

DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Derivative-Free Optimization for Automatic Tuning of an Oil Well Simulator

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:
DAS 5511: Projeto de Fim de Curso*

Willian de Medeiros Galvani

Florianópolis, Março de 2017

Derivative-Free Optimization for Automatic Tuning of an Oil Well Simulator

Willian de Medeiros Galvani

Esta monografia foi julgada no contexto da disciplina

DAS 5511: Projeto de Fim de Curso

e aprovada na sua forma final pelo

Curso de Engenharia de Controle e Automação

Prof. Eduardo Camponogara

For even the very wise cannot
see all ends.

J. R. R. Tolkien

Resumo

Para maximizar a produção, e assim os lucros, um campo de petróleo precisa operar em um ponto ótimo.

Encontrar esse ponto ótimo não é uma tarefa fácil. Para testar novos modelos, sistemas de controle e de predição, são utilizados simuladores.

Mas tais simuladores são complexos por si só, e precisam ser sintonizados regularmente para refletir as mudanças nas reservas de petróleo com o tempo. Esta sintonia pode tomar tempo e ser trabalhosa para o engenheiro responsável, além de não ser ótima.

A solução proposta é utilizar um método de Otimização Sem-Derivadas para sintonizar o um simulador de poços de petróleo, minimizando as distancias entre as variaveis reais medidas e os resultados simulados.

Palavras-chave: otimização sem derivada, poços de petróleo, simulação, sintonia automática

Abstract

Para obter o maior retorno financeiro possível, uma planta para extração de petróleo deve funcionar em um ponto de performance ideal. Como campos de petróleo são sistemas complexos e variam com o tempo, simuladores são utilizados para aproximar o sistema o real. Para reproduzir as medições observadas apropriadamente, os simuladores são sintonizados periodicamente pelo engenheiro de processo. Esta sintonia frequentemente é cansativa demorada, e pode ser não-ótima.

A abordagem aqui proposta é utilizar um método de otimização sem derivada para sintonizar um simulador miniimizando a discrepância entre os dados colhidos no mundo real e os simulados.

(RESULTS HERE)

(CONCLUSION HERE)

Results: What's the answer? Specifically, most good computer architecture papers conclude that something is so many percent faster, cheaper, smaller, or otherwise better than something else. Put the result there, in numbers. Avoid vague, hand-waving results such as "very", "small", or "significant." If you must be vague, you are only given license to do so when you can talk about orders-of-magnitude improvement. There is a tension here in that you should not provide numbers that can be easily misinterpreted, but on the other hand you don't have room for all the caveats.

Conclusions: What are the implications of your answer? Is it going to change the world (unlikely), be a significant "win", be a nice hack, or simply serve as a road sign indicating that this path is a waste of time (all of the previous results are useful). Are your results general, potentially generalizable, or specific to a particular case?

Keywords: derivative-free optimization, oil well, simulation, automatic tuning

Lista de ilustrações

Figura 1 – Demonstração do Nelder-Mead	14
Figura 2 – Experimento 1.	18
Figura 3 – Experimento 1.	19

Lista de tabelas

Sumário

1	INTRODUCTION	9
1.1	Motivation	9
1.2	Goals	9
1.3	Report Structure	10
2	OIL WELL MODELING AND SIMULATION	11
2.1	Offshore Oil Production	11
2.1.0.1	Wellbore	11
2.1.0.2	Wellhead	11
2.1.0.3	Choke	11
2.1.0.4	Manifold	12
2.1.0.5	Riser	12
2.2	Satellite Oil Wells	12
2.3	Oil Well	12
3	DERIVATIVE-FREE OPTIMIZATION	13
3.1	Visão Geral	13
3.2	Método do Simplex de Nelder-Mead	13
3.3	Baricenter Method	13
3.4	OrthoMADS	13
4	APPLICATION AND RESULTS	18
4.1	Marlim (or Pipesim) Simulator	18
4.2	Experimentos	18
4.2.1	Sintonia de curva	18
4.3	Discussion	19
5	CONCLUSION	20
	 APÊNDICES	 21
	APÊNDICE A – REVISÃO DE PROBABILIDADE	22

ANEXOS	23
ANEXO A – TITLE OF APPENDIX A	24

1 Introduction

1.1 Motivation

Uma das dificuldades da produção de petróleo é controlar o processo de modo a produzir exatamente o necessário em cada estação. A produção em falta ou excesso pode afetar as interfaces com linhas de transporte, assim como extressar os sistemas da plataforma.

