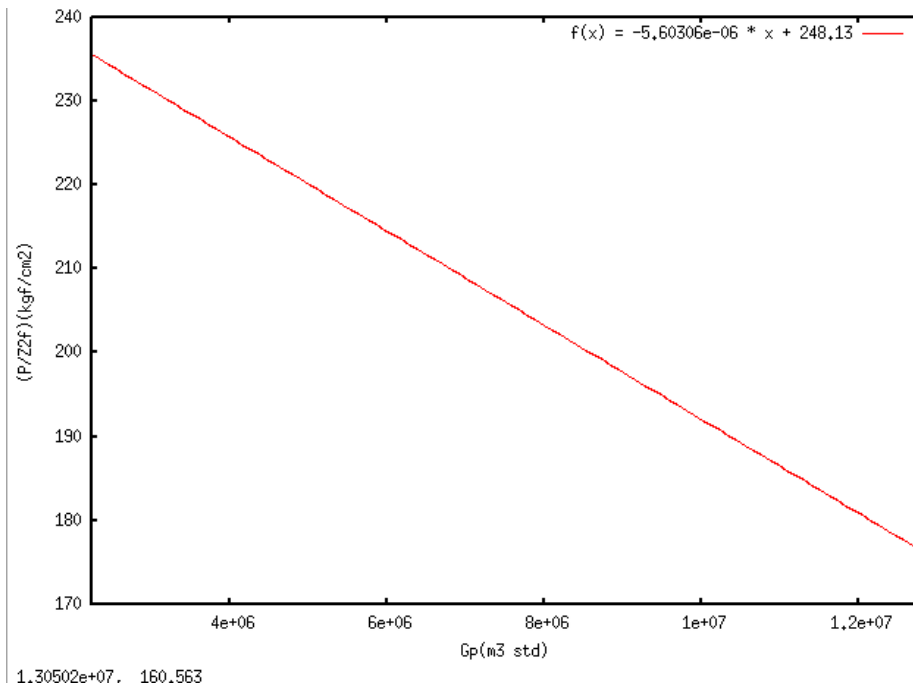


Figura 6.7: Gráfico Gás Condensado Retrógrado

6.2 Reservatório de óleo

6.2.1 Teste 8: gás em solução

Neste caso será testado o simulador para um reservatório de óleo com gás em solução. Para isto utilizaremos os dados de um exemplo para calcular o volume original de óleo.

No primeiro momento, escolhe-se “Óleo” como o tipo de fluido do reservatório e “Gás em Solução” como tipo de mecanismo. O número de medições feitas são três. Os dados de pressão, óleo produzido (N_p) e fator volume-formação do óleo (B_o) seguem na tabela abaixo.

Tabela 6.8: Dados para Teste de Reservatório de Óleo com Gás em Solução

n	Pressão (kgf/cm ²)	N_p (m ³ std)	B_o (m ³ /m ³ std)
1	220	1.2e5	1.322
2	180	2.0e5	1.333
3	140	2.5e5	1.345

Outros Dados:

- Pressão inicial - P_i : 260 kgf/cm²
- Fator volume-formação do óleo inicial - B_{oi} : 1.310 m³/m³ std
- Saturação inicial de água - S_{wi} : 0.20

- Compressibilidade da água - C_w : $4.27e-5 \text{ (kgf/cm}^2\text{)}^{-1}$
- Compressibilidade da formação - C_f : $12.23e-5 \text{ (kgf/cm}^2\text{)}^{-1}$

Apresenta-se na listagem o arquivo com a saída de tela do teste 8.

