MS 211 - LISTA DE EXERCICIOS 1
ARITMÉTICA DE PONTO FLUTUANTE – ERROS EM OPERAÇÕES NUMÉRICAS

2º sem/2012

1. Considere uma máquina com sistema de representação de números definido por: base 10 ($\beta = 10$); 5 dígitos na mantissa sendo que um deles é usado para armazenar informação do sinal; expoente no intervalo: $[-5; 5]$. Pede–se:
   a) qual o menor e o maior número em módulo representado nesta máquina?
   b) como será representado o número 23456 nesta máquina se for usado o arredondamento? E se for usado o truncamento?
   c) Se $a = 42450$ e $b = 3$ qual o resultado de $a + b$ se for usado o arredondamento? E se for usado o truncamento? Justifique o resultado.
   d) Considerando ainda, $a = 42450$ e $b = 3$, qual o resultado da operação: $a + \sum_{i=1}^{10} b$, considerando que está sendo realizado o truncamento?
   e) Repetir o item *(d)*, para a operação: $\sum_{i=1}^{10} b + a$.
   f) Considere o cálculo de $z = \frac{a*b}{c}$ onde $a = 4 * 10^4$, $b = 3 * 10^2$ e $c = 2 * 10^3$. Qual o resultado obtido no cálculo de z nas duas opções:
   i) calcula $m = a * b$ e em seguida obtém $z = m/c$;
   ii) calcula $m = a/c$ e em seguida $z = m * b$. Justifique.

2. Avalie as expressões abaixo em um sistema de ponto flutuante com cinco dígitos significativos. Compute o erro relativo em cada operação executada e no resultado final. Identifique qual a operação contribuiu mais expressivamente para o erro final.
   a) $72126 + 24.821 - 72160$; *b)* $\sqrt{1 + 1.4 \cdot 10^{-4}} - 1$;
   c) $10000 + \sum_{n=1}^{25000} 0.4$; *d)* $\left(\sum_{n=1}^{25000} 0.4\right) + 10000$.

3. Reorganize as expressões abaixo para amenizar possíveis erros de cáculo. Escolha valores para x que evidenciem os erros.
   a) $\sqrt{x^2 + 1} - x$; *b)* $\ln(x + 1) - \ln(x)$;
   c) $\sqrt{1 + x} - 1$; *d)* $(1 - \cos x)/\sin(x)$.

4. Considere a sequência de operações, onde `a = 4/3`:
   `b = a - 1;  c = 3*b;  f = 1 - c`
   O resultado exato é `f = 0`. Porém, executando esta sequência de cálculos através dos comandos do MatLab:
   (em precisão simples)
   ```
   format long
   sa = single(4/3);  sb = single(sa - 1);  sc = 3*sb;  sf = 1 - sc;
   ```
   (em precisão dupla)
   ```
   format long
   da = 4/3;  db = da - 1;  dc = 3*db;  df = 1 - dc;
   ```
   Obtemos: `sf = -1.1920929e-007` e `df = 2.220446049250313e-016`.
   Justifique estes resultados.

5. O objetivo é analisar o resultado de $\sum_1^n \rho$, para um número grande de parcelas n e valores para ρ iguais a 0.11 e 0.5. Execute este programa no MatLab, com 1000 parcelas. Interprete e justifique

os resultados.

```
format long e
cte = input('entre com valor para a constate--> ');
np = 1000;
soma = 0;
for i = 1:np
   soma = soma + cte;
end
soma
valexato = cte*np
erro = valexato - soma
```

6. O objetivo é calcular o somatório que envolve a adição de cem mil parcelas iguais a 2^{-53} ao número 1. Se $(r1 = 1 + \sum_1^{100000} 2^{-53})$ e $(r2 = \sum_1^{100000} 2^{-53} + 1)$, teoricamente devemos ter $r1 = r2$. Execute este programa no MatLab e justifique os resultados obtidos comparando os valores $r1$ e $r2$:

```
r1 = 1;
r2 = 0;
delta = 2^-53;
for j = 1:100000
   r1 = r1 + delta;
   r2 = r2 + delta;
end
 r2 = 1 + r2;
disp('valor obtido para r1 = ', r1)
disp('valor obtido para r2 = ', r2)
```

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

   Passo 1 : A $= 1$
   $s = 1 + A$
   $k = 1$

   Passo 2 : Enquanto $s > 1$, faça :
   $A = A/2$
   $s = 1 + A$
   $k = k + 1$

   Passo 3 : Faça $Prec = A * 2$ e imprimir $Prec$.

   a) Teste este algoritmo usando o MatLab ou uma linguagem de sua escolha. Trabalhe em precisão simples e em precisão dupla. O MatLab trabalha sempre em precisão dupla. Uma forma de trabalhar em precisão simples é declarar as variáveis como `single`. Exemplo:

```
A = single(1);
s = single(1 + A);
k = 1;
```

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

Compare os valores obtidos com o valor obtido ao se dar o comando `eps` do MatLab.

b) 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?

c) 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 número de dígitos na mantissa em cada representação? Justifique.

d) Na definição de 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 $Prec$.

d.1) Teste seu programa atribuindo para ω valores distintos. Por exemplo, potências p de 10, $p = \ldots, -4, -3, -2, -1, 0, 1, 2, 3, \ldots$. Para cada valor de ω execute seu programa em precisão simples e dupla.

d.2) Para cada valor de ω imprima o valor correspondente obtido para *Prec* e o número k de iterações realizadas. Justifique por que *Prec* se altera quando ω é modificado. E o número de iterações k se altera ou é aproximadamente o mesmo em todos os testes? Justifique estes resultados.

8. Cálculo de exp(x): O objetivo é calcular $exp(x)$ pela fórmula de Taylor em torno de *zero*:
$\exp(x) \simeq 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots\ldots$

a) Escreva um programa para o cálculo de $\exp(x)$ calculando cada termo da série através de: $termo = x^j/fj$ onde $fj = j!$ (fatorial de j) calculado da forma: $fj = 1*2*3*\ldots\ldots*j$. Observe a ocorrência de *overflow*!

b) O procedimento abaixo, obtém uma aproximação para exp(1):

```
termo = 1;   x = 1;   s = 1;   k = 1;
      while .........
        termo =termo*(x/k);
        s = s + termo;
        k = k + 1;
      end
```

Realizando os cálculos desta forma, evita–se o *overflow* no cálculo do fatorial e a série pode ser calculada com tantos termos quanto se queira. Qual seria um critério de parada para se interromper o cálculo da série?

c) Teste este algoritmo (no MatLab) com vários valores para x: positivos, negativos, ($x \approx 0$ e x distante de zero) e, para cada valor de x, imprima o número de iterações k para que o critério

de parada seja verificado. Analise os resultados obtidos.