In order to get the maximum financial returns from an oilfield, the processes must be always at a point of maximum performance. Since the oilfield systems are complex and change with time, simulators are used to approximate the real world system. In order to reproduce the observed measurements properly, these simulators are tuned routinely by the process engineer. This tuning is often a cumbersome and time-consuming task.

A well tuned simulator can be user for process modeling and prediction by control and optimization algorithms.

How long it takes to tune the simulator reflects on how long the production can be optimized, thus potentially affecting the oilfields revenue.

- Oil well simulators are tuned routinely in order to reproduce observed measurements, a process that can be cumbersome and time-consuming.
- The simulator of oil wells should be adequately tuned before it can be used for process modeling and prediction by control and optimization algorithms.
- The degree of accuracy can have major production and economic impact.

1.2 Goals

O objetivo final deste trabalho é o desenvolvimento de uma ferramenta capaz de auxiliar a otimização da produção em plataformas offshore. Nós pretendemos identificar, implementar, e testar métodos de otimização sem derivadas compatíveis para sintonizar um simulador. Inicialmente os métodos a ser estudados são O Simplex de Nelder-Mead, Baricentro, e OrthoMADs (implementação NOMAD). É esperado que o método desenvolvido ajude a aliviar a carga sobre o engenheiro, mudando sua função para monitoração do sistema.

- Study and identify suitable derivative-free optimization methods for an oil well simulator.

- Implement and test derivative-free optimization algorithms for an oilfield simulator.

1.3 Report Structure

Este relatório está dividido em quatro seções principais. Na seção 2, é dada uma visão geral dos campos de petróleo e como eles funcionam, explicando a estrutura de uma FPSO, assim como cada um de seus componentes. A medida que os sistemas são descritos, as variáveis usadas para BATATA e DOCE DE LEITE serão destacadas.

Na seção 3, damos uma apresentação geral dos conceitos de otimização sem derivada, demonstrando cada um dos métodos utilizados.

Na seção 4, nós apresentamos a estrutura da aplicação, como os componentes se conectam, as configurações e condições dos testes, entradas e saídas, e discussão dos resultados.

A seção 5 contém as conclusões dos experimentos.

2 Oil Well Modeling and Simulation

2.1 Offshore Oil Production

A produção em si pode ser estruturada em diversos layouts diferentes. Em águas rasas, até 100m, é comum a utilização de Complexos de Aguas Razas, os quais são compostos de diversas plataformas, geralmente conectadas por pontes, cada uma com uma função principal, como acomodações, riser, e geração de energia. Já entre 100 e 500m, é possível utilizar-se de bases de gravidade, grandes estruturas conicas e ocas de concreto, que tem função tanto estrutural quanto de armazenamento. Torres articuladas são uma estrutura semelhante às bases de gravidade, mas as base é conectada a plataforma por uma torre articulada, de forma a absorver os impactos de correntes marinhas e ventos. FPSOs (Floating, Production, Storage and Offloading) são geralmente construídas a partir de navios-tanque, com uma torre giratória na proa ou central aonde são conectadas as tubulações, permitindo que o navio se alinhe livremente ao vento e correntes marítimas. As FPSOs dominam os novos campos de petróleo, já que os novos campos de exploração estão predominantemente em áreas de maior profundidade.

Nas próximas seções são descritos os principais componentes do sistema de produção em uma FPSO.

2.1.0.1 Wellbore

O buraco em si. Cavado na rocha, o poço propicia acesso direto ao recurso buscado, seja ele petróleo ou gás. A estrutura do poço em si pode variar bastante dependendo da localização, profundidade, composição da rocha e tipo de recurso procurado.

