

CPE-737 Otimização Aspectos Teóricos e Métodos Numéricos

Projeto #1

Métodos numéricos para otimização escalar de funções monovariáveis

trabalho realizado por

Arthur DOS SANTOS XAUD
Carolina CALVO POSE SANTOS NEVES
Gabriel FELIPPE DA CRUZ PACHECO

Prof. Afonso DEL NERO GOMES

8 de novembro de 2015

Sumário

1	Considerações Iniciais	1
1.1	Enquadramento Inicial	1
2	Método de Fibonacci	2
2.1	Algoritmo	2
2.2	Resultado	4
3	Seção Áurea	4
3.1	Algoritmo	4
3.2	Resultado	6
4	Interpolação Parabólica e o Método de Brent	6
4.1	Princípios básicos	6
4.2	Algoritmo	7
4.3	Resultado	9

1 Considerações Iniciais

Nos algoritmos de otimização apresentados nas seções a seguir (Método de Fibonacci, Seção Áurea e Interpolação Polinomial), considera-se que as funções de entrada de cada um desses algoritmos são suaves e unimodais no intervalo enquadrante passado como argumento inicialmente. Tais funções são também tais que $f: \mathbb{R} \rightarrow \mathbb{R}$.

Neste relatório, apresenta-se o resultado da convergência em cada um dos três métodos para a mesma função, presente na equação 1. Outras funções podem ser testadas utilizando a interface gráfica em anexo a este relatório, conforme demonstrado na apresentação do trabalho em sala de aula.

$$f = x^4 + 10x^3 \quad (1)$$

1.1 Enquadramento Inicial

O problema de busca de um mínimo para uma dada função em um determinado intervalo envolve o conceito de unimodalidade de funções. Uma função unimodal em um dado intervalo possui apenas um mínimo naquele intervalo. Dessa maneira, se uma função f é unimodal em um intervalo $[a, b]$, então este intervalo enquadra um mínimo e é um intervalo de interesse nos métodos numéricos.

Para o método de brent, caso um intervalo seja escolhido, antes da otimização propriamente dita, faz-se um teste para verificar se o intervalo em questão é enquadrante.

Neste caso, utilizou-se o critério dos 3 pontos: sejam x_1, x_2 e $x_3 \in \mathbb{R}$ tais que $x_1 < x_2 < x_3$ e uma função $f: \mathbb{R} \rightarrow \mathbb{R}$ suave. Se $f(x_1) \geq f(x_2) < f(x_3)$, então o intervalo $[x_1, x_3]$ enquadra um mínimo. Vale ressaltar que o resultado continua válido se $x_1 > x_2 > x_3$. É importante frizar também que o critério dos três pontos garante que o intervalo é enquadrante mas não necessariamente unimodal, o que significa que pode haver mais de um mínimo neste intervalo.

Na situação em que não há um palpite inicial sobre o intervalo alvo para qualquer um dos métodos deste trabalho, procura-se, a partir de x_1, x_2 e x_3 que satisfaçam o critério dos três pontos. Dessa forma, acrescenta-se um Δ a x_1 e uma comparação de f_2 e f_1 é realizada. Se $f_2 \leq f_1$, procura-se x_3 neste mesmo sentido até encontrar x_3 tal que $f_3 > f_2$. Caso contrário, caso $f_2 > f_1$, inverte-se o sentido da busca. O algoritmo completo do enquadramento inicial por três pontos pode ser facilmente entendido através do diagrama de fluxo da figura 1.

