MS211, Cálculo Numérico, Turma E
Primeiro Semestre de 2020, UNICAMP

Resoluçao Lista de Exercícios 1

Entrega por email (roman@ime.unicamp.br) dos três exercícios marcados com (*) até Quarta Feira 25/03/2020.
Os exercícios (*) podem ser desenvolvidos em grupos de até três membros.

Note que, nestes exercícios usamos a notação FP(β, p, s) para a aritmética finita em base β com p dígitos na mantissa e s no expoente.

(1) No problema de medir a altura dos alunos, quais erros e de que tipos podem afetar o cálculo da media através o uso de um computador.

Resposta:

- Erro de modelagem da media (formula errada, numero alunos errado)
- Erro sistemático (medida errada devida ao instrumento (metro) usado)
- Erro random (respostas erradas dos alunos, erro no uso do metro)
- Erro numérico de representação na aritmética finita dos valores das alturas, e erro das operações soma e divisão na aritmética finita

(2) **(*) Peso** 30%
Para computar o erro numa medida é melhor usar o erro relativo ou o erro absoluto? Quando os dois erros tem o mesmo comportamento? Motive a sua resposta.

Resposta: O erro relativo differe do erro absoluto para a divisão respeito o valor aproximado $\bar{x}$ (ou respeito o valor para aproximar) $E_R = \frac{E_A}{\bar{x}} = \frac{|x-\bar{x}|}{|\bar{x}|}$. Este permite de medir o erro como proporção respeito o valor approximado (ou o valor par aproximar). Por isso é preferivel usar sempre o erro relativo, porem o erro absoluto pode ser utilizado, e é na verdade muito utilizado. Por exemplo dado $x = 12.5$, e seja seu valor aproximado $\bar{x} = 12.6$ e $y = 5.15$ com seu valor aproximado $\bar{y} = 5.2$. Podemos ver que o erro absoluto $EA_x = EA_y = 0.1$ Para ter certeza da precisão, precisamos do erro relativo dado por:

- $|ER_x| = \frac{|EA_x|}{|\bar{x}|} = 7.9 \times 10^{-3} = 0.79\%$
- $|ER_y| = \frac{|EA_y|}{|\bar{y}|} = 0.0192 = 1.9\%$

Enfim, o erro relativo permite nós de ver a ordem de precisão de um número. Os dois erros apresentam o mesmo comportamento, quando o valor aproximado está próximo da unidade. Exemplo $\frac{|2.3-1.2|}{|1.2|} \approx |2.3-1.2|$

(3) **(*) Peso** 40%
Considere o sistema do ponto flutuante $\mathrm{FP}(10,3,1)$ que não use nem a truncatura ou o arredondamento. Quais dos seguintes números tem representação exata neste sistema? -10, 1001, 0.00012, 0, 24.5700, -2220.2, 0.0013254, $1.5 * 10^{-11}$, $-1.5 * 10^{9}$, $1.5 * 10^{-10}$, -4556, -0.000001, 0.00000000011, 99999999, -999999999
Qual será a representação deles no sistema FP quando usamos a truncatura, e qual quando usamos o arredondamento?
Resposta: No caso que o numero não precisa arredondamento e nem truncamento para ser representado com t dígitos significativos e é dado não no formato $0.d_1...d_t \times 10^e$ deixei a resposta com Não/Sim como aceites na avaliação (não marquei como erro). Se bem a resposta mais certa seria Não

Número	Truncamento	Arred	Resposta
-10	$-0.1 * 10^2$	$-0.1 * 10^2$	Não
1001	$0.1 * 10^4$	$0.1 * 10^4$	Não
0.00012	$0.12 * 10^{-3}$	$0.12 * 10^{-3}$	Não
0	*	*	Não/Sim
24.5700	$0.245 * 10^2$	$0.246 * 10^2$	Não
-2220.2	$-0.222 * 10^4$	$-0.222 * 10^4$	Não
0.0013254	$0.132 * 10^{-2}$	$0.133 * 10^{-2}$	Não
$1.5 * 10^{-11}$	$0.15 * 10^{-10}$(*não existe*)	idem	Não
$-1.5 * 10^{9}$	$-0.15 * 10^{10}$(*não existe*)	idem	Não
$1.5 * 10^{-10}$	$0.15 * 10^{-9}$	idem	Sim
-4556	$-0.455 * 10^4$	$-0.456 * 10^4$	Não
-0.000001	$-0.1 * 10^{-5}$	idem	Não
0.00000000011	$0.11 * 10^{-9}$	idem	Não
99999999	$0.999 * 10^8$	$0.1 * 10^9$	Não
-999999999	$-0.999 * 10^9$	$-0.1 * 10^{10}$(*não existe*)	Não