2.1.0.2 Wellhead

A cabeça do poço é o componente mais upstream, fazendo a interface entre o poço em si e os sistemas de extração. Ela é composta por uma válvula mestre, medidor de pressão, cabeça do revestimento, válvula de asa(?) e pelo choke. Ela pode conter também vários outros instrumentos para medição, controle, e teste do fluxo de hidrocarbonetos. As variáveis PWF (Pressure at Well

2.1.0.3 Choke

A função do choke é controlar a abertura dos dutos, controlando assim a vazão do poço.

2.1.0.4 Manifold

Em alguns setups são utilizados manifolds que conectam a varios poços, centralizando o transporte de óleo para a superfície.

2.1.0.5 Riser

Os Risers são linhas verticais de escoamento, tanto para escoamento de óleo bruto quanto de injeção de gás para gás-lift

- Discuss in the general terms the structure of an offshore oilfield, which consists of subsea oil wells with risers connecting to a production platform.

Em um campo em alto mar, varios poços submarinos são conectados a um manifold comum, que liga-se ao riser para o transporte até a

- Discuss in general terms the surface processing facilities, valves, separator, compressor, flare and exportation.

2.2 Satellite Oil Wells

Discuss the structure of offshore oil wells that operate with continuous lift-gas injection (gas-lift).

2.3 Oil Well

- Give a brief presentation about Marlim and/or Pipesim.
- Present some curves of oil wells modeled in Pipesim.

3 Derivative-Free Optimization

3.1 Visão Geral

- [Motivate the use of derivative-free optimization](#). Frequentemente em problemas de engenharia os modelos utilizados são aproximações simplificadas da realidade, e em outros casos completamente desconhecidos.

Métodos de otimização sem derivadas são adequados em casos aonde o modelo matemático é não-explicito, custoso ou as derivadas não estão disponíveis, devido a inexistência do modelo explícito ou presença de ruídos impossibilitando a estimação das derivadas.

Os métodos de otimização sem derivadas procuram encontrar mínimos computando o menor número possível de pontos do problema, de modo a minimizar também o tempo de execução da otimização.

- [Give a general presentation/introduction to derivative-free optimization](#)

3.2 Método do Simplex de Nelder-Mead

Também conhecido como Downhill Simplex Method, ou Amoeba Method, consiste em utilizar um polígono com $n+1$ vértices em n dimensões que se expande, contraí, ou reflete de modo a se mover em direção ao gradiente da função objetivo.

Um exemplo da execução do algoritmo de Nelder-Mead pode ser visto na figura 1. A partir de um ponto inicial, o simplex se locomove como uma ameba até encontrar um ponto de mínimo.

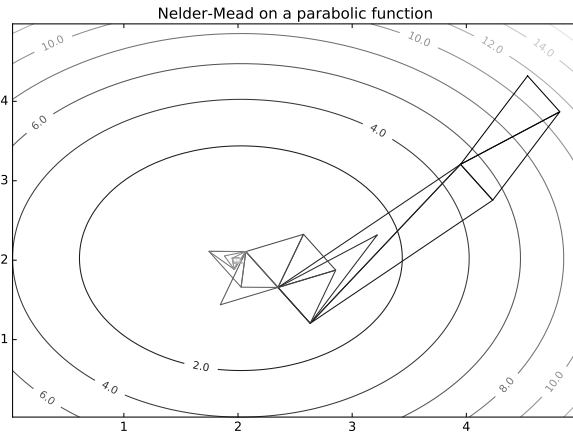
3.3 Baricenter Method

3.4 OrthoMADS

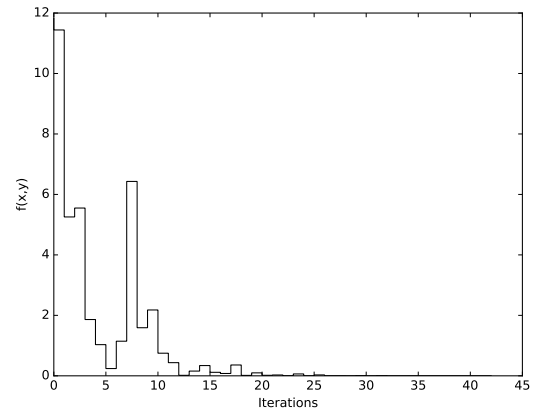
Os métodos MADS (Mesh Adaptive Direct Search) são uma classe de métodos baseados ...

