INF1608 – Análise Numérica

Lab 2: Raízes de Função

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

1. Implemente os seguintes métodos para determinação de raízes:

   (a) O método da bisseção para determinação de raízes da função $f(x)$ recebe como entrada o intervalo de busca $[a, b]$, assumindo $f(a).f(b) < 0$. O erro na avaliação da raiz é dado por $e = \frac{b-a}{2^{n+1}}$, onde n representa o número de iterações. Implemente uma função para determinar a raiz usando o método da bisseção, onde o erro avaliado tenha precisão de p dígitos, isto é, $e < 0.5 \times 10^{-p}$. Sua função também deve receber como parâmetro 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, seguindo protótipo:

   ```
   int bissecao (double a, double b, int p, double (*f) (double x), double* r);
   ```

   Sua implementação deve minimizar o número de avaliações da função $f(x)$.

   (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: $x_{i+1} = c$.
   Implemente uma função que calcule a raiz de uma função segundo o método IQI com o seguinte protótipo:

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

   onde $\mathtt{x}_i$ representam as estimativas iniciais da raiz, `f` a função e `p` a precisão em número de dígitos. A função deve preencher o valor da raiz encontrada no endereço `r` passado e deve ter como valor de retorno o número de iterações usadas para alcançar o resultado. Se não houver convergência, a função deve retornar zero. A precisão do resultado deve ser verificada pelo erro avaliado na entrada (*backward error*): $|f(x_i)| < 0.5 \times 10^{-p}$.
   Para calcular o coeficiente c da parábola, sugere-se usar a Regra de Crammer:

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

   onde:

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

Sua implementação deve minimizar o número de avaliações da função $f(x)$.

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

   (a) Compare os dois métodos para encontrar a raiz positiva da função $f(x) = \cos x - x^3 + x$, com 6 dígitos de precisão.

   (b) 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 `bissecao` 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 **quinta-feira, dia 30 de agosto**.