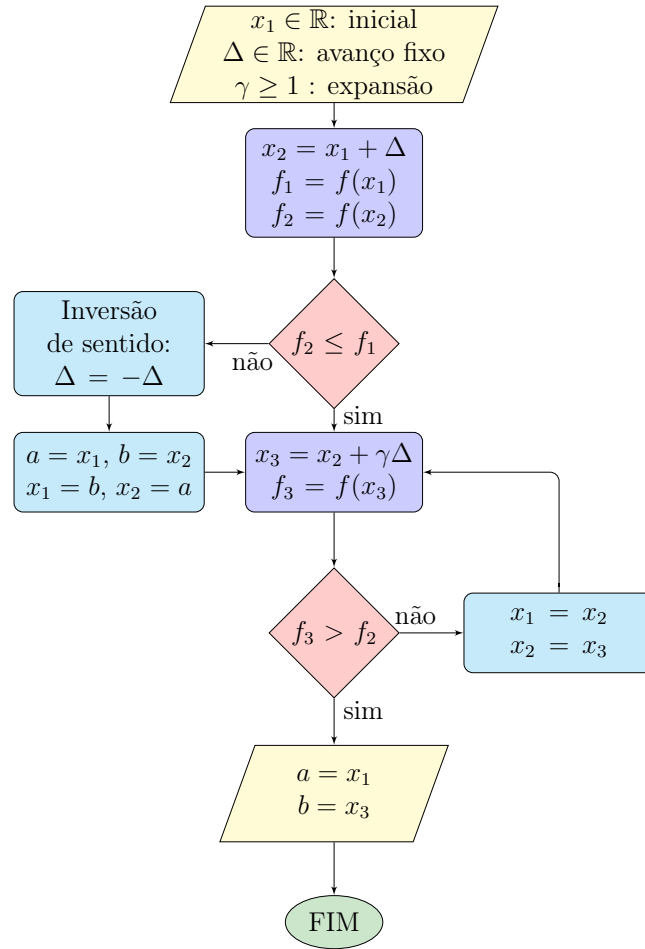


Figura 1: Enquadramento Inicial por três pontos

2 Método de Fibonacci

2.1 Algoritmo

Neste método, dado um intervalo enquadrante $I_k = [a, b]$, no qual a função f é unimodal, escolhem-se dois pontos internos simétricos em relação ao centro, conforme a figura 2.

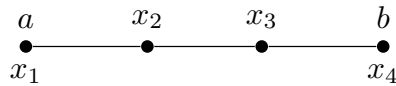


Figura 2: Pontos do método de Fibonacci

Um fator de redução inicial $0.5 < \alpha_1 < 0.7$ é usado para encontrar os primeiros valores de x_2 e x_3 dentro do intervalo enquadrante inicial $I_1 = [a, b] \subset \mathbb{R}$.

Então, a função-objetivo é avaliada nesses dois pontos x_2 e x_3 e o intervalo é reduzido de acordo com a comparação dos valores de $f(x_2) = f_2$ e $f(x_3) = f_3$: se $f_2 < f_3$, o novo intervalo é definido como $I_{k+1} = [x_1 x_3]$. Caso contrário, o intervalo seguinte passa a ser $I_{k+1} = [x_2 x_4]$. Finalmente, o fator de redução é recalculado de acordo com a equação 2 e o procedimento se repete até que o número de reduções desejado $n \in \mathbb{Z}$ seja atingido.

$$\alpha = \frac{L_{ini} - L_{fim}}{L_{fim}} \quad (2)$$

Novamente, o diagrama de fluxo do algoritmo implementado é apresentado na figura 3.