A cada iteração k são executados dois passos de busca, Search e Poll, analisando a feasibility e valor da função. O objetivo de cada nova iteração é encontrar um ponto $f(x) < f(x_k)$, aonde x_k é o melhor ponto encontrado até a iteração atual.

As buscas são feitas sempre em uma grade, definida por:



(a) Passos do Nelder-Mead



(b) Nelder-Mead aplicado em um parabolóide.

Figura 1 – Demonstração do Nelder-Mead

Algorithm 1 Nelder-Mead's Downhill Simplex**Require:** $X = (x_1, \dots, x_{n+1})$ pontos de teste:

```

1: while Não Convergiu do
2:   ordenar  $X$ ; ( $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$ )
3:   Calcular o centroide  $x_0$  de  $(x_1, \dots, x_{n+1})$ 
4:   Calcular  $x_r$  refletido:  $x_r = x_0 + \alpha(x_0 - x_{n+1})$ 
5:   (Reflecção)
6:   if  $f(x_1) < f(x_r) < f(x_{n+1})$  then
7:      $x_{n+1} \leftarrow x_r$ 
8:     Continue
9:   end if
10:  (Expansão)
11:  if  $f(x_r) < f(x_1)$  then
12:    Calcular ponto expandido  $x_e = x_0 + \gamma(x_r - x_0)$ 
13:    if  $f(x_e) < f(x_r)$  then
14:       $x_n \leftarrow x_e$ 
15:    else
16:       $x_n \leftarrow x_r$ 
17:    end if
18:    Continue
19:  end if
20:  (Contração)
21:  Computar o ponto contraído  $x_c = x_0 + \rho(x_{n+1} - x_0)$ 
22:  if  $f(x_c) < f(x_{n+1})$  then
23:     $x_{n+1} \leftarrow x_c$ 
24:    Continue
25:  end if
26:  (Encolhimento)
27:   $x_i \leftarrow x_1 + \sigma(x_i - x_1), i = 2 \dots n+1$ 
28: end while

```

$M_k = \{x + \Delta_k^m Dz : x \in V_k, z \in \mathbb{N}^{n_D}\} \subset \mathbb{R}^n$ Aonde M_k é o conjunto de pontos da grade, x é o ponto mínimo atual, Δ_k^m é o parametro de tamanho da malha, D é uma matrix $\mathbb{R}^{n \times n_D}$ composta por n_D direções que definem um conjunto gerador position no \mathbb{R}^n . Para o OrthoMads e LtMads, D é simplesmente definida como $[I_n - I_n]$ aonde I_n é a matriz identidade de dimensões n .

O passo search pode ser qualquer tipo de heurística que escolha um ponto mais adequado da malha para tentar acelerar a convergência.

O passo poll é a parte mais importante do método, que garante sua convergência. A cada iteração k os pontos a serem utilizados são definidos por:

$$P_k = \{x_k + \Delta_k^p d : d \in D_k\} \subset M_k$$

Aonde x_k é o ponto atual e cada coluna de D_k é formada por combinações inteiras das colunas de D de forma a criar um conjunto gerador positivo. Δ_k^p é o *parametro de tamanho de poll*.

