

Aula 1: Introdução

Prof.: Paulo Roberto Nunes de Souza

1 Conteúdos da Disciplina ENG10128 - Matemática Discreta

1.1. Estimativa e medição de erros	[10 horas]
1.2. Raízes reais de equações reais	[12 horas]
1.3. Sistemas Lineares	[12 horas]
1.4. Interpolação	[08 horas]
1.5. Ajuste de curva	[10 horas]
1.6. Integração Numérica	[08 horas]

2 Avaliações

12/04/2011 Prova - P1, (1.1, 1.2)

19/05/2011 Trabalho - T1, (1.2)

24/05/2011 Prova - P2, (1.3, 1.4)

28/06/2011 Trabalho - T2, (1.4)

30/06/2011 Prova - P3 , (1.5, 1.6)

07/07/2011 Prova Final - PF

2.1 Média Parcial

$$MP = \frac{P1 + T1 + P2 + T2 + P3}{5}$$

$$Resultado\ Parcial = \begin{cases} \text{Aprovado} & \text{se } MP \geq 7 \\ \text{Prova Final} & \text{se } MP < 7 \end{cases}$$

2.2 Média Final

$$MF = \frac{MP + PF}{2}$$

$$Resultado\ Final = \begin{cases} \text{Aprovado} & \text{se } MF \geq 5 \\ \text{Reprovado} & \text{se } MF < 5 \end{cases}$$

3 Trabalho 1

3.1 Divisão de dois números

Criar um programa em linguagem C que realize a divisão de dois números reais sem utilizar a operação de divisão oferecida pela linguagem de programação.

3.2 Operação

Deve ser implementada em linguagem C uma funcionalidade que possibilite a divisão de dois números reais, tendo como resposta um número real com precisão $\varepsilon = 10^{-7}$.

Para implementar esta operação é terminantemente proibido utilizar a operação de divisão fornecida pela linguagem de programação.

3.3 Entrada e saída de dados

O programa deve ser executado com a seguinte sintaxe:

```
PROG_NUM1_NUM2
```

Sendo que PROG é o executável do programa e NUM1 e NUM2 são os valores reais que devem ser utilizados na operação. o programa deve emitir como resultado de sua execução no STDOUT apenas a resposta da operação seguido de um "fim de linha".

3.4 Tratamento de exceções

O programa deve estar preparado para tratar situações indesejada, porém previsíveis, ou seja, divisão por zero, número muito grande e número muito pequeno. Caso alguma dessas situações indesejadas ocorra, o programa deve escrever como resultado, no STDOUT, a string que representa a respectiva exceção: divisão por zero(NaN), número muito grande(Overflow), número muito pequeno(Underflow).

3.5 Grupo

Os trabalhos devem ser feitos em grupos de até 3 pessoas.

3.6 Entrega

O código fonte deverá ser enviado ao professor, no e-mail paulonunes@cca.ufes.br, até o dia **19/05/2011**.

3.7 Critérios de correção

O trabalho será corrigido segundo 5 critérios:

P Ausência de plágio

C Conformidade com o solicitado

A Acertividade dos resultados

E Entrega no prazo

D Desempenho computacional

Para a obtenção da nota do trabalho, cada critério receberá um valor entre 0 e 1 seguindo os testes efetuados pelo professor. A nota será o resultado da seguinte ponderação:

$$Nota = P \times C \times (6 \times A + 3 \times E + D)$$

4 Trabalho 2

4.1 Zoom digital

Criar um programa que receba como entrada um arquivo do tipo Bitmap, aplique neste arquivo uma ampliação de **4x** utilizando **interpolação bilinear** e grave o resultado num arquivo de saída no formato Bitmap

4.2 Formato Bitmap

Tanto o arquivo de entrada quanto o arquivo de saída devem ser compatíveis com o formato Bitmap. A especificação deste formato pode ser encontrada, dentre outros lugares, no site http://en.wikipedia.org/wiki/BMP_file_format.

Leia, compreenda e faça testes para interpretar o arquivo de entrada e gerar o arquivo de saída corretamente.