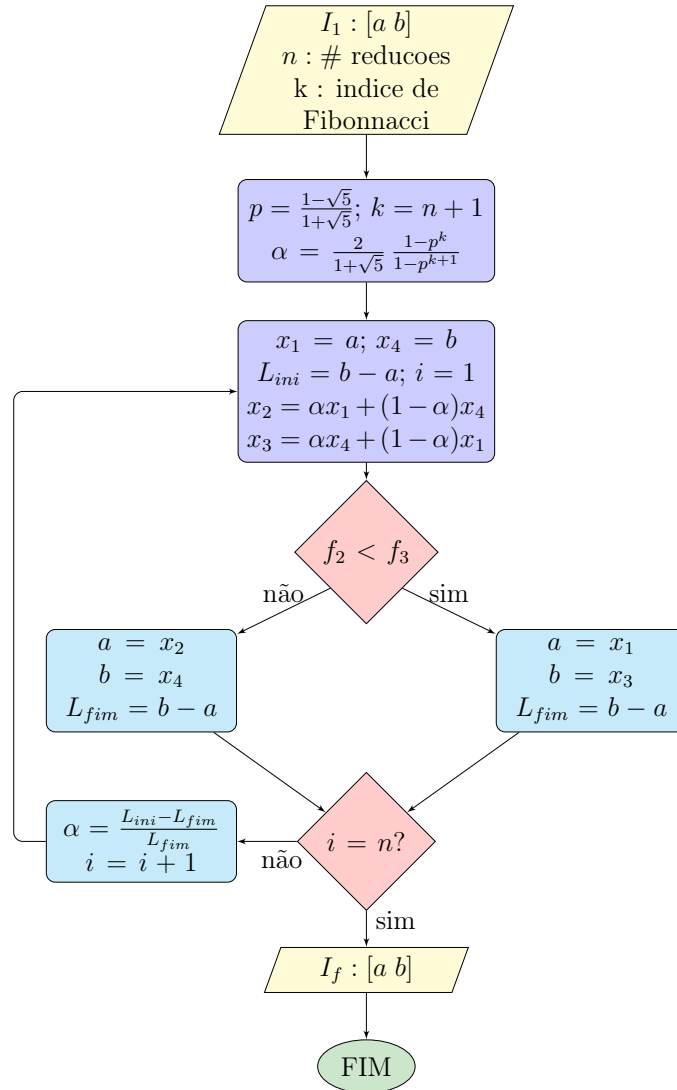


Figura 3: Método de Fibonacci

2.2 Resultado

O resultado obtido utilizando a função teste enunciada em 1 é apresentado na figura 4

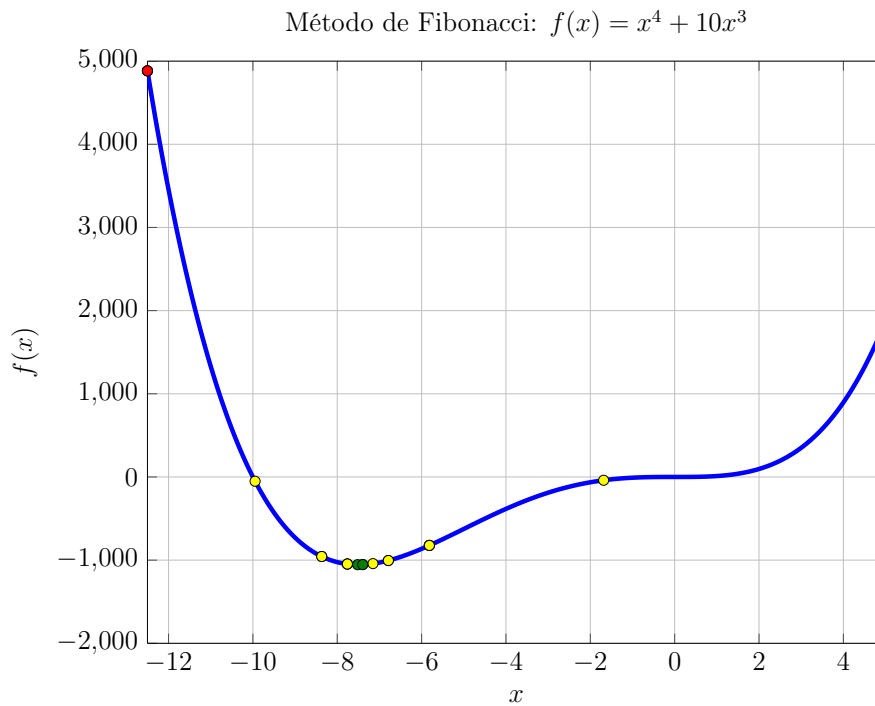


Figura 4: Número de iterações: $n = 10$. Tamanho do último intervalo $L_f = 0.1215$

O número de iterações escolhido foi $n = 10$ para que o método fosse suficientemente distinto do método da seção áurea. Tal número de iterações nos fornece um intervalo final $L_f = 0.1215$. O resultado poderia ser melhorado arbitrariamente apenas aumentando o valor de n .

