Universidade Federal do Espírito Santo

Departamento de Computação e Eletrônica - CEUNES

Programação Funcional

Prof. Oberlan Romão

Exercício Programa 2

Raízes de equações

O problema de calcular as raízes reais de uma equação sempre foi objeto de estudo da matemática ao longo dos séculos. Encontrar essas raízes faz parte de muitos problemas importantes das áreas relacionadas a matemática.

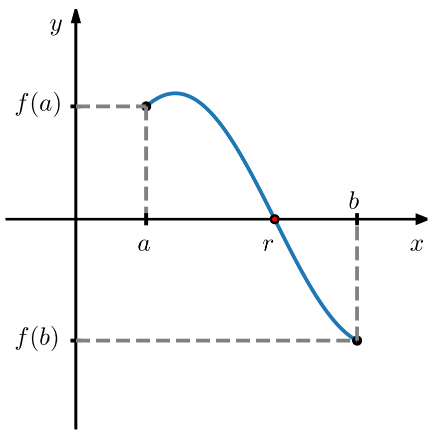
Matematicamente falando, dado uma função f, o objetivo é encontrar o(s) valor(es) de r, tal que $f(r) = 0$, onde r é chamado de raiz da equação ou zero da função f. Em geral, f pode ser qualquer função. Para algumas formas de f, existem soluções analíticas, por exemplo, se f for um polinômio de grau 1 ou 2. No entanto, para outras funções, temos que usar métodos numéricos, ou algoritmos, para encontrar uma solução exata ou aproximada para $f(r) = 0$.

Outro problema típico de encontrar raízes é quando precisamos encontrar os pontos de intersecção entre duas curvas. Seja $y = f(x)$ e $y = g(x)$ duas funções contínuas. Desejamos encontrar um ponto de intersecção dessas duas curvas, ou seja, precisamos encontrar um r tal que $f(r) = g(r)$ ou $f(r) - g(r) = 0$. Portanto, um ponto de intersecção das curvas $y = f(x)$ e $y = g(x)$ é um zero da função $h(x) = f(x) - g(x)$.

Dada uma equação, uma pergunta natural é se essa equação possui ou não alguma raiz. O *Teorema de Bolzano* nos fornece condições suficientes para garantir a existência do zero de uma função. Esse é uma aplicação direta do *Teorema do Valor Intermediário*[1].

Teorema: (*Teorema de Bolzano*) Seja $f(x)$ uma função continua no intervalo fechado $[a, b]$. Se $f(a)f(b) < 0$, então existe um $r \in (a, b)$, tal que $f(r) = 0$.

Em outras palavras, se $f(x)$ é uma função contínua em um dado intervalo no qual ela troca de sinal, então ela tem pelo menos um zero nesse intervalo (veja a figura abaixo)



[1] https://pt.wikipedia.org/wiki/Teorema_do_valor_intermedi%C3%A1rio

Como dito, em muitas equações não é possível encontrar suas raízes reais usando soluções analíticas, sendo necessário algum método numérico. Nesse EP, seu objetivo é implementar e analisar/comparar **três** métodos clássicos para encontrar raízes reais (aproximadas) de equações.

Método da Bisseção

O método da bisseção explora o fato de que uma função contínua f no intervalo fechado $[a, b]$ com $f(a)f(b) < 0$ tem uma raiz no intervalo (a, b) (Teorema de Bolzano). Assim, a ideia desse método para aproximar o zero de uma função $f(x)$ é tomar, como primeira aproximação, o ponto médio do intervalo $[a, b]$, isto é:

$$x_1 = \frac{a+b}{2}$$

Existem duas possibilidades:

- $f(x_1) = 0$. Nesse caso, o zero de $f(x)$ é $r = x_1$
- $f(x_1) \neq 0$. Nesse caso, devemos ter ou $f(a)f(x_1) < 0$ ou $f(x_1)f(b) < 0$.

No segundo caso, se $f(a)f(x_1) < 0$, então $r \in (a, x_1)$. Nesse caso, tomamos como segunda aproximação da raiz de $f(x)$ o ponto médio do intervalo $[a, x_1]$, isto é, $x_1 = \dfrac{a + x_1}{2}$. No outro caso, temos $f(x_1)f(b) < 0$ e, então, tomamos $x_1 = \dfrac{x_1 + b}{2}$. Se repetirmos esse procedimento várias vezes teremos uma sequência de aproximações: $x_1, x_2, x_3, \ldots$, que converge para um zero da função $f(x)$, isso é, $\lim\limits_{n\to\infty} x_n = r$.

O erro relativo do método é dado por: $\left|\dfrac{a-b}{2}\right|$.

