

Proposta de Cálculo de Parâmetros de Perfuração de Poços de Petróleo a partir de Coordenadas Espaciais com o Método da Bissecção*

Tiago C. A. Amorim²

^aPetrobras, Av. Henrique Valadares, 28, Rio de Janeiro, 20231-030, RJ, Brasil

Abstract

O Método de Mínima Curvatura é reconhecido como o mais aceito no cálculo de trajetória de poços de petróleo. A formulação para cálculo de coordenadas cartesianas a partir de parâmetros de perfuração é direta, e o cálculo inverso não tem formulação direta.

Neste trabalho é proposto um algoritmo para calcular parâmetros de perfuração a partir de coordenadas cartesianas. O algoritmo proposto tem a forma $g(x) = x$, e encontrar uma solução passa por um problema de encontrar a raiz de uma função.

Foi aplicado o Método da Bissecção para resolver o problema proposto. Os testes realizados mostraram boa coerência entre os valores estimados com o processo iterativo e as respectivas respostas exatas.

Keywords: Método da Mínima Curvatura, Método da Bissecção

1. Introdução

O desenvolvimento de técnicas para construção de poços direcionais na indústria do petróleo iniciou nos anos 1920 [1]. A construção de poços direcionais pode ter diferentes objetivos, desde acessar acumulações que seriam difíceis de serem alcançadas com poços verticais (áreas montanhosas, acumulações abaixo de leitos de rios etc.), para aumento da produtividade (maior exposição da formação portadora de hidrocarbonetos) ou até para interceptar outros poços (poços de alívio em situações de *blowout*¹).

O Método da Mínima Curvatura é largamente aceito como o método padrão para o cálculo de trajetória de poços [2]. Neste método a geometria do poço é descrita como uma série de arcos circulares e linhas retas. A transformação de parâmetros de perfuração (ΔS , θ , ϕ) em coordenadas cartesianas (ΔN , ΔE , ΔV) tem formulação explícita. A operação inversa, de coordenadas cartesianas em parâmetros de perfuração não tem formulação explícita.

No planejamento de novos poços de petróleo as coordenadas espaciais são conhecidas, e é necessário calcular os futuros parâmetros de perfuração. Os parâmetros de perfuração são utilizados para diferentes análises, como o de máximo DLS (*dogleg severity*), que é uma medida da

curvatura de um poço entre dois pontos de medida, usualmente expressa em graus por 30 metros².

Este relatório apresenta uma proposta de metodologia para cálculo dos parâmetros de perfuração a partir das coordenadas cartesianas de pontos ao longo da geometria do poço. A formulação foi derivada das fórmulas utilizadas no Método da Mínima Curvatura, e é implícita. Para resolver o problema foi aplicado o Método da Bissecção.

2. Metodologia

2.1. Método da Mínima Curvatura

Ao longo da perfuração de um poço de petróleo são realizadas medições do comprimento perfurado (comumente chamado de comprimento medido), inclinação (ângulo com relação à direção vertical) e azimute (ângulo entre a direção horizontal e o norte). A partir das coordenadas geográficas do ponto inicial do poço e deste conjunto de medições ao longo da trajetória, é possível calcular as coordenadas cartesianas (N, E, V) de qualquer posição do poço. A Figura 1 apresenta um esquema dos parâmetros de perfuração de um poço direcional:

- ΔS : comprimento medido entre dois pontos ao longo da trajetória.
- θ : inclinação do poço no ponto atual.
- ϕ : azimute do poço no ponto atual.
- α : curvatura entre dois pontos ao longo da trajetória.

*Relatório número 1 como parte dos requisitos da disciplina IM253: Métodos Numéricos para Fenômenos de Transporte.

**Atualmente cursando doutorado no Departamento de Engenharia de Petróleo da Faculdade de Engenharia Mecânica da UNICAMP (Campinas/SP, Brasil).

Email address: t100675@dac.unicamp.br (Tiago C. A. Amorim)

¹Um *blowout* é um evento indesejado, de produção descontrolada de um poço.

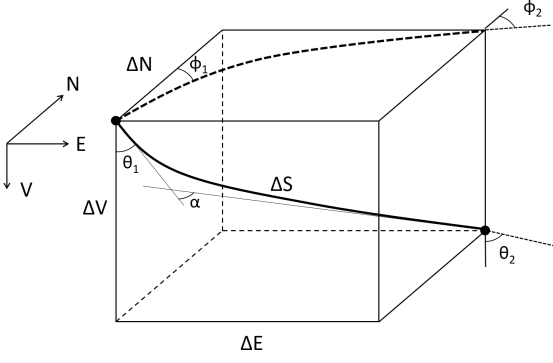


Figura 1: Parâmetros de perfuração entre dois pontos ao longo de um poço direcional.