Ambos LtMads e OrthoMads utilizam um parametro ℓ_k chamado de *índice de malha* para atualizar os parametros de tamanho de poll e search de acordo com esta lógica:

$$\Delta_k^p = 2^{-\ell_k} \text{ e } \Delta_k^m = \min\{1, 4^{-\ell_k}\} \quad (3.1)$$

A cada nova iteração, se em uma iteração um novo *incumbente* não é encontrado, ela é dita mal sucedida, e $\ell_{k+1} \leftarrow \ell_k + 1$ (reduzindo Δ_k^m e Δ_k^p), por outro lado, se for encontrado um novo *incumbente*, a iteração é dita bem sucedida, e $\ell_{k+1} \leftarrow \ell_k - 1$ (aumentando Δ_k^m e Δ_k^p). Devido a 3.1, no caso de uma iteração mal-sucedida o parametro de poll diminui mais rápido que o de malha, de modo a permitir o uso de muitos mais pontos, refinando a malha.

A diferença entre o OrthoMads e LtMads se dá na geração da base D_k . O LtMads utiliza uma matriz triangular inferior para a geração da base, fazendo permutações entre os elementos e completando ela em uma base maximal ou minimal, sem garantir ortogonalidade entre as direções, de modo que os ângulos entre as direções podem ser grandes, causando grandes cones de espaço não explorado. Já o OrthoMads utiliza uma base maximal definida por $[H_k - H_k]$, aonde as colunas de H_k formam um base ortogonal de \mathbb{R}^n . Além disso, as direções de D_k são inteiras, de modo que os pontos gerados estão automaticamente contidos na malha definida por $D = [I_n - I_n]$.

Para a geração de D_k , o OrthoMads utiliza a sequencia pseudo-aleatoria de Halton, que cobre mais uniformemente o espaço que uma sequência aleatória real, para gerar vetores u_t .

A saída da sequência de Halton, no entanto, não respeita as restrições impostas pela malha. É necessário arredondar, escalar, e rotacionar o vetor u_t . O índice ℓ é utilizado para transformar a direção u_t na *direção ajustada de Halton* $q_{t,\ell} \in \mathbb{Z}^n$, uma direção cuja norma é próxima a $2^{\frac{|\ell|}{2}}$

HERE, BLACK MAGIC IS USED TO MAKE THE VECTOR ALIGN TO THE MESH.

Para definir $q_{t,\ell}$, primeiramente são definidas duas funções baseadas na t -ésima direção de Halton u_t :

$$q_t(\alpha) = \text{round} \left(\alpha \frac{2u_t - e}{\|2u_t - e\|} \right) \in \mathbb{Z}^n \cap \left[-\alpha - \frac{1}{2}, \alpha + \frac{1}{2} \right]^n$$

Aonde *round* é a operação arredondar para cima ($\text{round}(0, 5) = 1$, $\text{round}(-0, 5) = -1$) e $\alpha \in \mathbb{R}_+$ é um fator de escala.

Desta forma, temos um problema de otimização, precisamos encontrar um $\alpha_{t,\ell}$ tal que $\|q_t(\alpha_{t,\ell})\|$ seja o mais próximo possível de $2^{\frac{|\ell|}{2}}$ sem ultrapassá-lo.

$$\begin{aligned} \alpha_{t,\ell} \in \underset{\alpha \in \mathbb{R}_+}{\operatorname{argmax}} & \|q_t(\alpha)\| \\ \text{s.t.} & \|q_t(\alpha)\| \leq 2^{\frac{|\ell|}{2}} \end{aligned}$$

O problema pode ser resolvido facilmente, já que os degraus da função $\|q_t(\alpha)\|$ acontecem em todos os α no conjunto

$$\left\{ \frac{(2j+1)\|2u_t - e\|}{2|u_t^i - e|} : i = 1, 2, \dots, n, j \in \mathbb{N} \right\}$$

De forma que o problema pode ser solucionado varrendo os pontos do conjunto.

Com um vetor normalizado e na malha, $q \in \mathbb{Z}^n$, é necessário transformá-lo em uma base ortonormal de \mathbb{R}^n . Para isto é utilizada a transformação de Householder:

$$H = \|q\| (I_n - 2vv^T), \text{ onde } v = \frac{q}{\|q\|} \quad (3.2)$$

Aonde H é uma base ortonormal gerada a partir de q .

Com a base ortonormal criada, podemos utilizar o algoritmo 2, comum ao LtMads e OrthoMads.