3 Seção Áurea

3.1 Algoritmo

Este método se baseia na divisão áurea de um segmento. Dado um segmento de reta \overline{AB} , dividi-lo na chamada razão áurea significa achar um ponto M deste segmento tal que o segmento todo está para a parte maior assim como esta parte maior está para a parte menor, conforme a figura 5.

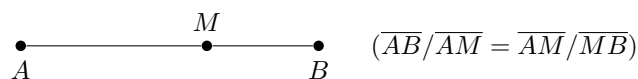


Figura 5: Divisão áurea de um segmento

O valor da razão entre esses segmentos que realiza isso é dada pela equação 3.

$$u = \frac{\sqrt{5} - 1}{2} \approx 0.6180 \quad (3)$$

O método da seção áurea é bastante parecido com o método de Fibonacci. A diferença reside no fato de que os pontos centrais x_2 e x_3 são determinados através da razão áurea da equação 3. Além disso, nesse caso, não se determina um número desejado de reduções como no método de Fibonacci, mas sim uma tolerância $\varepsilon \in \mathbb{R}$ para o intervalo final.

De fato, um inconveniente do algoritmo de Fibonacci está relacionado à necessidade de se conhecer a priori o número total de reduções e o índice k associado, ao passo que, no método da divisão áurea, isso não é necessário. Na verdade, demonstra-se inclusive que, se o número de reduções prevista no algoritmo de Fibonacci for grande ($n > 10$), a primeira redução do método de Fibonacci corresponde justamente à divisão áurea.

O diagrama de fluxo da implementação do algoritmo da seção áurea é apresentado na figura 6.

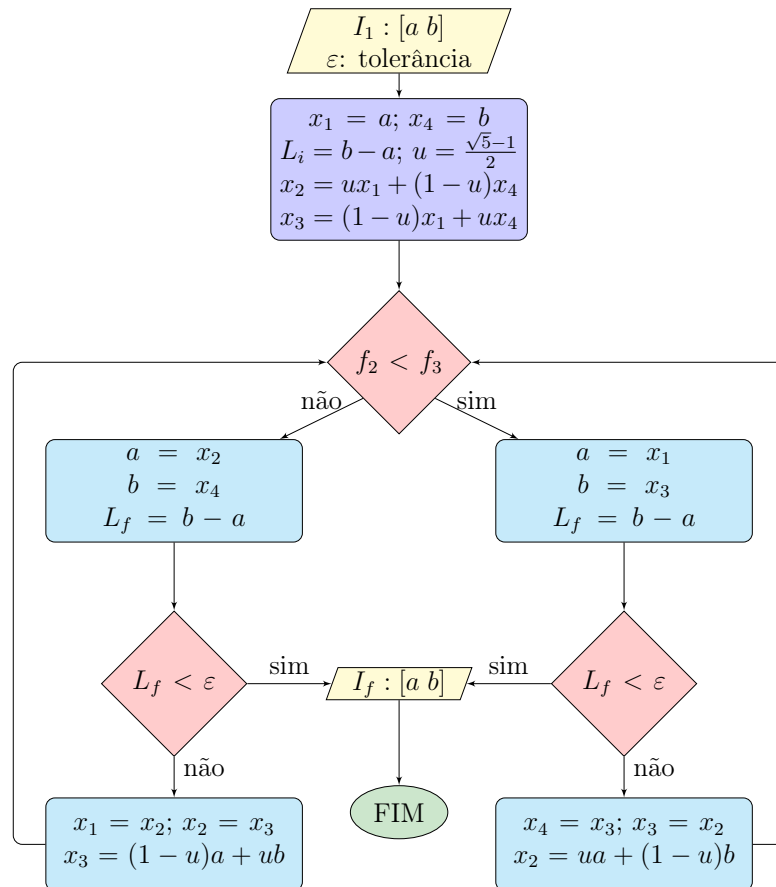


Figura 6: Seção Áurea

3.2 Resultado

O resultado obtido utilizando a função teste enunciada em 1 é apresentado na figura 7.