4.3 Ampliação

O programa sempre aplicará uma ampliação de 4x ao arquivo recebido como entrada. A técnica utilizada para a ampliação deve ser a **interpolação bilinear**. De tal forma que uma imagem de entrada de $N \times N$ pixels se torne uma imagem de saída de $(2N-1) \times (2N-1)$ pixels.

4.4 Entrada e saída de dados

O programa deve ser executado com a seguinte sintaxe:

```
PROG_entrada.bmp_saida.bmp
```

Sendo que PROG é o executável do programa, entrada.bmp é a imagem original e saída.bmp é a imagem ampliada a ser gerada.

4.5 Grupo

Os trabalhos devem ser feitos em grupos de até 3 pessoas.

4.6 Entrega

O código fonte deverá ser enviado ao professor, no e-mail paulonunes@cca.ufes.br, até o dia **28/06/2011**.

4.7 Critérios de correção

O trabalho será corrigido segundo 5 critérios:

P Ausência de plágio

C Conformidade com o solicitado

A Acertividade dos resultados

E Entrega no prazo

D Desempenho computacional

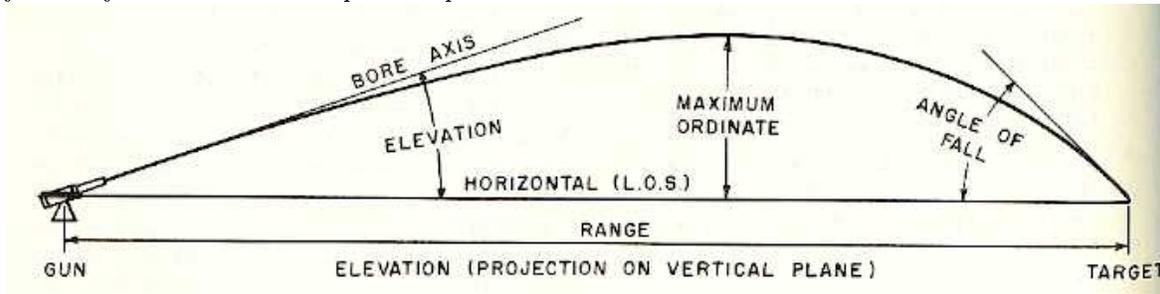
Para a obtenção da nota do trabalho, cada critério receberá um valor entre 0 e 1 seguindo os testes efetuados pelo professor. A nota será o resultado da seguinte ponderação:

$$Nota = P \times C \times (6 \times A + 3 \times E + D)$$

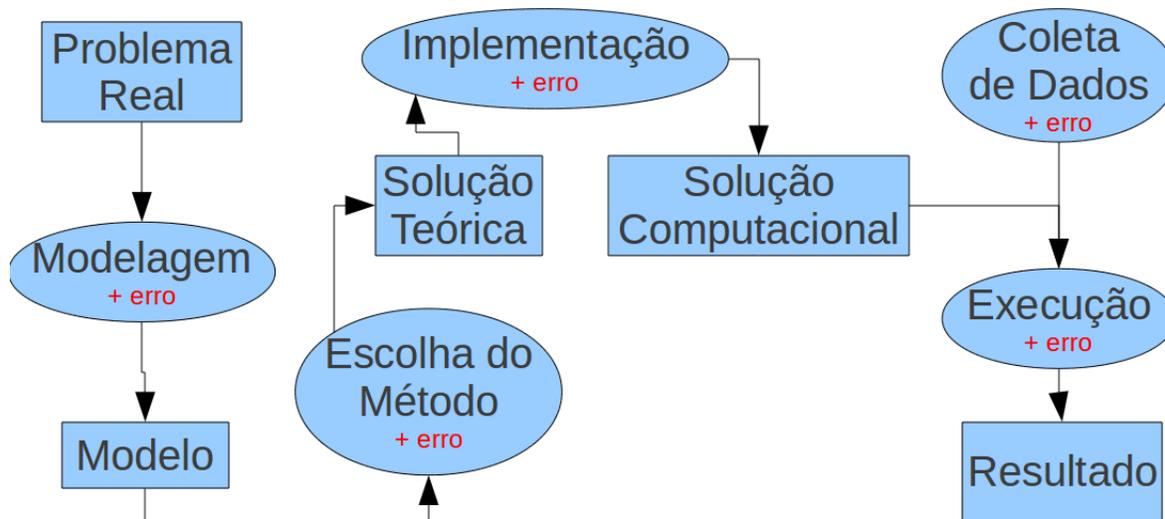
5 Algoritmos Numéricos

A disciplina de Algoritmos Numéricos, também conhecida como Cálculo Numérico ou Computação Científica, apresenta metodologias para resolver problemas matemáticos utilizando um computador. Esses problemas podem ser os mais diversos dentro de virtualmente qualquer área da ciência: física, química, matemática, engenharias, biologia, etc.