As fórmulas que associam os parâmetros de perfuração e as coordenadas cartesianas de dois pontos ao longo de um poço direcional segundo o Método da Mínima Curvatura (equações 2.1) são discutidas em detalhe em [2].

$$\Delta N = \frac{\Delta S}{2} f(\alpha) (\sin \theta_1 \cos \phi_1 + \sin \theta_2 \cos \phi_2) \quad (2.1a)$$

$$\Delta E = \frac{\Delta S}{2} f(\alpha) (\sin \theta_1 \sin \phi_1 + \sin \theta_2 \sin \phi_2) \quad (2.1b)$$

$$\Delta V = \frac{\Delta S}{2} f(\alpha) (\cos \theta_1 + \cos \theta_2) \quad (2.1c)$$

$$(2.1d)$$

O termo $f(\alpha)$ é definido com:

$$\alpha = 2 \arcsin \sqrt{\sin^2 \frac{\theta_2 - \theta_1}{2} + \sin \theta_1 \sin \theta_2 \sin^2 \frac{\phi_2 - \phi_1}{2}} \quad (2.2a)$$

$$f(\alpha) = \begin{cases} 1 + \frac{\alpha^2}{12} \{1 + \frac{\alpha^2}{10} [1 + \frac{\alpha^2}{168} (1 + \frac{31\alpha^2}{18})]\}, & \text{se } \alpha < 0.02 \\ \frac{2}{\alpha} \tan \frac{\alpha}{2}, & \text{c.c.} \end{cases} \quad (2.2b)$$

A proposta de método para calcular os parâmetros de perfuração a partir das coordenadas cartesianas parte de manipulações das equações 2.1. É assumido que para calcular os parâmetros de perfuração entre dois pontos quaisquer são conhecidos os parâmetros de perfuração do ponto inicial³ (θ_1, ϕ_1) e as distâncias cartesianas entre os pontos ($\Delta N, \Delta E, \Delta V$). O objetivo é calcular a direção do poço no ponto seguinte (θ_2 e ϕ_2) e a distância ao longo da trajetória entre os pontos (ΔS)⁴. Por simplicidade, ao longo deste relatório este problema será chamado de **Inverso da Mínima Curvatura**.

² $DLS[^\circ/30m] = \alpha/\Delta S \cdot 30$, com α em graus e ΔS em metros.

³Para o primeiro ponto da trajetória é assumido um poço na vertical: $\theta = 0, \phi = 0$.

⁴Este termo é usualmente conhecido por distância medida, ou MD.

É possível inverter a equação 2.1c para obter uma expressão para θ_2 :

$$\cos \theta_2 = 2 \frac{\Delta V}{\Delta S f(\alpha)} - \cos \theta_1 \quad (2.3)$$

Dividindo a equação 2.1a pela equação 2.1b obtém-se duas expressões para ϕ_2 :

$$\sin \phi_2 = \frac{-\Delta N}{\Delta H^2} \left(\frac{\Delta \Psi \sin \theta_1}{\sin \theta_2} \right) + \frac{\Delta E}{\Delta H^2} \sqrt{\Delta H^2 - \left(\frac{\Delta \Psi \sin \theta_1}{\sin \theta_2} \right)^2} \quad (2.4a)$$

$$\cos \phi_2 = \frac{\Delta E}{\Delta H^2} \left(\frac{\Delta \Psi \sin \theta_1}{\sin \theta_2} \right) + \frac{\Delta N}{\Delta H^2} \sqrt{\frac{1}{\Delta H^2} - \left(\frac{\Delta \Psi \sin \theta_1}{\sin \theta_2} \right)^2} \quad (2.4b)$$

onde

$$\begin{aligned} \Delta \Psi &= \Delta N \sin \phi_1 - \Delta E \cos \phi_1 \\ \Delta H^2 &= \Delta N^2 + \Delta E^2 \end{aligned}$$

Fazendo a soma dos quadrados das equações 2.1 é possível obter uma expressão para $\Delta S f(\alpha)$:

$$\Delta S f(\alpha) = 2 \sqrt{\frac{\Delta N^2 + \Delta E^2 + \Delta V^2}{A^2 + B^2 + C^2}} \quad (2.5)$$

onde

$$\begin{aligned} A &= \sin \theta_1 \cos \phi_1 + \sin \theta_2 \cos \phi_2 \\ B &= \sin \theta_1 \sin \phi_1 + \sin \theta_2 \sin \phi_2 \\ C &= \cos \theta_1 + \cos \theta_2 \end{aligned}$$

Com as equações propostas é possível construir um algoritmo para cálculo de $\Delta S f(\alpha)$ como uma função do tipo $g(x) = x$ (Algoritmo 1).