(*) Nas maquinas (computadores) 0 é representado com bit nulos, é um valor especial que é reconhecido.

(4) Considere os três números $x = 9.274 * 10^5$; $y = 23.2$; $z = -2.3456 * 10^{-2}$. Compute na aritmética finita $\mathrm{FP}(10,3,1)$ que usa o arredondamento, a operação $(x+y) \cdot z$. Que valor toma este calculo? Escreva cada passagem que usou para obter o resultado. Qual é o erro absoluto e o erro relativo respeito a computação feita com a sua calculadora?

Resposta:

Passo 1: Representação de x, y e z em FP$(10,3,1)$

$$\overline{x} = 0.927 * 10^6; \quad \overline{y} = 0.232 * 10^2; \quad \overline{z} = -0.235 * 10^{-1}$$

Passo 2: Fazer a soma

$$\begin{aligned}\overline{x} + \overline{y} &= \left(0.927 + 0.232 * 10^2 * 10^{-6}\right) * 10^6 \\ &= \left(0.927 + 0.232 * 10^{-4}\right) * 10^6 \\ &= (0.927 + 0.0000232) * 10^6 \\ &= 0.9270232 * 10^6\end{aligned}$$

Passo 3: Aproximar a soma do Passo 2 em FP.

$$\overline{x + y} := \overline{\overline{x} + \overline{y}} = 0.927 * 10^6$$

Passo 4: Fazer o produto

$$\begin{aligned}\overline{x + y} * \overline{z} =& (0.927 * 10^6) * (-0.235 * 10^{-1}) \\ =& -0.217845 * 10^5\end{aligned}$$

Passo 5: Escrever o resultado do Passo 4 em FP(10,3,1)

$$\overline{\overline{x + y} * \overline{z}} = -0.218 * 10^5$$

(5) Dar um exemplo de dois números x, y para que a operação de subtração $x - y$ na aritmética finita FP$(10,5,1)$ da um erro relativo superior a 0.2 (20%). Sugestão: Estudar a teoria do *cancelamento subtrativo.*

Resposta: ...

(6) Determine o majorante do erro absoluto e relativo da operação $(xy + xz)$ na genérica aritmética finita FP(β, p, s).

Resposta: ...

(7) Escreva um algoritmo e o relativo código na linguagem de programação escolhida que permite dado como input um numero no sistema binário de obter a sua conversão em decimal.

Resposta: ...

(8) **(*) Peso** 30%
Escreva um algoritmo e o relativo código que dado como input um numero no sistema decimal dá em output a sua conversão em binário.

Resposta:

Algoritmo

Require: num, o numero no sistema decimal

$k \leftarrow -1$
while $num \neq 0$ **do** ▷ Terminamos o ciclo quando num é nulo
    $k \leftarrow k + 1$
    $r \leftarrow$ resto da divisão inteira de num por 2
    $b_k \leftarrow r$
    $num \leftarrow$ quociente da divisão inteira de num por 2
end while
return $b_k b_{k-1} \ldots b_1 b_0$ ▷ O output é o numero binário $b_k b_{k-1} \ldots b_1 b_0$

Código em Python

```
from __future__ import print_function

print('Insira numero decimal x= ',end='')
num = int(input())

lista = []
while(num != 0):
  r = num % 2
  num = num // 2
  lista.append(r)

print('Numero binario b= ', end='')
for i in range(len(lista) - 1, -1, -1):
  print(lista[i], end='')

print("\n")
```