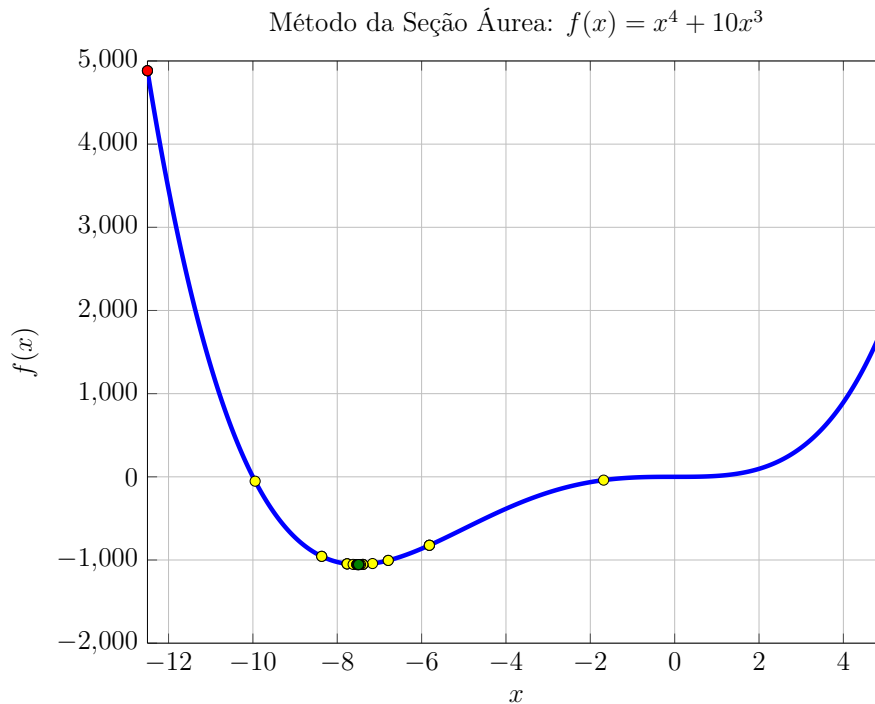


Figura 7: Tolerância: $\epsilon = 0.01$. Número de iterações: $n = 17$

Para atingir a tolerância estipulada de $\epsilon = 0.01$, foram necessárias 17 iterações do algoritmo. O algoritmo da seção áurea é extremamente robusto e seguro, no entanto, veremos como uma aproximação da função objetivo por um polinômio quadrático mesclado ao algoritmo da seção áurea pode ter uma performance superior.

4 Interpolação Parabólica e o Método de Brent

4.1 Princípios básicos

A idéia por trás desse método é utilizar uma aproximação da função $f(x)$ por um polinômio de grau 2 $g(x)$ que passa por três pontos pertencentes a um intervalo enquadrante. Dessa maneira, aproxima-se o mínimo de $f(x)$ pelo mínimo da parábola obtida. A fórmula 4 permite encontrar a abscissa x correspondente ao mínimo da parábola para os pontos a , b e c .

$$x = b - \frac{1}{2} \frac{(b-a)^2[f(b) - f(c)] - (b-c)^2[f(b) - f(a)]}{(b-a)[f(b) - f(c)] - (b-c)[f(b) - f(a)]} \quad (4)$$

Redefine-se, então, o intervalo para a próxima iteração de acordo com a avaliação da função objetivo nesses três pontos, descartando o pior deles e guardando o minimizador desta parábola aproximada como novo ponto. Tal interpolação parabólica é dita inversa pois buscamos a abscissa x e não a ordenada $f(x)$.

Da expressão 4, podemos notar que existe uma singularidade quando os três pontos são colineares. Além disso, o método funciona da mesma maneira tanto na presença de um mínimo quanto de um máximo.

Por isso, para contornar tais dificuldades, a estratégia proposta por Brent é utilizar uma busca segura (ex: seção-áurea) quando que a função for não cooperativa e trocar para para interpolação parabólica inversa sempre que possível. Assim, incorpora-se a robustez do método da seção-áurea com a rapidez da interpolação quadrática.

