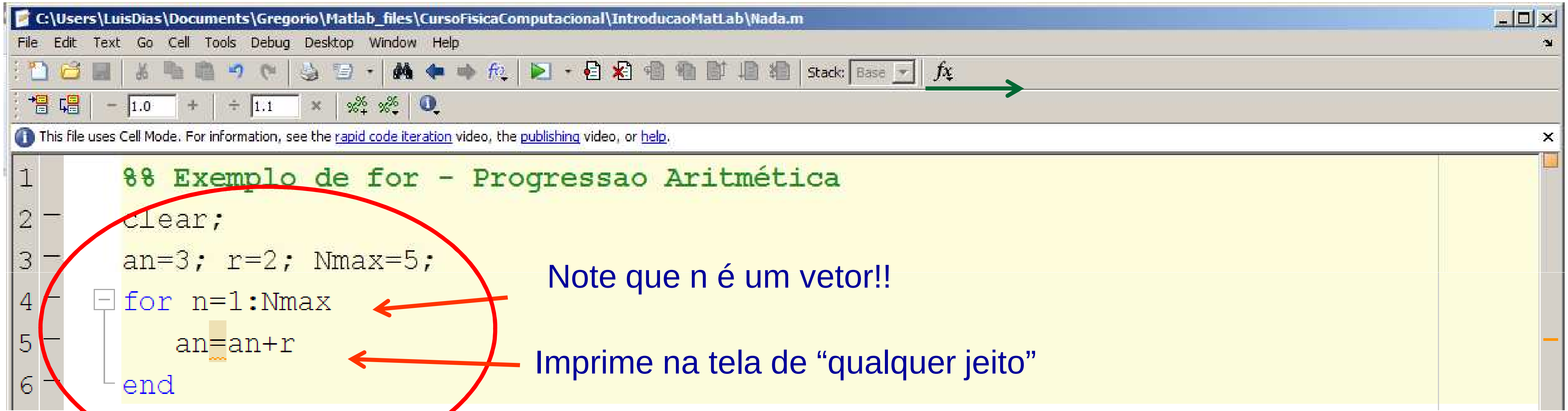
Aula 3
Loop tipo “for” e derivada numérica

Estruturas de loop: “for”

A estrutura “for” é usada para repetir um comando iterativamente.



Note que n é um vetor!!

Imprime na tela de “qualquer jeito”

Qual o último valor de an?

Estruturas de loop: “for”

Dica: melhore a formatação de saída dos seus dados com fprintf

C:\Users\LuisDias\Documents\Gregorio\Matlab_files\CursoFisicaComputacional\IntroducaoMatLab\Aula3_For.m

File Edit Text Go Cell Tools Debug Desktop Window Help

This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.

```
%% Exemplos de for
clear;
%% Exemplo 1 - Progressao Aritmética
an=3; r=2; Nmax=5;
fprintf('PA: n=%i: an=%i \n',0,an); % imprime na tela formatado (vide help)
for n=1:Nmax
    an=an+r;
    fprintf('PA: n=%i: an=%i \n',n,an);
end
disp('----------------'); % Apenas imprime na tela
```

Não imprime na tela (tem “;” no final)

Imprime “n” e “an” com o formato “%i” (números inteiros)

Imprime sem formatação

Experimente o formato “%f” ou “%.4f”...

Estruturas de loop: “for”

C:\Users\LuisDias\Documents\Gregorio\Matlab_files\CursoFisicaComputacional\IntroducaoMatLab\Aula3_For.m

```matlab
%% Exemplos de for
clear;
%% Exemplo 1 - Progressao Aritmética
an=3; r=2; Nmax=5;
fprintf('PA: n=%i: an=%i \n',0,an); % imprime na tela formatado (vide help)
for n=1:Nmax
    an=an+r;
    fprintf('PA: n=%i: an=%i \n',n,an);
end
disp('----------------'); % Apenas imprime na tela
%% Exemplo 2 - Progressao Geométrica
an=3; q=2; Nmax=5;
fprintf('PG: n=%i: an=%i \n',0,an);
for n=1:Nmax
    an=q*an;
    fprintf('PG: n=%i: an=%i \n',n,an);
end
```

Tarefas da aula 3

Lembrando: toda aula haverá tarefas!! (20% da média final!!!)

- Tarefa 1: *Gere uma matriz 100x100 onde o elemento geral da matriz é*

$$M_{ij} = i + j$$

Dica: para criar uma matriz NxN, use “ones(N,N)” ou “rand(N,N)” ou ....

Obs: Faça cada tarefa em um arquivo .m diferente!

Exemplo: Aula3_Tarefa1.m , etc,

Obtendo uma derivada numericamente

Derivada de uma função f(t) (contínua, diferenciável, etc.):

$$\boxed{\frac{df(t)}{dt} = \lim_{\Delta t \to 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}}$$

Para Δt “pequeno”, podemos aproximar:

$$\boxed{\frac{df(t)}{dt} \approx \frac{\Delta f}{\Delta t} = \frac{f(t+\Delta t) - f(t)}{\Delta t}}$$

Mas quão “pequeno” tem que ser o Δt ?????

Tarefas da aula 3 (cont.)

- Tarefa 2: *Dada a função* $f(x) = e^{-2x^2}$

calcule $\boxed{\frac{df(x)}{dx} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}}$ *em x=0 usando passos*

Δx *=1 , 0.1 , 0.01 , ... 10*$^{-8}$

Para cada passo, imprima Δx *e* $\frac{\Delta f}{\Delta x}$ *com 10 casas de precisão.*

Dica: para imprimir, utilize fprintf com formato "%.10f"

Obs: Para tarefas mais elaboradas como essa, procure fazer cada tarefa em um arquivo .m diferente.