Pseudocódigo do Método

Para calcular a raiz aproximada de uma função usando o Método da Bisseção com precisão ϵ, devemos:

(1) Inicializar $i = 1$ e selecionar a e b tal que a função mude de sinal, isto é, $f(a)f(b) < 0$ (isso pode ser feito analisando o gráfico da função).

(2) Estimar a raiz como x_i dada por $x_i = \dfrac{a+b}{2}$;

(3) Verificar o critério de parada, isto é, se $f(x_i) = 0$ ou se $\left|\dfrac{a-b}{2}\right| < \epsilon$ ou se o número máximo de iterações for atingido, então x_i é a aproximação desejada e a retornamos;

(4) Se o critério de parada não for satisfeito, determinar o próximo intervalo:

- Se $f(a)f(x_i) < 0$, então a raiz está no intervalo (a, x_i), faça $b = x_i$ e volte ao Passo (2);
- Se $f(b)f(x_i) < 0$, então a raiz está no intervalo (x_i, b), faça $a = x_i$ e volte ao Passo (2).

Exemplo: Encontrar o valor aproximado de $\sqrt{2}$ usando o Método da Bisseção.

Queremos encontrar x tal que $x = \sqrt{2}$, ou seja, $x^2 - 2 = 0$, o que é equivalente a localizar a raiz positiva de $f(x) = x^2 - 2$.

Primeiro, devemos selecionar dois números a e b tal que $f(a)f(b) < 0$. Sabemos que o valor de $\sqrt{2} \in [1, 2]$, ou seja, $a = 1$ e $b = 2$ satisfaz o critério de $f(a)f(b) < 0$.

Na primeira iteração ($i = 1$), temos $x_1 = \frac{2+1}{2} = 1.5$. O valor da função em x_1 é $f(x_1) = (1.5)^2 - 2 = 0.25$. Como $f(x_1)$ é positivo, $b = 2$ é substituído por $b = 1.5$ para garantir que na próxima iteração $f(a)$ e $f(b)$ tenha sinais opostos. À medida que isso continua, o intervalo entre a e b se tornará cada vez menor, convergindo para a raiz da função. A tabela abaixo mostra o resultado das 15 primeiras iterações.

i	a	b	x_i	$\left\|\frac{a-b}{2}\right\|$
1	1.000000000000000	2.000000000000000	1.500000000000000	0.500000000000000
2	1.000000000000000	1.500000000000000	1.250000000000000	0.250000000000000
3	1.250000000000000	1.500000000000000	1.375000000000000	0.125000000000000
4	1.375000000000000	1.500000000000000	1.437500000000000	0.062500000000000
5	1.375000000000000	1.437500000000000	1.406250000000000	0.031250000000000
6	1.406250000000000	1.437500000000000	1.421875000000000	0.015625000000000
7	1.406250000000000	1.421875000000000	1.414062500000000	0.007812500000000
8	1.414062500000000	1.421875000000000	1.417968750000000	0.003906250000000
9	1.414062500000000	1.417968750000000	1.416015625000000	0.001953125000000
10	1.414062500000000	1.416015625000000	1.415039062500000	0.000976562500000
11	1.414062500000000	1.415039062500000	1.414550781250000	0.000488281250000
12	1.414062500000000	1.414550781250000	1.414306640625000	0.000244140625000
13	1.414062500000000	1.414306640625000	1.414184570312500	0.000122070312500
14	1.414184570312500	1.414306640625000	1.414245605468750	0.000061035156250
15	1.414184570312500	1.414245605468750	1.414215087890625	0.000030517578125

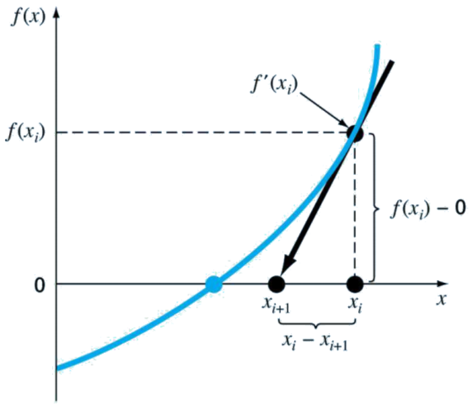
Tomando o último x_i (ou seja, x_{15}) como solução de $f(x)$, temos que o valor aproximado de $\sqrt{2} \approx 1.414215087890625$.

Método de Newton-Raphson

O método de Newton-Raphson usa a inclinação (tangente) da função $f(x)$ na solução iterativa atual (x_i) para encontrar a solução (x_{i+1}) na próxima iteração. Analisando a figura abaixo, notamos que a inclinação em $(x_i, f(x_i))$ é dada por:

$$f'(x) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

.



