MS 211 - ATIVIDADES COMPUTACIONAIS

1º sem/2014

1. Precisão da Máquina

   (a) A precisão da máquina também pode ser definida como sendo o menor número positivo ε em aritmética de ponto flutuante tal que : $(1 + \varepsilon) > 1$. O algoritmo abaixo estima a precisão da máquina :

   Passo 1 : A $= 1$
   $s = 1 + A$
   $k = 0$
   Passo 2 : Enquanto $s > 1$, faça :
   $A = A/2$
   $s = 1 + A$
   $k = k + 1$
   Passo 3 : Faça $Prec = A * 2$ e imprimir $Prec$.

   i. Teste este algoritmo usando o MatLab ou uma linguagem de sua escolha. Implima o resultado ao se trabalhar em precisão simples[1] e em precisão dupla.

   ii. Interprete o passo 3 do algoritmo, isto é, por que a aproximação para *Prec* é escolhida como sendo o dobro do último valor de A obtido no passo 2?

   iii. O valor final de k representa o número de iterações necessárias até que a condição do `while` se verifique. Qual foi este número quando o programa foi executado em precisão simples? E em precisão dupla? Este número está relacionado com o resultado *Prec* e com o número de dígitos na mantissa de cada representação? Explique (escreva uma expressão que relaciona, ao menos de maneira aproximada, essas quantidades).

   (b) Na definição da precisão da máquina, usamos como referência o número 1. No algoritmo abaixo a variável ω é um dado de entrada, escolhido pelo usuário:

   Passo 1 : A $= 1$
   $s = \omega + A$;
   Passo 2 : Enquanto $s > \omega$, faça :
   $A = A/2$
   $s = \omega + A$
   Passo 3 : Faça $Prec = A * 2$ e imprimir $Prec$.

[1] O MatLab trabalha sempre em precisão dupla. Uma forma de se trabalhar em precisão simples é declarar as variáveis como `single`. Exemplo:

```
A = single(1);
s = single(1 + A);
k = 0;
while (s > 1)
      A = A/2;
      s = 1+A;
      k = k+1;
end
prec = A*2;
```

i. Teste seu programa atribuindo para ω valores distintos. Utilize as potências p de 2, $p = -3, -2, -1, 0, 1, 2, 3$. Para cada valor de ω imprima o valor correspondente obtido para *Prec* e o número k de iterações realizadas. (trabalhe em precisão dupla)

ii. Justifique por que *Prec* (e consequentemente o valor de k) se altera quando ω é modificado, e com base nisso explique porque a precisão da máquina é definida com base em $\omega = 1$.

2. Função Inversa

Dada uma função $f(x)$, o problema de se encontrar o valor de x (caso ele exista) tal que $f(x) = c$ para um dado valor c é o problema de se encontrar o valor de sua inversa $f^{-1}(c)$ calculada no ponto c, pois $x = f^{-1}(c)$. Assim, o valor de x pode ser obtido numericamente encontrando-se a raiz da equação $f(x) - c = 0$.

(a) Encontre o(s) valor(es) de x (caso ele(s) exista(m)) tal(is) que $f(x) = c$ para cada valor de c abaixo

$$c \in \{0.1,\ 0.14,\ 0.2 \text{ e } -0.27\}$$

usando[2] $f(x) = xe^{-x}\ln(x)$. Justifique por meio de gráficos o número de zeros em cada caso, e a sua escolha de aproximação inicial x_0 para cada um deles. Aplique o método de Newton e obtenha os resultados com uma precisão de $\varepsilon = 10^{-15}$ em x ($|x_k - x_{k-1}| < \varepsilon$).

(b) Repita o exercício anterior para $c = -0.17$ e $c = -0.27$ usando $x_0 = 0.1$ (apenas para a menor raiz de cada), mas agora imprima as precisões $\varepsilon_k = |x_k - x_{k-1}|$ a cada iteração k e faça um gráfico de $\log_{10}\varepsilon_k \times k$. Observe o que ocorre com a convergência em cada situação e *justifique* a diferença observada entre elas. Faça ainda um gráfico de $\log_2(-\log_2\varepsilon_k) \times k$ e calcule *graficamente* (apenas de maneira aproximada, da maneira que conseguir) a ordem de convergência[3] em cada caso. Compare com a ordem de convergência assintótica $p = 2$ do método de Newton.

(c) Repita novamente o exercício para $c = -0.27$ mas agora usando $x_0 = 0.25$ e obtenha a raiz com uma precisão de $\varepsilon = 10^{-16}$. Explique o que ocorre ao tentar se obter uma precisão maior do que essa.

[2] No MATLAB usar a função `realllog`(x) ao invés de `log`(x) ao aplicar o logarítmo natural. Isso porque a função `log(x) é definida no domínio dos números complexos, e se aplica também a números negativos. Isso pode gerar falsas soluções quando trabalhamos no domínio dos números reais.`

[3] A ordem de convergência é definida como sendo o número p tal que $e_k \approx (e_{k-1})^p$. Dessa forma, o erro segue aproximadamente uma regra do tipo $e_k \approx A^{p^k}$, com A constante, o que implica que $\log_2(e_k) \approx \log_2(A).p^k$. Como $\log_2(A)$ é um número negativo para valores de A menores do que 1 ($A \approx e_0$ é dado pelo erro inicial), então tomamos o negativo antes de aplicar o logarítmo novamente. Assim, $\log_2(-\log_2 e_k) \approx \log_2(-\log_2(A)) + k.\log_2(p)$. Portanto, o coeficiente angular do gráfico $\log_2(-\log_2 \varepsilon_k) \times k$ é dado por $\log_2(p)$.