Um Método Híbrido Dicotomia-Secante

EP1 - CompIII - **Data de entrega: 16/09/2021**

1 Introdução

Dada uma função contínua $f : [a, b] \to \mathbb{R}$ tal que $f(a) * f(b) < 0$, o método da dicotomia calcula uma raiz aproximada $\bar{x}$ de f no intervalo (a, b). Porém a convergência é lenta e por essa razão buscam-se métodos de convergência mais rápida para aproximar raízes. Entre eles, o método da secante é muito popular e, após a sua inicialização, envolve apenas uma avaliação de f por iteração.

O método da secante é descrito da seguinte forma. Tendo-se calculado as aproximações x_{k-1} e x_k, a nova aproximação x_{k+1} é obtida pela interseção da reta que une os pontos $(x_{k-1}, f(x_{k-1}))$ e $(x_k, f(x_k))$ com o eixo x. Ou seja, a interseção com eixo x da secante ao gráfico de f que passa por esses dois pontos. Matematicamente, obtemos a expressão (**exercício**)

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k), \tag{1}$$

convenientemente expressa como a aproximação anterior mais uma correção.

Pode-se mostrar que sob algumas hipóteses, para x_0 *suficientemente próximo* da raiz $\bar{x}$, as iterações convergem para $\bar{x}$ de acordo com

$$|\bar{x} - x_{k+1}| \leq C|\bar{x} - x_{k+1}|^p,$$

onde $p = 0.5(1 + \sqrt{5}) \approx 1.618$ (razão áurea!). Temos convergência superlinear mas, como enfatizado acima, somente quando estamos próximos da raiz. Esta é uma dificuldade de métodos que convergem rapidamente: pode-se garantir apenas convergência *local*. Uma maneira de lidar com esta dificuldade é usar um método híbrido, combinando-se um método robusto e lento, como a dicotomia, com um método rápido e local.

2 O método de Dekker

Descreveremos aqui um método híbrido que combina o método da dicotomia com o método da secante, inspirado em ideias de T.J. Dekker[1]. Como no método da dicotomia, iniciamos com um intervalo $[a_0, b_0]$ tal que $f(a_0) * f(b_0) < 0$. Em cada passo:

- b_k é a iteração atual, i.e., a aproximação atual para a raiz $\bar{x}$ de f.

[1] Veja o link `https://en.wikipedia.org/wiki/Brent's_method#Dekker's_method` e o livro L.F. Shampine, R.C. Allen, Jr., S. Pruess, *Fundamentals of Numerical Computing*, John Wiley & Sons, Inc., 1997, para mais detalhes.

- a_k é o ponto antípoda, i.e., um ponto tal que $f(a_k)$ e $f(b_k)$ têm sinais opostos. Além disso, $|f(b_k)|$ deve ser menor ou igual a $|f(a_k)|$, esperando-se com isso que b_k seja uma aproximação melhor do que a_k.

- b_{k-1} é a iteração anterior (quando $k = 1$, faça $b_0 = a_0$).

Para a próxima iteração, dois valores provisórios são calculados. O primeiro é obtido pelo método da secante (1):

$$s = \begin{cases} b_k - \frac{b_k - b_{k-1}}{f(b_k) - f(b_{k-1})} f(b_k) = b_k - \Delta_k, & \text{se } f(b_k) \neq f(b_{k-1}), \\ \frac{a+b}{2}, & \text{caso contrário,} \end{cases}$$

e o segundo é obtido do método da dicotomia:

$$m = \frac{a+b}{2}.$$

Supondo-se que b_k é uma aproximação melhor para a raiz do que a_k e que o método da secante esteja funcionando adequadamente, espera-se que s esteja mais próximo de b_k do que a_k. Logo, se s estiver contido *estritamente* entre b_k e m, então ele será a nova iteração ($b_{k+1} = s$). Caso contrário, o ponto médio é usado ($b_{k+1} = m$). Então, o novo ponto antípoda é calculado. Se $f(a_k)$ e $f(b_{k+1})$ tiverem sinais sinais opostos, então o antípoda permanece: $a_{k+1} = a_k$. Caso contrário, a raiz está entre b_{k+1} e b_k (POR QUE?) e escolhemos $a_{k+1} = b_k$. Finalmente, se $|f(a_{k+1})| < |f(b_{k+1})|$, então a_{k+1} é *provavelmente* uma aproximação melhor para a solução do que b_{k+1}, e então estes dois valores são permutados.

O parágrafo anterior descreve um passo do método de Dekker. As iterações são calculadas até que

$$\left| \frac{b_k - a_k}{2} \right| < \text{TOL}, \tag{2}$$

aceitando-se b_k como a aproximação para a solução. A tolerância TOL é definida por dois parâmetros ERROABS e ERROREL, que especificam erros absoluto e relativo, respectivamente:

$$\text{TOL} = \max\{\text{ERROABS}, |b_k| * \text{ERROREL}\} \quad \text{(pense!)} \tag{3}$$

A prática sugere algumas precauções adicionais para a melhoria do algoritmo. Se s estiver muito próximo de b_k, i.e., $|\Delta_k| < \text{TOL}$, é conveniente deslocar um pouco a aproximação e usar $b_{k+1} = b_k + \text{sinal}(b_k - a_k) * \text{TOL}$. Esta escolha resulta em um valor entre a_k e b_k. Se a raiz $\bar{x}$ estiver a uma distância de b_k maior do que TOL, então esta escolha estará mais próxima da raiz do que s. E caso $\bar{x}$ diste menos do que TOL de b_k, esta aproximação e b_k irão isolar a raiz e o algoritmo irá convergir no próximo teste de erro.