Isolando x_{i+1} na equação anterior temos:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x)},$$

sendo x_1 uma aproximação inicial dada. Perceba que devemos ter $f'(x) \neq 0$ para que o método funcione.

O erro relativo do método é dado por: $\left|\dfrac{x_{i+1} - x_i}{x_{i+1}}\right|$.

Exemplo: Encontrar o valor aproximado de $\sqrt{2}$ usando o Método de Newton-Raphson.

Como já sabemos, queremos encontrar a raiz positiva da função $f(x) = x^2 - 2$. A derivada de $f(x)$ é dada por $f'(x) = 2x$, assim: $x_{i+1} = x_i - \dfrac{x^2 - 2}{2x}$. Considerando $x_1 = 2$ e $\epsilon = 10^{-15}$, temos a seguinte solução:

i	x_i	x_{i+1}	$\left\|\dfrac{x_{i+1} - x_i}{x_{i+1}}\right\|$
1	2.000000000000000	1.500000000000000	0.333333333333333
2	1.500000000000000	1.416666666666667	0.058823529411765
3	1.416666666666667	1.414215686274510	0.001733102253033
4	1.414215686274510	1.414213562374690	0.000001501823965
5	1.414213562374690	1.414213562373095	0.000000000001128
6	1.414213562373095	1.414213562373095	0.000000000000000

Tomando x_7 como solução de $f(x)$, temos que o valor aproximado de $\sqrt{2} \approx 1.414213562373095$.

Método da Secante

Para implementar o método de Newton-Raphson, $f'(x)$ precisa ser encontrado analiticamente e avaliado numericamente. Em alguns casos, a avaliação analítica (ou numérica) pode não ser viável ou desejável.

O método da Secante é uma variação do método de Newton-Raphson, evitando a necessidade de conhecer-se a derivada analítica de $f(x)$. Dada uma função $f(x)$, a ideia é aproximar sua derivada pela razão fundamental:

$$f'(x) \approx \frac{f(x_{i-1}) - f(x_i)}{x_{i-1} - x_i}. \tag{1}$$

Então, o método de Newton-Raphson pode ser modificado por:

$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)}{f'(x_i)} \\ &= x_i - \frac{f(x_i)}{\frac{f(x_{i-1})-f(x_i)}{x_{i-1}-x_i}} \\ &= x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \end{aligned}$$

Observe que para inicializarmos a iteração acima precisamos de duas aproximações iniciais, a saber, x_1 e x_2.

O erro relativo do método é dado por: $\left|\dfrac{x_{i+1} - x_i}{x_{i+1}}\right|$.

Tarefas

Sua tarefa nesse EP é escolher três funções $f(x)$ (**não escolha um polinômio**), calcular o valor aproximado de **uma** das raízes reais de cada função usando os três métodos apresentados e fazer um relatório técnico com uma análise e discussão dos resultados obtidos. Após o programa ser feito, você deve preencher o arquivo Jupyter Nootebook (EP2.ipynb).

Seu programa deve ter, obrigatoriamente, o seguinte:

- Uma função que define a função escolhida por você (pode ser `lambda`);
- Uma função que define a derivada da função escolhida (pode ser `lambda`);
- Uma função para calcular o valor aproximado da raiz usando o Método da Bisseção. Essa função deve retornar, pelo menos, uma lista com os valores aproximados da raiz da função.
- Uma função para calcular o valor aproximado da raiz usando o Método de Newton-Raphson. Ela deve retornar, pelo menos, uma lista com os valores aproximados da raiz da função.
- Uma função para calcular o valor aproximado da raiz usando o método da Secante. Ela deve retornar, ao menos, uma lista com os valores aproximados da raiz da função.
- `main()`: função que chama as funções anteriores para calcular o valor aproximado da raiz da função e gerar o(s) gráfico(s) dos resultados. Se necessário, esse função também pode ser usada para imprimir os resultados obtidos por cada método.

Se necessário, seu código pode conter outras funções auxiliares.

Para cada método, considere o número máximo de iterações como sendo 50 e a tolerância (ϵ) como sendo 10^{-15}. Esse valores devem ser parâmetros padrões (*default*) das funções. **Sempre que for imprimir valores do tipo ponto flutuante, imprima com, pelo menos, 15 casas decimais.**

No relatório técnico enviado, você deve apresentar as três funções $f(x)$ escolhidas, mas no arquivo .py e no Jupyter Notebook você pode deixar apenas a que achar mais interessante.