Listing 6.8: Arquivo de saída de tela do teste 8

```
Tipo de Fluido:
(1) Gas
(2) Oleo
2

Tipo de mecanismo:
(1) Gas em solucao
(2) Capa de Gas
(3) Sob Influxo de Agua
1

Numero de medicoes:3
Pressao Inicial (kgf/cm2): 260
Boi (m3/m3 std): 1.310
Saturacao Inicial de Agua: 0.20
Compressibilidade da agua ((kgf/cm2)^-1): 4.27e-5
Compressibilidade da formacao ((kgf/cm2)^-1): 12.23e-5
Pressao (1)(kgf/cm2): 220
Oleo Produzido (1)(m3 std): 1.2e5
Bo (1)(m3/m3 std): 1.322
Pressao (2)(kgf/cm2): 180
Oleo Produzido (2)(m3 std): 2.0e5
Bo (2)(m3/m3 std): 1.333
Pressao (3)(kgf/cm2): 140
Oleo Produzido (3)(m3 std): 2.5e5
Bo (3)(m3/m3 std): 1.345

Eo+Ef,w (m3/m3 std)                                F(m3)
0.02057                                                158640
0.04014                                                266600
0.0607101                                              336250

Regressao Linear
N (m3 std) = b = 4.41592e+06
```

6.2.2 Teste 9: capa de gás

Neste caso será testado o simulador para um reservatório de óleo com capa de gás. Para isto utilizaremos os dados de um exemplo para calcular o volume original de óleo.

No primeiro momento, escolhe-se “Óleo” como o tipo de fluido do reservatório e “Capa de Gás” como tipo de mecanismo. O número de medições feitas são quatro. Os dados de pressão, óleo produzido (N_p), fator volume-formação do óleo (Bo), razão gás-óleo acumulada (R_p), razão de solubilidade gás/óleo (R_s) e fator volume-formação do gás (B_g) seguem na tabela abaixo.

Tabela 6.9: Dados para Teste de Reservatório de Óleo com Capa de Gás

n	Pressão (kgf/cm ²)	N_p (m ³ std)	Bo (m ³ /m ³ std)
1	221.47	0.5239e6	1.2353
2	210.92	0.9385e6	1.2222
3	200.37	1.4074e6	1.2122
4	189.83	1.8288e6	1.2022

n	R_p (m ³ std/m ³ std)	R_s (m ³ std/m ³ std)	B_g (m ³ /m ³ std)
1	187.01	84.96	0.00517
2	188.79	80.15	0.00539
3	206.60	75.70	0.00567
4	219.96	71.42	0.00601

Outros Dados:

- Pressão inicial - P_i : 234.18 kgf/cm²
- Fator volume-formação do óleo inicial - Bo_i : 1.2511 m³/m³ std
- Fator volume-formação do gás inicial - B_{gi} : 0.00488 m³/m³ std
- Razão de solubilidade gás/óleo inicial - R_{si} : 90.83 m³/m³ std
- Saturação inicial de água - Sw_i : 0.20

Apresenta-se na listagem o arquivo com a saída de tela do teste 9.

Listing 6.9: Arquivo de saída de tela do teste 9

```

Tipo de Fluido:
(1) Gas
(2) Oleo
2

Tipo de mecanismo:
(1) Gas em solucao
(2) Capa de Gas
(3) Sob Influxo de Agua
2

Numero de medicoes:4
Pressao Inicial (kgf/cm2): 234.18
Boi (m3/m3 std): 1.2511

```

Saturacao Inicial de Agua: 0.20
 Compressibilidade da agua ((kgf/cm2)⁻¹): 4.27e-5
 Compressibilidade da formacao ((kgf/cm2)⁻¹): 12.23e-5
 Pressao (1)(kgf/cm2): 221.47
 Oleo Produzido (1)(m3 std): 0.5239e6
 Bo (1)(m3/m3 std): 1.2353
 Pressao (2)(kgf/cm2): 210.92
 Oleo Produzido (2)(m3 std): 0.9385e6
 Bo (2)(m3/m3 std): 1.2222
 Pressao (3)(kgf/cm2): 200.37
 Oleo Produzido (3)(m3 std): 1.407e6
 Bo (3)(m3/m3 std): 1.2122
 Pressao (4)(kgf/cm2): 189.83
 Oleo Produzido (4)(m3 std): 1.8288e6
 Bo (4)(m3/m3 std): 1.2022
 Bgi (m3/m3 std): 0.00488
 Rsi (m3/m3 std): 90.83
 .