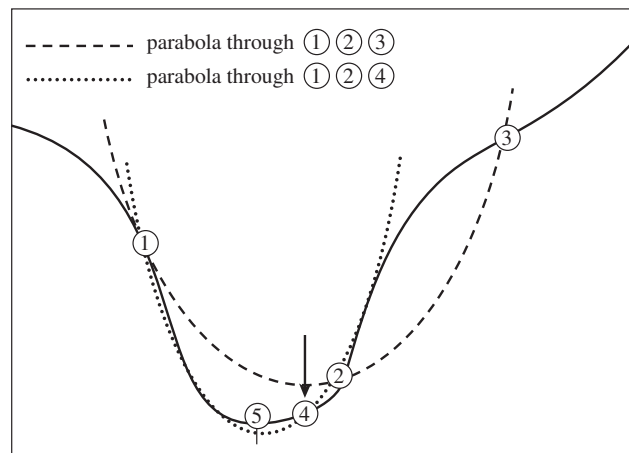


Figura 8: Convergência pelo Método de Brent. Figura retirada de [1]

A figura 8 mostra duas iterações da interpolação parabólica. É possível notar que a primeira estimativa do minimizador é obtida em 4 e que, em seguida, esta muda para 5, se aproximando ainda mais do real valor do minimizador da função $f(x)$.

4.2 Algoritmo

Durante qualquer fase do algoritmo, este possui um conjunto de 6 pontos (a, b, u, v, w, x) . Os pontos a e b definem o intervalo enquadrante de cada iteração; x é o minimizador de $f(x)$ obtido até o momento; w é o ponto de segundo menor valor da função objetivo até o momento e v é o valor anterior de w ; u , por sua vez, é o ponto onde a função foi avaliada mais recentemente. Um outro ponto que também aparece no algoritmo é o ponto médio entre a e b , i.e x_m , no entanto, a função objetivo não é avaliada nele.

Se após o passo d da interpolação parabólica, o melhor ponto x pertencer ao intervalo $[a, b]$ e, ainda, se o deslocamento provocado por este passo for inferior ao deslocamento provocado na penúltima iteração, significa que x está convergindo e que a interpolação parabólica pode ser usada. Caso alguma dessas condições não seja satisfeita, o passo d é calculado utilizando a razão áurea do maior segmento dentro do intervalo $[a, b]$.