O que entregar

Você deve entregar, **via AVA**, o código fonte (arquivo .py), um relatório (formato pdf) e o arquivo Jupyter Nootebook (EP2.ipynb) preenchido.

Data de entrega: até às 6h do dia 01/12/2018.

Observações:

1. Não é permitido usar **estruturas de repetição (loop)**, como `while`, `for` e **funções impuras**. A utilização dessas estruturas/funções implicará em nota 0.
2. Não use variáveis globais para evitar a possibilidade de uma função se tornar impura.
3. Use apenas instruções/comandos visto em sala de aula (teórica ou prática). Se quiser usar algo que ainda não foi visto, me pergunte;
4. Documente o seu programa: comente e use `"""docstring"""` nas funções. Código sem comentários e sem `"""docstring"""` valerá, no máximo, 9,0 pontos;
5. Em caso de plágio, será atribuído 0 a todos os envolvidos.

Critérios de Avaliação

A nota do EP se dará pela seguinte fórmula:

$$R \times S \times NB \times (N_{EP}),$$

onde,

- $R = \begin{cases} 1, & \text{se enviou o relatório técnico;} \\ 0, & \text{caso contrário (ou se enviou um arquivo em branco).} \end{cases}$
- $S = \begin{cases} 1, & \text{se enviou o } \textit{script} \text{ com as implementações solicitadas;} \\ 0, & \text{caso contrário (ou se enviou um arquivo em branco).} \end{cases}$
- $NB = \begin{cases} 1, & \text{se enviou o Jupyter Notebook;} \\ 0, & \text{caso contrário (ou se enviou um arquivo inalterado).} \end{cases}$
- N_{EP}: Nota geral do EP, sendo $0.0 \leq N_{EP} \leq 10.0$, detalhado abaixo.

A nota geral do EP (N_{EP}) será avaliada da seguinte forma:

- Relatório técnico: 4.0 pontos (caprichem). O relatório deve conter, pelo menos, os seguintes pontos:
  - Apresentação das funções escolhidas por você juntamente com o gráfico da função (usando o `Matplotlib`);
  - Tabelas com os valores gerados pelo seu programa;
  - Análise e discussão dos resultados obtidos pelos três métodos, mostrando vantagens e desvantagens de cada método. Você pode comprar, por exemplo, qual dos métodos se aproxima mais rápido para o valor das raízes, qual método é mais simples e/ou eficiente, etc.;
  - Gráficos (feitos com o `Matplotlib`) comparando os resultados.
- Jupyter Nootebook: 1.0 ponto;
- Implementação dos métodos: 5.0 pontos;
  - Método Bisseção: 1.5 pontos;
  - Método de Newton-Raphson: 1.5 pontos;
  - Método da Secante: 1.5 pontos;
  - Função `main()`: 0.5 ponto;

Perceba que se não for enviado um dos arquivos (relatório técnico ou arquivo .py ou o Jupyter Notebook), sua nota será 0.

Exemplo de saída

Para simplificar, o número de iterações foi definido como 8.

```
Função escolhida: f(x) = x**2 - 2
Derivada: f'(x) = 2*x

Número máximo de iterações: 8
Tolerância (epsilon): 10**-15

==> Método da Bisseção: a = 1 e b = 2
-------------------------------------------
i           raiz               erro
-------------------------------------------
1    1.500000000000000    0.500000000000000
2    1.250000000000000    0.250000000000000
3    1.375000000000000    0.125000000000000
4    1.437500000000000    0.062500000000000
5    1.406250000000000    0.031250000000000
6    1.421875000000000    0.015625000000000
7    1.414062500000000    0.007812500000000
8    1.417968750000000    0.003906250000000
-------------------------------------------
==> Método de Newton-Raphson: x1 = 2
-------------------------------------------
i           raiz               erro
-------------------------------------------
1    1.500000000000000    0.333333333333333
2    1.416666666666667    0.058823529411765
3    1.414215686274510    0.001733102253033
4    1.414213562374690    0.000001501823965
5    1.414213562373095    0.000000000001128
6    1.414213562373095    0.000000000000000
-------------------------------------------
==> Método da Secante: x1 = 1 e x2 = 2
-------------------------------------------
i           raiz               erro
-------------------------------------------
1    1.333333333333333    0.500000000000000
2    1.400000000000000    0.047619047619048
3    1.414634146341463    0.010344827586207
4    1.414211438474870    0.000298900047824
5    1.414213562057320    0.000001501599551
6    1.414213562373095    0.000000000223287
7    1.414213562373095    0.000000000000000
-------------------------------------------
```

Obs.: No relatório você pode apresentar outras saídas e/ou tabelas para que o mesmo fique mais completo.