Uma área que historicamente impulsionou muito o desenvolvimento dos Algoritmos Numéricos é a área bélica. A situação mais direta de como esses algoritmos podem ser aplicados é no cálculo da trajetória de um projétil, seja ele um foguete, míssil, morteiro, etc. O grande interesse nessas situações é garantir que o projétil atinja o alvo com maior precisão possível.



Independentemente da área de aplicação, os algoritmos numéricos são métodos que, a grosso modo, se aproximam do resultado analítico dos problemas estudados, entretanto ao se controlar o nível de erro da resposta numérica, esta pode se tornar confiável perante ao custo da obtenção da resposta analítica exata. Tendo isto em mente, é importante que seja controlado o nível de erro inserido em cada etapa até se obter a resposta final do problema real.



Cada etapa mostrada no diagrama anterior, insere um erro com características diferente, sendo necessário conhecer cada um deles para tentar evitar ou minimizar os que forem possíveis.

5.1 Modelagem

Na fase de modelagem observa-se o problema real e representa-se este problema com ferramentas as matemáticas mais precisas possíveis, considerando a disponibilidade de recursos. Muitas vezes se faz necessário gerar um modelo matemático menos preciso, e conseqüentemente imbutir um certo nível de erro no resultado, pois o modelo mais preciso não poderia ser processado em tempo hábil. Por exemplo, gastar 30 horas de processamento para se gerar uma previsão meteorológica das próximas 24 horas.

No caso de um problema de queda livre de um objeto, pode-se analisar o problema considerando ou não a resistência do ar. O que nos levaria a duas modelagens diferentes.

Sem resistência do ar:

$$s(t) = \frac{gt^2}{2} + v_0t + s_0$$

Com resistência do ar:

$$s(t) = \frac{mg}{k}t - \frac{m}{k} \left(v_0 - \frac{mg}{k} \right) e^{-kt/m} + \frac{m}{k} \left(v_0 - \frac{mg}{k} \right) + s_0$$

5.2 Escolha do método numérico

Baseado no modelo gerado, vários métodos numéricos para solucionar este problema poderão ser utilizados, cada um com suas características de nível de erro e tempo de execução. Essas características devem ser ponderada para que o resultado final atenda às necessidades.

5.3 Implementação

Após definido o método numérico a ser utilizado, a escolha das estruturas e tipos de dados envolvidos na implementação do código também imbuem erros no resultado.

Na representação de um valor numérico pode-se utilizar alguns tipos de dados diferentes, por exemplo: inteiro, ponto flutuante de precisão simples, ponto flutuante de precisão dupla, ponto flutuante de precisão quádrupla, etc.

5.4 Coleta de dados

Antes de iniciar a execução do modelo, são necessárias informações iniciais para das início ao processamento. Essas informações são coletadas no mundo real utilizando-se instrumentos de medida que tem o seu nível de precisão e, portanto, o seu nível de erro. Esses dados alimentarem a solução computacional, inserem junto consigo os erros inerentes à sua medição.

Numa situação de medição de temperatura, pode-se utilizar vários tipos de termômetros, entretanto todos eles terão um limite de precisão no resultado desta medição.

5.5 Execução

A máquina que executa o programa sobre os dados coletados, por mais poderosa que seja, possui uma limitação física na sua precisão. Quando os valores calculados excedem este limite surgem erros decorrentes do próprio cálculo.

Ao final da execução, tem-se um resultado que deve ser analisado frente a valores esperados de resposta. Isto é necessário pois todos os erros se acumulam, podendo gerar um resultado completamente incoerente com a realidade.