Ao utilizar $\Delta S f(\alpha)$ como parâmetro principal, evita-se calcular α e $f(\alpha)$ durante o processo iterativo (Equações 2.2b). Os valores de θ_2 e ϕ_2 só precisam ser calculados ao final, evitando usar arccos e arcsin muitas vezes. Alguns cuidados adicionais precisam ser tomados ao utilizar este algoritmo:

- Por construção, o valor mínimo de $\Delta S f(\alpha)$ é uma linha reta entre os pontos:

$$\Delta S f(\alpha) \geq \sqrt{\Delta N^2 + \Delta E^2 + \Delta V^2}$$

- $\Delta S f(\alpha)$ tem um segundo limite inferior a ser atendido, definido pelos valores limite da equação 2.3 quando $\Delta V \neq 0$:

$$\Delta S f(\alpha) \geq \begin{cases} \Delta V \frac{2}{\cos \theta_1 + 1}, & \text{se } \Delta V > 0 \\ \Delta V \frac{2}{\cos \theta_1 - 1}, & \text{se } \Delta V < 0 \end{cases}$$

- Se $\theta_2 = 0$, então ϕ_2 fica indefinido. Neste caso a recomendação é fazer $\phi_2 = \phi_1$.
- Se $\Delta N = \Delta E = 0$ então $|\phi_1 - \phi_2| = \pi$.

Algorithm 1 Algoritmo do Inverso da Mínima Curvatura

Entrada: $\Delta N, \Delta E, \Delta V, \theta_1, \phi_1, \Delta S f(\alpha)_{in}$ **Repete**

$\cos \theta_2 \leftarrow f(\Delta S f(\alpha)_{in}) \quad \triangleright 2.3$
 $\sin \theta_2 \leftarrow \sqrt{1 - \cos^2 \theta_2}$
 $\sin \phi_2 \leftarrow f(\cos \theta_2, \sin \theta_2) \quad \triangleright 2.4a$
 $\cos \phi_2 \leftarrow f(\cos \theta_2, \sin \theta_2) \quad \triangleright 2.4b$
 $\Delta S f(\alpha)_{out} \leftarrow f(\cos \theta_2, \sin \theta_2, \cos \phi_2, \sin \phi_2) \quad \triangleright 2.5$
Se $|\Delta S f(\alpha)_{out} - \Delta S f(\alpha)_{in}| > \epsilon$ **então**
 \triangleright Função para melhorar a estimativa de $\Delta S f(\alpha)$
 $\Delta S f(\alpha)_{in} = f(\Delta S f(\alpha)_{in}, \Delta S f(\alpha)_{out})$

Até $|\Delta S f(\alpha)_{out} - \Delta S f(\alpha)_{in}| < \epsilon$

$\cos \theta_2 \leftarrow f(\Delta S f(\alpha)_{out}) \quad \triangleright 2.3$
 $\sin \theta_2 \leftarrow \sqrt{1 - \cos^2 \theta_2}$
 $\theta_2 \leftarrow \arccos(\cos \theta_2) \quad \triangleright 0 \leq \theta_2 \leq \pi$
 $\sin \phi_2 \leftarrow f(\cos \theta_2, \sin \theta_2) \quad \triangleright 2.4a$
 $\cos \phi_2 \leftarrow f(\cos \theta_2, \sin \theta_2) \quad \triangleright 2.4b$
 $\phi_2 \leftarrow f(\cos \phi_2, \sin \phi_2) \quad \triangleright 0 \leq \theta_2 < 2\pi \rightarrow \arcsin e \arccos$
 $f(\alpha) \leftarrow f(\cos \theta_2, \sin \theta_2, \cos \phi_2, \sin \phi_2) \quad \triangleright 2.2$
 $\Delta S \leftarrow \Delta S f(\alpha)_{out} / f(\alpha)$