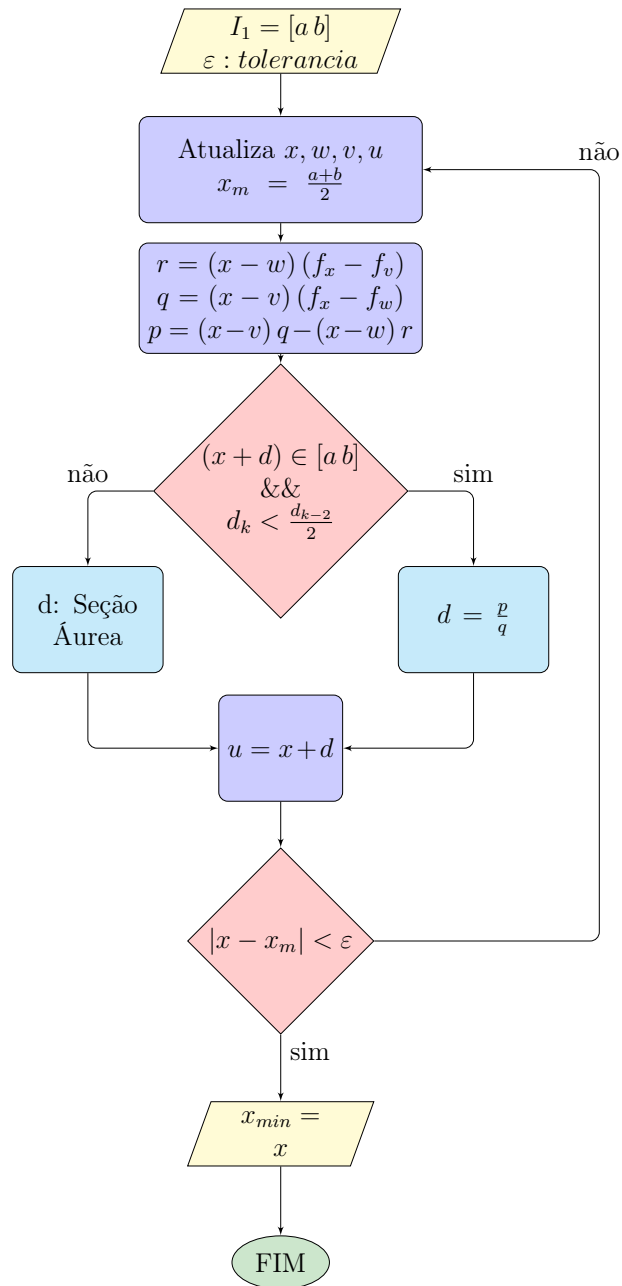


Figura 9: Fluxograma do Método de Interpolação

O algoritmo retoma o início dos cálculos dos novos pontos (a, b, u, v, w, x) até que

a tolerância seja atingida ($|x - x_m| < tol$), alternando, portanto, entre a interpolação polinomial e o método da razão áurea na determinação dos intervalos para as próximas iterações. O diagrama de fluxo 9 exemplifica o algoritmo.

4.3 Resultado

O resultado obtido para o método de Brent utilizando a função teste apresentada previamente neste relatório (eq. 1) pode ser observado na figura 10.

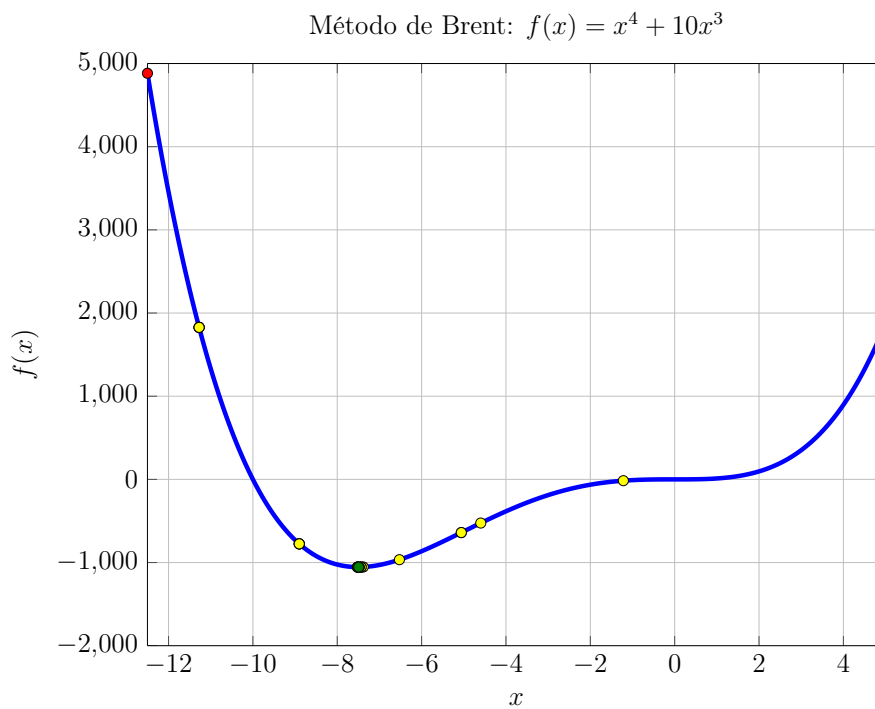


Figura 10: Tolerância: $\epsilon = 0.01$. Número de iterações: $n = 12$

É interessante notar que a performance foi bem similar ao método da seção áurea já que a tolerância ϵ escolhida foi a mesma. No entanto, o método de Brent é consideravelmente mais rápido devido à utilização da interpolação polinomial. Ao passo que foram necessárias 17 iterações para Seção Áurea convergir, somente 12 bastaram para garantir o mesmo resultado usando o método de Brent.

Referências

- [1] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. Numerical recipes: The art of scientific computing (cambridge, 1992).