INF1608 – Análise Numérica

Lab 2: Raízes de Função

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

1. Neste laboratório, pede-se para implementar métodos abertos para determinação de raízes. É importante que a implementação **minimize o número de avaliações de** $f(x)$. Para cada método, a precisão da solução será avaliada medindo o erro relativo entre estimativas consecutivas, isto é:

$$e_i = \frac{|x_i - x_{i-1}|}{|x_i|}$$

Como trata-se de métodos abertos, é necessário verificar a convergência dos mesmos. Para este laboratório, um método será considerado não convergente se o número de iterações for superior a 50.

   (a) O método da *secante* para determinação de raízes da função $f(x)$ recebe como entrada duas estimativas iniciais: x_0 e x_1. A próxima estimativa é dada por:

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

se $|f(x_1) - f(x_0)| < 10^{-15}$, faça:

$$x_2 = \frac{x_0 + x_1}{2}$$

Se o erro relativo não estiver abaixo da precisão desejada, repete-se o processo considerando como estimativas iniciais os valores de x_1 e x_2, e assim sucessivamente. Implemente uma função para determinar a raiz usando o método da secante, onde o erro relativo avaliado tenha precisão de 8 dígitos, isto é, $e < 0.5 \times 10^{-8}$. Sua função deve receber como parâmetros as estimativas iniciais, a função $f(x)$ cuja raiz deseja-se calcular e o endereço da variável que armazenará a raiz calculada. Sua função deve retornar o número de iterações usado na determinação da raiz. Se não houver convergência, a função deve retornar zero.

```
int secante (double x0, double x1, double (*f) (double x), double* r);
```

   (b) O método da *interpolação quadrática inversa* (IQI) para determinação de raízes da função $f(x)$ considera três estimativas iniciais x_0, x_1 e x_2 da raiz. A partir dessas três estimativas, o método ajusta uma parábola inversa $x(y) = ay^2 + by + c$, onde $y_i = f(x_i)$, adotando como próxima estimativa a interseção desta parábola com o eixo x, isto é, o valor do coeficiente c: $x_3 = c$. Se o erro relativo não estiver abaixo da precisão desejada, repete-se o processo considerando como estimativas iniciais os valores de x_1, x_2 e x_3, e assim sucessivamente.

Implemente uma função que calcule a raiz de uma função segundo o método IQI, de forma similar ao item anterior, com o seguinte protótipo:

```
int IQI (double x0, double x1, double x2, double (*f) (double x), double* r);
```

Para calcular o coeficiente c da parábola, sugere-se usar a Regra de Crammer:

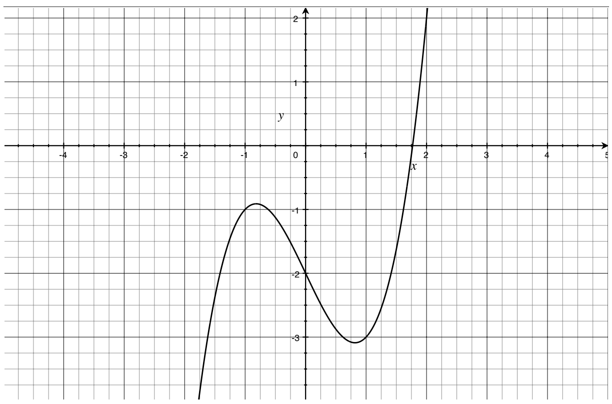
$$c = \frac{\det A_c}{\det A}$$

onde:

$$A = \begin{bmatrix} f(x_0)^2 & f(x_0) & 1 \\ f(x_1)^2 & f(x_1) & 1 \\ f(x_2)^2 & f(x_2) & 1 \end{bmatrix} \qquad A_c = \begin{bmatrix} f(x_0)^2 & f(x_0) & x_0 \\ f(x_1)^2 & f(x_1) & x_1 \\ f(x_2)^2 & f(x_2) & x_2 \end{bmatrix}$$

2. Teste suas implementações, analisando os valores de raízes encontrados e o número de iterações necessárias:

   (a) Compare os dois métodos para encontrar a raiz da função $f(x) = x^3 + x - 7$, com diferentes estimativas inicias.

   (b) Para verificar o critério de não convergência, teste os dois métodos para encontrar as raízes da função $f(x) = x^4 + x + 1$.

   (c) Compare os dois métodos na resolução do seguinte problema: a velocidade de um paraquedista em queda livre pode ser dada por:

   $$v = \frac{gm}{c}\left(1 - e^{-\frac{c}{m}t}\right)$$

   onde $g = 9.8m/s^2$. Para um paraquedista com um coeficiente de arrasto $c = 15Kg/s$, calcule a massa m para que a velocidade seja $v = 35m/s$ em $t = 9s$.

Organize seu código da seguinte forma. O arquivo "raiz.c" deve conter as implementações das função `secante` e `IQI`, com seus respectivos protótipos no arquivo "raiz.h". O arquivo "main.c" deve conter os testes realizados.

Entrega: O código fonte deste trabalho (isto é, os arquivos "raiz.h", "raiz.c" e "main.c") devem ser enviados via página da disciplina no EAD. O prazo final para envio é **sexta-feira, dia 27 de agosto**.