ALGO MAIS DEVE SER ESCRITO AQUI, MAS OQ?

Algorithm 2 OrthoMads

```

1: [0] Inicialização
2:  $x_0 \in \Omega, \ell_0 \leftarrow 0, k \leftarrow 0, t_0 \leftarrow p_n$ 
3: while Não Convergiu do
4:   (ITERAÇÃO  $k$ )
5:     Search (opcional)
6:     Avalia  $f$  em um conjunto finito  $S_k \subset M_k$ 
7:     POLL
8:     if o tamanho do parametro POLL é o menor até então ( $\Delta_k^p = \min\{\Delta_j^p : j = 0, 1, \dots, k\}$ ) then
9:        $t_k \leftarrow \ell_k + t_0$ 
10:    else (Já foram considerados tamanhos menores)
11:       $t_k \leftarrow 1 + \max\{t_j : j = 0, 1, \dots, k - 1\}$ 
12:    end if
13:    Computa  $u_{tk}, q_{tk}\ell_k$  and  $D_k = [H_{tk} \quad -H_{tk}]$ 
14:    UPDATES
15:    if A iteração foi bem sucedida ( se existe um  $x_s \in S_k$  ou  $x_p \in P_k$  tal que  $f(x_s) < f(x_k)$  ou  $f(x_p) < f(x_k)$  ) then
16:       $x_{k+1} \leftarrow x_s$  or  $x_p$ 
17:       $\ell_{k+1} \leftarrow \ell_k - 1$ 
18:    else(iteração falhou)
19:       $x_{k+1} \leftarrow x_k$ 
20:       $\ell_{k+1} \leftarrow \ell_k + 1$ 
21:    end if
22:     $k \leftarrow k + 1$ 
23: end while

```

4 Application and Results

4.1 Marlim (or Pipesim) Simulator

Sistemas de produção de petróleo são sistemas nos quais se deseja sempre produzir o máximo possível e de forma segura. O PIPESIM, da Schlumberger, é um simulador de fluxo multifásico em regime permanente que pode ser utilizado tanto para o projeto como para planejamento de operações em campos de petróleo. Ele permite que sejam simuladas situações alternativas de forma mais rápida e segura que testes reais.

4.2 Experimentos

Foram realizados XX experimentos, em crescente ordem de complexidade

4.2.1 Sintonia de curva

Primeiramente foi configurado um unico poço de petróleo, que pode ser visto na figura 2a. A seguir foram escolhidos dados arbitrariamente para compor a curva "real" de produção. Neste Experimento a curva sintonizada foi a de fluxo de líquido (Barris padrões por dia) por gás injetado (milhões de pés cúbicos padrões por dia). Escolheu-se a pressão estática do reservatório como sendo 4000 psi absoluto, e um índice de produtividade liquido de 25 STDB/d/psi(Barris padrões por dia por pressão estática). Desta forma a figura 2b demonstra a curva "real" a ser sintonizada.