Retorna: $\Delta S, \theta_2, \phi_2$

2.2. Método da Bissecção

Uma equação do tipo $g(x) = x$ pode ser resolvida buscando a raiz de $f(x) = x - g(x)$. Nesta primeira tentativa foi implementado o Método da Bissecção para resolver o Inverso da Mínima Curvatura. O algoritmo implementado foi baseado no pseudo-código descrito em [3]. Uma forma simplificada é apresentada no Algoritmo 2. O Método da Bissecção baseia-se no teorema do valor médio. O intervalo de busca pela raiz é sucessivamente dividido em dois. O método tem garantia de que a raiz pertence ao intervalo ao manter os valores da função avaliada nos limites do intervalo com sinais opostos⁵.

Algorithm 2 Método da Bissecção

Entrada: x_a, x_b **Se** $\text{sin}al(f(x_a)) = \text{sin}al(f(x_b))$ **então** \triangleright Erro de Definição dos Limites!**Enquanto** $|x_b - x_a| > \epsilon$ **faça** $x_{medio} \leftarrow \frac{x_a + x_b}{2}$ **Se** $\text{sin}al(f(x_a)) = \text{sin}al(f(x_{medio}))$ **então** $x_a \leftarrow x_{medio}$ **Caso contrário** $x_b \leftarrow x_{medio}$ **Retorna:** x_{medio}

Foram adicionados critérios adicionais ao algoritmo para controlar o processo iterativo:

- Foram implementados dois métodos de cálculo do critério de convergência:

⁵Assumindo que o intervalo inicial fornecido também tem esta propriedade.

– Critério *Direto*: $|x_i - x_{i-1}|$.– Critério *Relativo*⁶: $\frac{|x_i - x_{i-1}|}{|x_i|}$.

- No início do código é verificado se $|x_b - x_a| < \zeta$, onde ζ é calculado em função do critério de convergência estabelecido⁷. Se for *verdadeiro*, não é feito o *loop* do método.
- Se $\text{sin}al(f(x_a)) = \text{sin}al(f(x_b))$ o código apresenta uma mensagem de alerta e não é feito o *loop* do método. Optou-se por não gerar um erro, e mesmo neste caso é retornado um valor.
- É feito um término prematuro do processo iterativo caso algum $|f(x)| < \epsilon$. O valor padrão de ϵ é 10^{-7} (variável *epsilon* no código). Este teste também é feito antes de entrar no *loop*.
- Antes de sair da função, são comparados os três últimos resultados guardados ($f(x_a)$, $f(x_b)$, $f(x_{medio})$) e é retornado o valor de x com $f(x)$ mais próximo de zero.

3. Resultados

Para facilitar a análise da qualidade do código desenvolvido, foram criadas funções que realizam diversos testes onde a resposta exata é conhecida:

tests_bisection() Testa o Método da Bissecção em diferentes funções: linear, quadrática, exponencial e trigonométrica. Também foram aplicados casos específicos para o algoritmo tratar: raiz em um dos limites, uso de critério de convergência relativo, saída do *loop* sem atingir o critério de convergência, má definição do intervalo inicial (função com mesmo sinal nos limites) e intervalo inicial muito pequeno ($|x_b - x_a| < \zeta$).

tests_minimum_curvature() Testa o algoritmo proposto para resolver o inversa da mínima curvatura para diferentes geometrias de trecho de poço: poço vertical, horizontal, inclinado, em curva.

Algumas das definições que foram feitas no código que implementa o Método da Bissecção são resultado dos testes realizados.

A definição de um critério de parada prematura se mostrou importante para evitar que o método continue buscando uma raiz quando já encontrou uma solução *aceitável*. A definição deste limite $|f(x)| < \epsilon = 10^{-7}$ foi empírica e deve ser revista em função do problema a ser resolvido.

⁶Caso $|x_i| < \epsilon$, é utilizado $|x_{i-1}|$ no denominador. E se também $|x_{i-1}| < \epsilon$ o valor da convergência é considerado *zero*!

⁷Dado que c_{lim} é o limite de convergência, se for utilizado o critério de convergência *direto* então $\zeta = c_{lim}$. Se for utilizado o critério de convergência relativo então $\zeta = c_{lim} \min |x_a|, |x_b|$ (a avaliação do menor valor segue as mesmas regras do cálculo do critério de convergência, ignorando qualquer $|x| < \epsilon$ e retorna *zero* caso ambos valores são pequenos).