Rp (1)(m3 std/m3 std): 187.01
 Rs (1)(m3/m3 std): 84.96
 Bg (1)(m3/m3 std): 0.00517
 Rp (2)(m3 std/m3 std): 188.79
 Rs (2)(m3/m3 std): 80.15
 Bg (2)(m3/m3 std): 0.00539
 Rp (3)(m3 std/m3 std): 206.60
 Rs (3)(m3/m3 std): 75.70
 Bg (3)(m3/m3 std): 0.00567
 Rp (4)(m3 std/m3 std): 219.96
 Rs (4)(m3/m3 std): 71.42
 Bg (4)(m3/m3 std): 0.00601

Eg/Eo(adm)	F/Eo(m3 std)
5.11058	6.34856e+07
4.56129	5.91865e+07
4.31962	5.86482e+07
4.27578	5.65456e+07

Regressao Linear
 N (m3 std) = a = 2.59602e+07

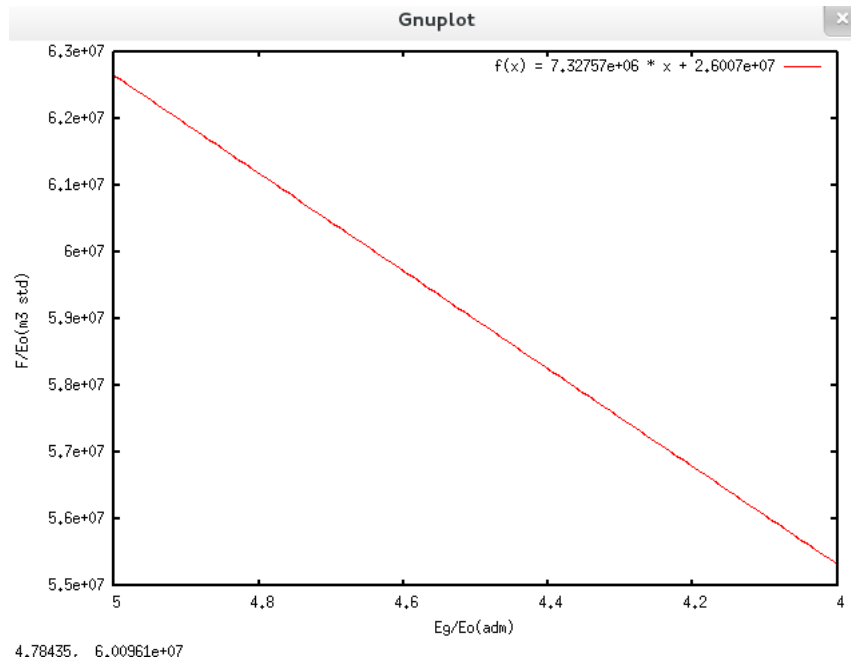


Figura 6.8: Gráfico Capa de Gás

6.2.3 Teste 10: sob influxo de água

Neste caso será testado o simulador para um reservatório de óleo sob influxo de água. Para isto utilizaremos os dados de um exemplo para calcular o volume original de óleo.

No primeiro momento, escolhe-se “Óleo” como o tipo de fluido do reservatório e “Sob Influxo de Água” como tipo de mecanismo. O número de medições feitas são três. Os dados de pressão, óleo produzido (N_p), fator volume-formação do óleo (B_o), razão gás-óleo acumulada (R_p), razão de solubilidade gás/óleo (R_s), fator volume-formação do gás (B_g) e influxo de água acumulado (W_e) seguem na tabela abaixo.