Exemplo: Aula3_Tarefa2.m , etc,

Tarefas da aula 3 (cont.)

$$f(x) = e^{-2x^2}$$

- <u>Tarefa 3:</u> *Compare os resultados da tarefa anterior usando*

$$\boxed{\frac{df(x)}{dx} \approx \frac{f(x) - f(x - \Delta x)}{\Delta x}}$$

$$\boxed{\frac{df(x)}{dx} \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}}$$

e os mesmos valores de Δx*. Qual dos três é "melhor"? Por quê?*

Obs: Neste caso, pode fazer no mesmo arquivo da Tarefa anterior.

Exemplo: Aula3_Tarefa2e3.m

Tarefas da aula 3 (cont.)

$$f(x) = e^{-2x^2}$$

- <u>Tarefa 4:</u> *Faça um gráfico de f(x) e de sua derivada calculada via*

$$\frac{df(x)}{dx} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

no intervalo -1 ≤ x ≤ 1 usando o passo Δx *=0.1.*
Repita com Δx *=0.01.*

Dica 1 : defina f(x) e df/dx(x) em vetores de tamanhos <u>iguais</u>.
(isso gera algum problema? qual? como resolver?)
Dica 2 : você já sabe como plotar duas funções no mesmo gráfico!

Aula 3 – Parte 2
Primeira Simulação

Plotando múltiplas curvas no mesmo gráfico usando ‘hold on’.

File Edit Text Go Cell Tools Debug Desktop Window Help

This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.

```matlab
%% Plotando duas curvas no mesmo gráfico
clear;
vecx=1:0.01:10;
f1=1+vecx.^2;
f2=1+vecx.^3;
fig=figure(1);
plot(vecx,f1,'color','blue','Linewidth',1);
hold on; % "segura" a Figura.
plot(vecx,f2,'color','red','Linewidth',1);
legend('1+x^2','1+x^3');
```

“handle” do objeto “figure(1)”

“Segura”

Opção de cores

Opção de largura de linha

Um exemplo físico: decaimento radioativo

Decaimento radioativo de um átomo de Urânio:

$$\boxed{\frac{dN_U(t)}{dt} = -\frac{N_U(t)}{\tau}}$$

$N_U(t)$ - número de núcleos de Urânio no tempo t.

τ - tempo de vída médio (tempo) .

Solução analítica:

$$\int_{N_U(0)}^{N_U(t)} \frac{1}{N_U} dN_U = \int_0^t \frac{-dt'}{\tau} \Rightarrow \ln \frac{N_U(t)}{N_U(0)} = -\frac{t}{\tau}$$

$$\boxed{N_U(t) = N_U(0) e^{-t/\tau}}$$

Decaimento **exponencial** com o tempo!

Um exemplo físico: decaimento radioativo

Solução numérica de: $$\frac{dN_U(t)}{dt} = -\frac{N_U(t)}{\tau}$$

$$\boxed{\frac{dN_U(t)}{dt} \approx \frac{N_U(t+\Delta t) - N_U(t)}{\Delta t}}$$

- método de Euler!

$$N_U(t+\Delta t) = N_U(t) - N_U(t)\frac{\Delta t}{\tau}$$

$$\boxed{N_U(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) N_U(t)}$$

Começando em t=0, podemos calcular $N_U(\Delta t)$, $N_U(2\Delta t)$... $N_U(n\Delta t)$!

Tarefas da aula 3 (cont.)

- Tarefa 5: Considere um decaimento radioativo com tempo de vida médio τ=1s e população inicial $N_U(0)$= 100 núcleos.
  - *Calcule $N_U(t_n)$ usando o método de Euler para $t_n=n.\Delta t$ de $t_1=0$ até $t_N=5s$ com passo de* $\boldsymbol{\Delta t=0.5s}$.
  - *Para cada passo, imprima t_n e $N_U(t_n)$ com 10 casas de precisão.*
  - *Faça um gráfico de $N_U(t_n)$ vs t (use o símbolo '-o')*
  - *No* ***mesmo gráfico****, plote a solução analítica $N_U(t)$ para o decaimento (para os mesmos valores de tempo)*.

Dica 1 : Lembrando que $t_n=n.\Delta t$

Dica 2 : Você já conhece duas maneiras de plotar duas curvas no mesmo gráfico. Quais?

Tarefas da aula 3 (cont.)

- <u>Tarefa 6:</u> Considere um decaimento radioativo com tempo de vida médio τ=1s e população inicial $N_U(0)$= 100 núcleos.
  - *Repita o processo da Tarefa 5 para Δt=0.5, 0.2, 0.1 e 0.05 s (sem imprimir os resultados na tela) e faça um gráfico com as 4 curvas $N_U(t_n)$ vs t com símbolos '-o'.*
  - *No <u>mesmo gráfico</u> plote a solução analítica $N_U(t)$ para o decaimento (para os valores de tempo com menor passo) na cor* ***preta*** *em com* **'LineWidth', 3.**

Dica 1 : Defina um vetor Deltavec=[0.5 0.2 0.1 0.05] e faça um loop:

```
for Delta=Deltavec
    (...)
end
```

Dica 2: Dentro deste loop, plote a curva usando plot() e hold on;