Todas as funções foram definidas para trabalhar com números do tipo `double`. Inicialmente as funções estavam definidas para trabalhar com `float`, mas estes mostraram não conseguirem trabalhar com valores de convergência muito baixos. A avaliação da função $f(x) = -3x + 0.9$ na sua raiz foi testada usando diferentes tipos de números de ponto flutuante:

- `float`: $f(0.3) \approx -3.57628 \cdot 10^{-8}$
- `double`: $f(0.3) \approx 1.11022 \cdot 10^{-16}$
- `long double`: $f(0.3) \approx 5.35872 \cdot 10^{-312}$

Mesmo que o `long double` consiga o melhor resultado, considerou-se que trabalhar com `double` já é *suficiente*. Foi feita uma sensibilidade do critério de convergência (c_{lim}) a um dos testes do problema do inverso da mínima curvatura (trecho de poço com mudança em inclinação e direção). Nesta sensibilidade foi possível trabalhar com valores de c_{lim} baixos sem aparente perda da qualidade da resposta por problemas com aritmética de máquina (Figura 2).

É sempre feita uma verificação ao final do código para avaliar qual é o valor entre x_a , x_b e x_{medio} que minimiza $|f(x)|$. O método da bissecção tem garantia de convergência, mas não é garantido que o melhor resultado será o x_{medio} da última iteração.

A metodologia proposta para resolver o inverso do método da mínima curvatura tem algumas particularidades, como a sua indefinição quando é testado com um valor muito baixo de $\Delta Sf(\alpha)$. O Método da Bissecção se encaixa bem na resolução do problema proposto por trabalhar em uma faixa de valores pré-determinada. Devido a erros de aritmética de máquina, não é sempre possível garantir que $\text{sin}(f(x_a)) \neq \text{sin}(f(x_b))$ quando a resposta exata está muito próxima do limite inferior. Desta forma, ao invés de gerar um erro quando $\text{sin}(f(x_a)) = \text{sin}(f(x_b))$, é gerada uma mensagem de alerta ao usuário e o código retorna o valor de x (igual a x_a ou x_b) que minimizar $|f(x)|$.

Este problema ficou evidente no cálculo do 4º trecho do poço em 'S' descrito nos testes propostos (Figura 3). O valor exato de θ_2 é zero (o 5º trecho é reto e vertical). Neste caso, o valor mínimo de $\Delta Sf(\alpha)$ é limitado pela equação 2.3, e a resposta exata é igual a este limite. Neste ponto o algoritmo calcula $f(263.305) \approx -2.58273 \cdot 10^{-7}$, que não atinge o critério de parada prematura, mas tem o mesmo sinal de $f(\Delta Sf(\alpha)_{max})$.

Os limites da região em que o problema do inverso da mínima curvatura deve ser avaliado são definidos por um lado pelo menor valor possível de $\Delta Sf(\alpha)$ discutido em 2.1. O valor máximo foi definido como $[\Delta Sf(\alpha)]_{min} \cdot f(\alpha = 0.95\pi)$. O limite superior é equivalente a aproximadamente um arco de 180° , o que não é algo factível em uma aplicação real, e que serve como um limite superior na região de busca.

Foi implementada uma função para estimar o número de iterações necessárias para alcançar o critério de convergência definido (equação 3.1). Quando o critério de convergência (c_{lim}) é *direto*, a estimativa se mostrou exata (exceto nos casos em que há parada prematura), o que indica que o algoritmo implantado converge com a taxa que era esperada. Buscou-se realizar a mesma estimativa para o caso de uso do critério de convergência *relativo*, e neste caso as estimativas não exatas, mas tem boa previsão (Tabela 1).

$$n_{int} = \left\lceil \frac{\log \frac{|x_b - x_a|}{|x_{referencia}|} \frac{1}{c_{lim}}}{\log 2} \right\rceil \quad (3.1)$$

onde

$$x_{referencia} = \begin{cases} 1, & \text{se Critério de conv. Direto} \\ \min(|x_a|, |x_b|), & \text{c.c.} \end{cases}$$

A Figura 3 apresenta a evolução do erro, com relação à resposta exata, ao longo do processo iterativo de diferentes funções, utilizando o Método da Bissecção. São apresentadas duas retas tracejadas onde a razão entre os erros de iterações sucessivas é $\frac{1}{2}$, que é o que se espera do Método da Bissecção após um número considerável de iterações. Observa-se que os resultados numéricos não seguem exatamente uma linha reta, mas têm aproximadamente a mesma inclinação das retas tracejadas.