Tabela 6.10: Dados para Teste de Reservatório de Óleo Sob Influxo de Água

n	Pressão (kgf/cm ²)	N_p (m ³ std)	B_o (m ³ /m ³ std)
1	184	1.0e6	1.23
2	174	1.6e6	1.22
3	170	2.0e6	1.21

n	R_p (m ³ std/m ³ std)	R_s (m ³ std/m ³ std)	B_g (m ³ /m ³ std)	W_e (m ³)
1	187.01	115	0.0037	273402
2	188.79	111	0.0045	956023
3	206.60	106	0.0049	1841477

Outros Dados:

- Pressão inicial - P_i : 204 kgf/cm²
- Fator volume-formação do óleo inicial - B_{oi} : 1.25 m³/m³ std

- Fator volume-formação do gás inicial - Bgi: $0.0021 \text{ m}^3/\text{m}^3 \text{ std}$
- Razão de solubilidade gás/óleo inicial - Rsi: $128 \text{ m}^3/\text{m}^3 \text{ std}$
- Saturação inicial de água - Swi: 0.20
- Compressibilidade da água - Cw: $4.27\text{e-}5 \text{ (kgf/cm}^2\text{)}^{-1}$
- Compressibilidade da formação - Cf: $12.23\text{e-}5 \text{ (kgf/cm}^2\text{)}^{-1}$
- Razão entre o volume original de gás na capa e o volume original de óleo em condições de reservatório - m: 0.05

Apresenta-se na listagem o arquivo com a saída de tela do teste 10.

Listing 6.10: Arquivo de saída de tela do teste 10

```
Tipo de Fluido:
(1) Gas
(2) Oleo
2

Tipo de mecanismo:
(1) Gas em solucao
(2) Capa de Gas
(3) Sob Influxo de Agua
3

Numero de medicoes:3
Pressao Inicial (kgf/cm2): 204
Boi (m3/m3 std): 1.25
Saturacao Inicial de Agua: 0.20
Compressibilidade da agua ((kgf/cm2)^-1): 4.27e-5
Compressibilidade da formacao ((kgf/cm2)^-1): 12.23e-5
Pressao (1)(kgf/cm2): 184
Oleo Produzido (1)(m3 std): 1.0e6
Bo (1)(m3/m3 std): 1.23
Pressao (2)(kgf/cm2): 174
Oleo Produzido (2)(m3 std): 1.6e6
Bo (2)(m3/m3 std): 1.22
Pressao (3)(kgf/cm2): 170
Oleo Produzido (3)(m3 std): 2.0e6
Bo (3)(m3/m3 std): 1.21
Bgi (m3/m3 std): 0.0021
Rsi (m3 stf/m3 std): 128
m (m3/m3): 0.05
.

Rp (1)(m3 std/m3 std): 187+01
Rs (1)(m3/m3 std): Bg (1)(m3/m3 std): 115
```


Influxo Acumulado We (1)(m3): 273402
Rp (2)(m3 std/m3 std): 188.79
Rs (2)(m3/m3 std): 111
Bg (2)(m3/m3 std): 0.0045
Influxo Acumulado We (2)(m3): 956023
Rp (3)(m3 std/m3 std): 206.60
Rs (3)(m3/m3 std): 106
Bg (3)(m3/m3 std): 0.0049
Influxo Acumulado We (3)(m3): 1841477

We/Eo (m3 std)	F/Eo(m3 std)
15.1658	1.18659e+06
8.1068e+06	2.13018e+07
1.21845e+07	2.25356e+07