Pode haver situações onde as iterações b_k são calculadas pelo método da secante e estão convergindo para a raiz, mas o ponto antípoda permanece fixo. Nestes casos, os tamanhos dos intervalos isolando a raiz podem diminuir a uma razão lenta. Para nos precavermos, podemos adotar a seguinte estratégia: se após quatro ierações não houver uma redução do tamanho do intervalo por um fator de $1/8$, então o algoritmo usará o método da dicotomia por três vezes consecutivas, antes de retomar as comparações com o método da secante.

Note que não é necessário armazenar todos os extremos dos intervalos. O algoritmo pode usar duas variáveis a e b onde $f(a) * f(b) < 0$, $|f(b)| \leq |f(a)|$ e b é a aproximação atual. Uma outra variável c é usada para armazenar a aproximação anterior.

O método pode então ser resumido assim: se o valor s do método da secante não estiver entre b e m ou se a redução do tamanho do intervalo não for satisfatória, use o método da dicotomia. Se s estiver muito próximo de b, um deslocamento mínimo por TOL é usado. Caso contrário, s é usado. Após decidir como calcular a próxima iteração, ela é obtida explicitamente e substitui b, com a antiga aproximação substituindo c. Se $f(b) = 0$, o algoritmo termina. Caso contrário, as variáveis são atualizadas para a próxima iteração: o antigo a é mantido ou substituído por c, de acordo com a troca de sinal e a e b podem ser permutados, se necessário.

Em uma situação normal, com a e b especifcados na *entrada* satisfazendo $f(a) * f(b) < 0$, após a execução do algoritmo teremos $f(b) = 0$, ou os valores $f(a)$ e $f(b)$ satisfazendo $f(a) * f(b) < 0$, $|f(b)| \leq |f(a)|$ e os valores de *saída* a e b satisfazendo (2), com (3) calculado usando-se b no lugar de b_k.

3 Tarefa

Implemente o método de Dekker conforme descrito na seção anterior, usando a linguagem Python (especificamente, Python 3). Documente bem o programa e pense cuidadosamente em quais informações relevantes fornecer a um usuário, tanto para a entrada como para a saída do programa. No relatório, discuta decisões relevantes para a elaboração do programa, as escolhas dos parâmetros ERROABS e ERROREL, bem como a análise dos resultados dos testes abaixo.

3.1 Queda de um corpo sob a ação da resistência do ar

A equação diferencial para a velocidade de um corpo de massa M em queda sob a ação de uma força de resistência do ar proporcional à velocidade é

$$M\dot{v}(t) - Mg + kv(t) = 0.$$

A solução com velocidade inicial $v(0) = v_0$ é dada por (verifique!)

$$v(t) = \frac{Mg - e^{-kt/M}(Mg - v_0 k)}{k}.$$

Supondo as constantes $g = 10$ e $M = 1$, determine o valor de k sabendo-se que com $v_0 = 3$ temos $v(2) = 20$.

3.2 Altura de fios de transmissão de eletricidade

Usando princípios da Física pode-se mostrar que quando um cabo flexível for pendurado entre dois postes, ele tomará a forma de uma curva $y = f(x)$ que satisfaz a equação diferencial

$$\frac{d^2y}{dx^2} = \frac{\rho g}{T}\sqrt{1 + \left(\frac{dy}{dx}\right)^2}$$

onde ρ é a densidade linear do cabo, g é a aceleração da gravidade e T é a tensão do cabo em sua parte mais baixa. Uma solução para a equação acima é dada por (verifique!)

$$f(x) = \frac{T}{\rho g} \cosh\left(\frac{\rho g}{T} x\right).$$

Um cabo pendurado sempre toma a forma de uma *catenária*

$$y = f(x) = \alpha + \beta \cosh\left(\frac{x}{\beta}\right)$$

onde α e β são constantes. Suponha que tenhamos um cabo pendurado entre $x = -10$ e $x = 10$. Determine o valor de β tal que a diferença de altura $f(10) - f(0)$ seja igual a 0.5.

3.3 Interseção de curvas em coordenadas polares

Queremos saber quais os pontos de interseção duas a duas das curvas dadas em coordenadas polares abaixo. Para todas as curvas considere $0 \leq \theta \leq 2\pi$. É conveniente gerar figuras.

1. Curva da borboleta: $r = e^{\sin\theta} - 2\cos(4\theta)$.
2. Cardióide: $r = 1 - \sin\theta$.
3. $r = \sin^2(4\theta) + \cos(4\theta)$.
4. $r = 1 + 2\sin(\theta/2)$.

Observações:

- Veja que pedimos 6 conjuntos de interseções.
- Na curva da borboleta podemos ter $r < 0$. Assim, convém lembrar que um ponto com coordenadas polares (r, θ) é o mesmo que com coordenadas $(-r, \theta + \pi)$.
- Como o método descrito necessita de um intervalo onde a função troca de sinal, é possível que algumas interseções escapem.
- Um ponto de partida para aprocura automática dos zeros seria particionar o intervalo $[0, 2\pi]$ em, digamos, 100 subintervalos e depois testar cada um deles.