O Método da Bissecção gerou bons resultados na aplicação proposta. Em um dos testes proposto o método não conseguiu ser aplicado porque a resposta correto é igual ao limite inferior e, por erro de aritmética de máquina, o valor da função não é zero neste ponto. Para diversos testes a solução foi trivial, pois a resposta exata é igual ao limite inferior de $\Delta Sf(\alpha)$. Na Tabela 2 todos os testes reportados foram feitos utilizando como critério de parada $|x_{i+1} - x_i| < c_{lim} = 1E - 3$.

O código de ambos métodos foi implementado em um único arquivo. O código é apresentado em duas partes neste documento para facilitar a leitura. O código pode ser encontrado em <https://github.com/TiagoCAAmorim/numerical-methods>.

4. Conclusão

O algoritmo proposto para o cálculo de parâmetros de perfuração em função das coordenadas cartesianas de pontos ao longo de um poço se mostrou eficaz. Foram feitos ajustes ao código implementado de forma a evitar o uso de funções trigonométricas ao longo do processo iterativo. O Método da Bissecção se mostrou adequado para uso com o algoritmo proposto. As funções propostas não são válidas para quaisquer valores de entrada, e a delimitação de uma região de busca foi importante para garantir a convergência do problema.

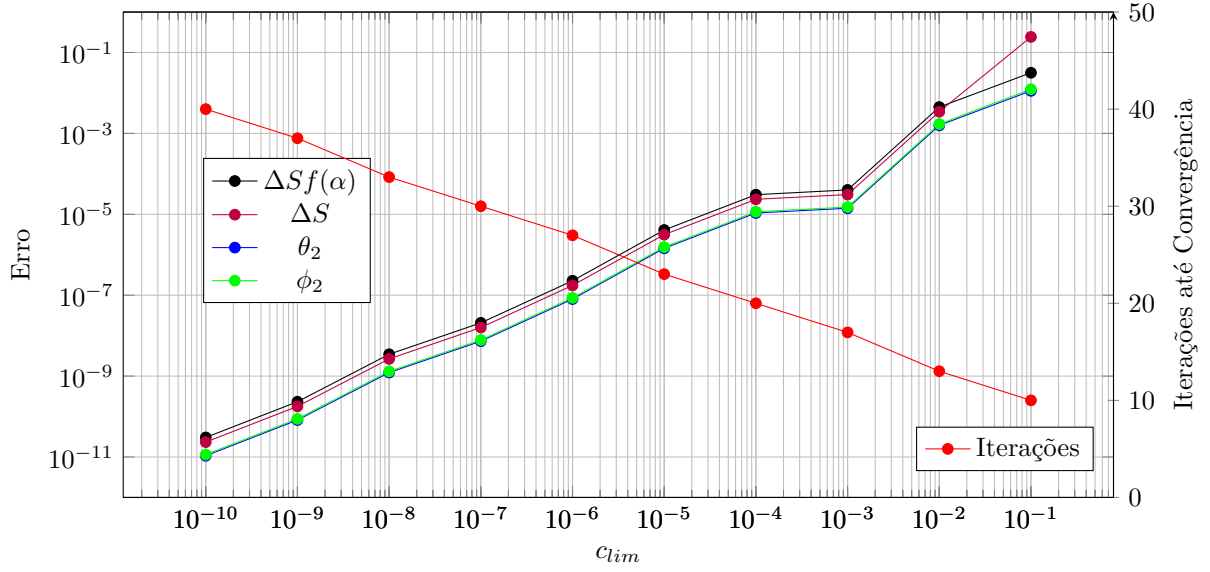


Figura 2: Erro das variáveis de interesse em função do limite de convergência utilizado.

Tabela 1: Comparação entre o número de iterações previsto e realizado para diferentes testes.

Função	x_a	x_b	Critério <i>Direto</i>		Critério <i>Relativo</i>	
			Previsão	Realizado	Previsão	Realizado
Linear	0.	2.	11	11	11	13
Quadrática	-0.25	1.	11	11	11	14
Exponencial	0.	10.	14	14	14	13
Trigonométrica	0.	5.	13	13	13	12
1/4 círculo horizontal	14.1421	120.417	17	17	13	13
Seção 2 do poço em S	130.526	1111.4	20	20	13	13
Poço 3D	9.84918	83.8634	17	17	13	13

Tabela 2: Erro das respostas de interesse com relação ao valor exato para testes realizados com algoritmo proposto e o Método da Bissecção.

