UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
DISCIPLINA: Cálculo Numérico para Computação - INE5409
PROFESSOR: Júlio Felipe Szeremeta

TRABALHO 01: UTILIZAÇÃO DA AVALIAÇÃO DO ERRO COMETIDO PARA APROXIMAÇÃO DE NÚMEROS

Lucas Pereira Da Silva

Florianópolis
Março de 2011

SUMÁRIO

1 CÁLCULO DE UMA RAIZ QUADRADA QUALQUER COM PRECISÇAO 10^{-T}

O algorítimo abaixo calcula a raiz quadrada de um número qualquer com uma precisão de T dígitos. Para realizar esta operação será utilizada a seguinte formula para aproximação da raiz quadrada:

$$X1 = \frac{X0 + \frac{C}{X0}}{2}$$

```
Ler C
Ler T
X0 = 1
X1 = (1+C)/2
Precisao = 10^-T
Erro = 1

Enquanto Erro > Precisao Faça
     X0 = X1
     X1 = (X0+C/X0)/2
     Erro = |X1-X0|/max{|X1|,|X0|}
Fim Enquanto
Escreva X1
```

O programa foi implementado na linguagem Java e para exemplificar, ao fazermos C = 10 e T = 5 foi impresso na tela o seguinte resultado: 3.162277660168379.

2 CÁLCULO DO NÚMERO NEPERIANO

A função do programa abaixo é calcular o valor do número neperiano (**e**). O valor de **e** pode ser calculado através de uma série de Taylor do seguinte formato:

$$e=\sum_{n}^{\infty}\frac{1}{n!}=\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+\frac{1}{3!}+\ldots$$

```
#Textos precedidos por “#” são comentários e não são
#incorporados ao código.

E1 = 1
E0 = 0
N = 0
Erro = 1

Enquanto Erro != 0 Faça #O operador “!=” significa diferente
     N = N+1
     E0 = E1
     E1 = E1+1/N!       #“N!” significa fatorial de N
     Erro = |E1-E0|/max{|E1|,|E0|}

     Escreva E1
     Escreva Erro
Fim Enquanto
```

No programa acima temos a variável E1 que é a N-ésima parcela da série descrita anteriormente e E0 é a N-1-ésima parcela da série. A variável Erro serve para nos indicar o quão perto estamos do verdadeiro número. Um Erro ≤ 10^{-T} indica que E1 possui T dígitos do valor exato de **e**.

O que o programa acima faz é a cada laço de repetição calcular E0, E1 e o Erro, além de imprimir estes dois últimos. O programa irá realizar estas operações até que ocorra um erro de truncamento, onde a nova parcela da série (1/N!), somada ao valor de E1 não poderá ser incorporado a mantissa do próprio E1. Quando isso ocorrer E1 será igual a E0 e portanto o laço será encerrado. Para implementação do código acima foi usada a linguagem de programação Java e a impressão dos valores de E1 e Erro a cada repetição se encontra na tabela abaixo:

N	E1	Erro

1	2	0.5
2	2.5	0.2
3	2.6666666666666665	0.062499999999999944
4	2.708333333333333	0.015384615384615332
5	2.7166666666666663	0.0030674846625766768
6	2.7180555555555554	5.109862033725888E-4
7	2.7182539682539684	7.299270073000844E-5
8	2.71827876984127	9.124004343056667E-6
9	2.7182815255731922	1.013777232521701E-6
10	2.7182818011463845	1.013777129910655E-7
11	2.718281826198493	9.216155641522974E-9
12	2.7182818282861687	7.680130103798931E-10
13	2.7182818284467594	5.907799927402019E-11
14	2.71828182845823	4.2198804297394E-12
15	2.718281828458995	2.813253619825475E-13
16	2.718281828459043	1.7644099357790124E-14
17	2.7182818284590455	9.802277420994503E-16
18	2.7182818284590455	0.0

Analisando a tabela acima um fato que percebemos na 18ª execução do laço é que Erro vale zero. Isso ocorre porque a parcela somada a E1 não pode ser incorporada a mantissa e E1 continuou possuindo exatamente o mesmo valor que no laço anterior. Isso fez com que em nosso código E1 fosse igual a E0 e assim o Erro calculado resultou em zero. Ou seja, no computador em que foi processado o algoritmo descrito anteriormente conseguimos obter **e** com 16 (expoente positivo do 17ª Erro) dígitos de confiança. Assim os 16 primeiros dígitos do verdadeiro **e** valem: 2.718281828459045.