Regressao Linear
N (m3 std) = a = 2.45894e+06

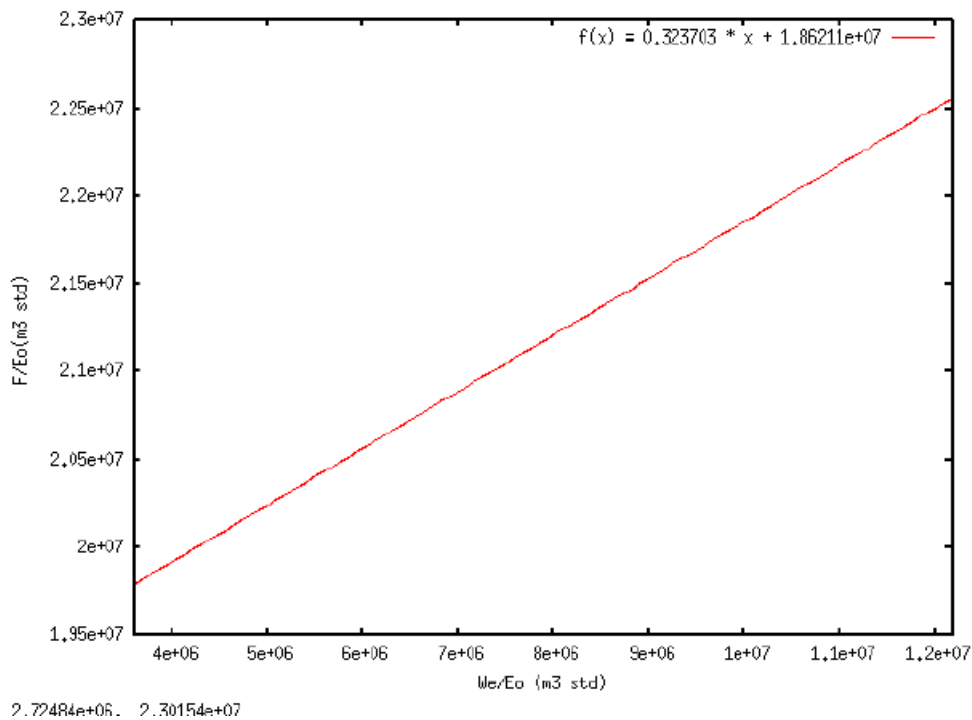


Figura 6.9: Gráfico óleo sob Influxo de Água

Capítulo 7

Documentação

**SIMULADOR DE ESTIMATIVA DE
RESERVA DE RESERVATÓRIO
UTILIZANDO EBM, COM ENFOQUE NOS
RESERVATÓRIOS DE GÁS**

6 de abril de 2015

7.1 Objetivo

O programa tem o objetivo de simular as condições de reservatório para estimar a reserva de gás.

7.2 Fundamentos

O programa faz a previsão do volume que vai ser encontrado dentro do reservatório estudado utilizando dados reais obtidos em testes. Com essas informações, utiliza-se a Equação de Balanço de Materiais para fazer a previsão, sendo adaptada de acordo com cada mecanismo.

7.3 Procedimento

1. Escolher o fluido presente no reservatório:
 - (a) Para reservatório de gás, escolha 1;
 - (b) Para reservatório de óleo, escolha 2;
2. Escolher o mecanismo de produção do reservatório:
 - (a) Em reservatório de gás:
 - i. Volumétrico de gás seco, escolha 1;
 - ii. Gás seco anormalmente pressurizado, escolha 2;
 - iii. Gás seco sob influxo de água, escolha 3;
 - iv. Gás seco sob influxo modelo Pot Aquifer, escolha 4;
 - v. Gás seco sob influxo modelo Schilthuis, escolha 5;
 - vi. Gás condensado não retrógrado, escolha 6;
 - vii. Gás condensado retrógrado, escolha 7;
 - (b) Em reservatório de óleo:
 - i. Com gás em solução, escolha 1;
 - ii. Com capa de gás, escolha 2;
 - iii. Sob influxo de água, escolha 3;
3. Dizer o número de medições de dados feitas nos testes.
4. Informar os dados do reservatório pedidos.
5. Obter o volume original de gás ou óleo.
6. Gerar gráfico pelo Gnuplot.

Índice Remissivo

Análise de domínio, 12

Análise orientada a objeto, 14

AOO, 14

Casos de uso, 3

colaboração, 21

comunicação, 21

Diagrama de colaboração, 21

Diagrama de sequência, 20

Elaboração, 5

Eventos, 20

Implementação, 23

Mensagens, 20