Teste	Iter.	$\Delta S f(\alpha)$	θ_2	ϕ_2	ΔS
Linha reta inclinada	0	0	0	1.11e-16	0
1/4 de círculo	7	2.52e-4	1.26e-5	0	1.26e-4
Seção 2 do poço em S	20	4.37e-5	2.52e-6	0	2.89e-05
Poço 3D	17	4.00e-5	1.41e-5	1.51e-5	3.07e-05

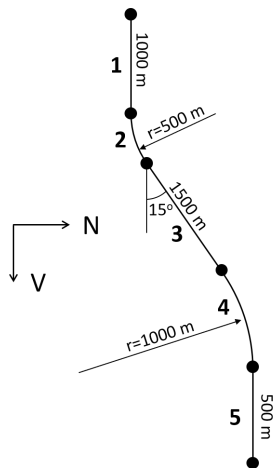


Figura 3: Esquema do poço em 'S' descrito nos testes de `tests_minimum_curvature()`.

Referências

- [1] I. A. of Drilling Contractors (IADC), IADC Drilling Manual, International Association of Drilling Contractors (IADC), 2015, prévia do livro em <https://iadc.org/wp-content/uploads/2015/08/preview-dd.pdf> (accessado em 28/08/2023).
- [2] A Compendium of Directional Calculations Based on the Minimum Curvature Method, Vol. All Days of SPE Annual Technical Conference and Exhibition. **arXiv:** <https://onepetro.org/SPEATCE/proceedings-pdf/03ATCE/A11-03ATCE/SPE-84246-MS/2895686/spe-84246-ms.pdf>, doi:10.2118/84246-MS. URL <https://doi.org/10.2118/84246-MS>
- [3] R. L. Burden, J. D. Faires, A. M. Burden, Análise numérica, Cengage Learning, 2016.

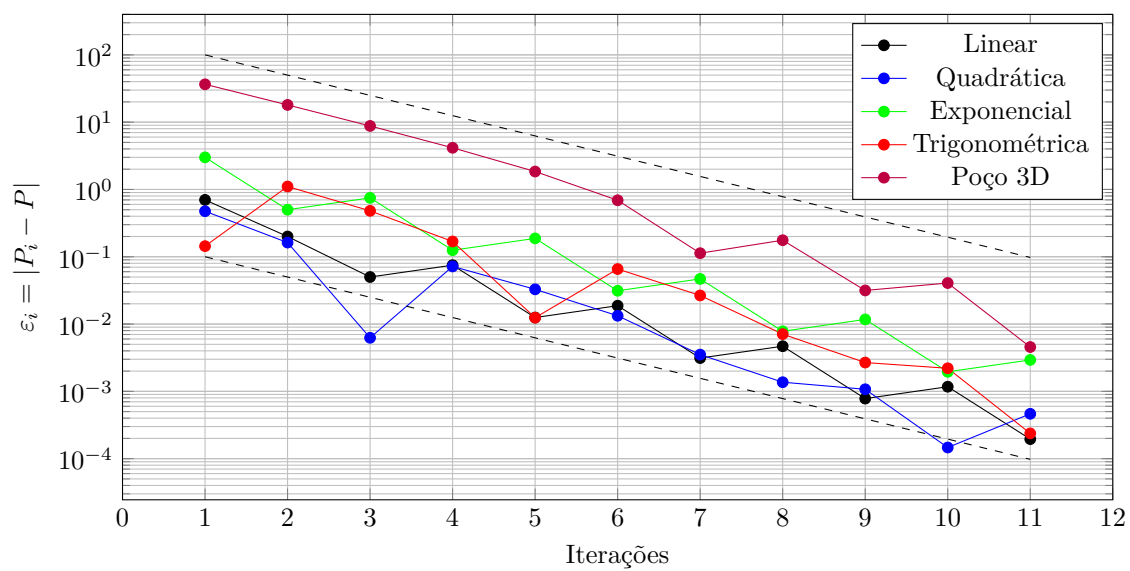


Figura 4: Evolução do erro do Método da Bissecção com diferentes funções (linhas tracejadas representam $\varepsilon_{i+1}/\varepsilon_i = 0.5$).