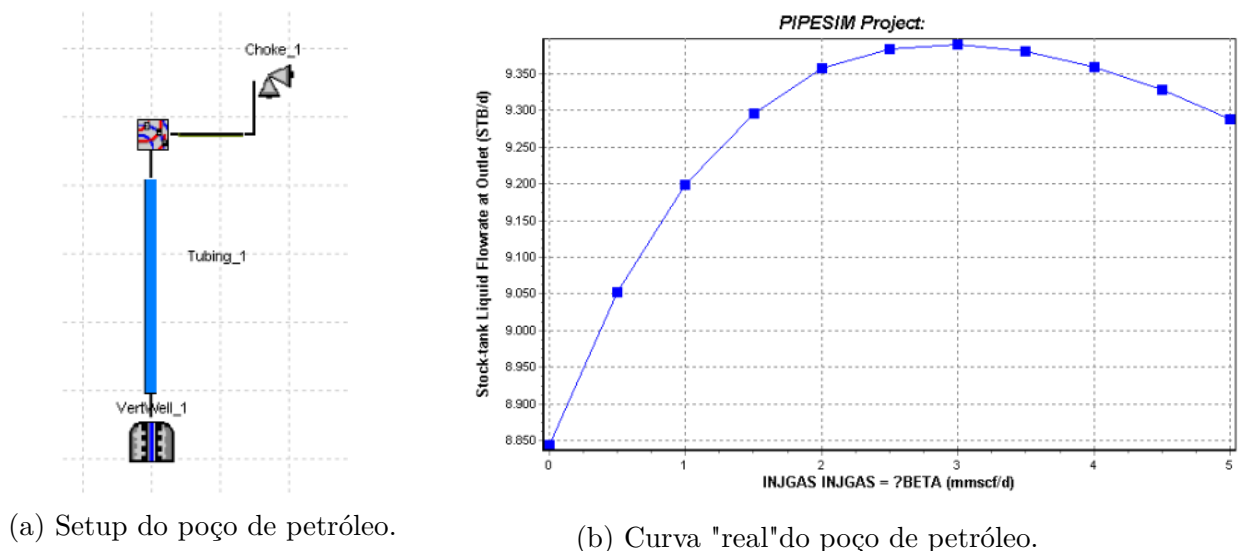
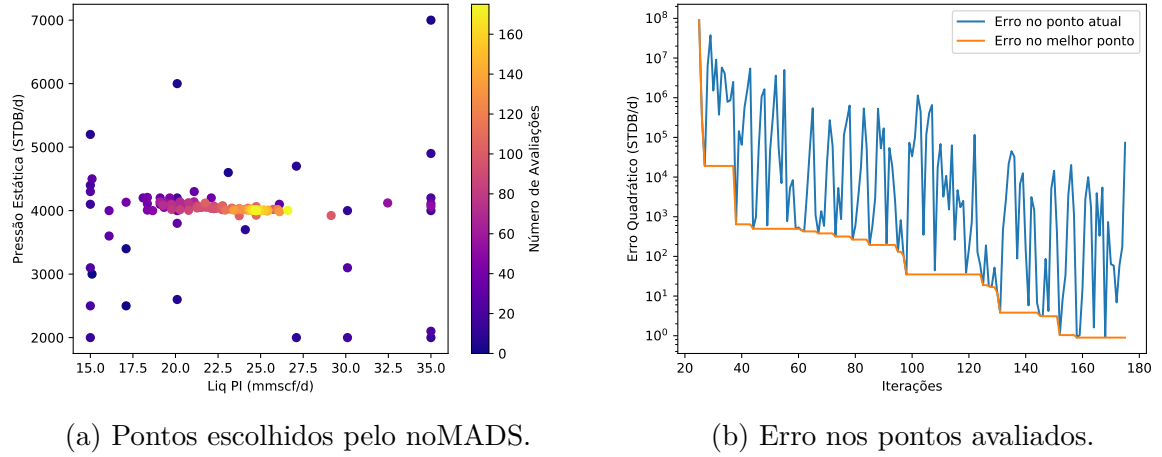


Figura 2 – Experimento 1.



(a) Pontos escolhidos pelo noMADS.

(b) Erro nos pontos avaliados.

Figura 3 – Experimento 1.

Para os testes, os parametros SP (pressão estática) e Liq PI (Índice de produção de líquido) foram iniciados respectivamente em 3000 e 15. Adicionalmente, foram impostos os seguintes limites ao solver do noMADS:

$$MIN_MESH_SIZE = 5e - 4 \quad (4.1)$$

$$H_MIN = 0.001 \quad (4.2)$$

$$F_TARGET = 0.00001 \quad (4.3)$$

em 467 segundos (7:47 minutos) o noMADS convergiu para SP=4004.15, PI = 24.72, com a função custo em 0.896. A parada se deu pelo MIN_MESH_SIZE. A performance do algoritmo pode ser vista na figura 3a

4.3 Discussion

5 Conclusion

Apêndices

APÊNDICE A – Revisão de Probabilidade

Anexos

ANEXO A – Title of Appendix A