

U.C. 21180

Computação Numérica

15 a 25 de novembro de 2024

”

E-fólio A | Instruções para a realização do E-fólio



- Leia estas instruções na totalidade antes de iniciar a resolução da prova.
- Este enunciado constitui o elemento de avaliação designado por "e-fólio A" no âmbito da avaliação contínua e tem a cotação total de 4 valores. A sua resolução deve ser entregue até às 23h55 do dia final do período de realização pelos alunos que escolheram a modalidade de avaliação contínua.
- A resolução deve ser entregue através de um único ficheiro compactado .zip que:
 - (i) contém os ficheiros .m que constituem o código dos programas, prontos a serem executados;
 - (ii) contém um ficheiro de nome relatorio.pdf (sem acento) com as informações solicitadas e/ou complementares de modo a permitir uma fácil compreensão do trabalho realizado. É desnecessário incluir uma listagem integral do código;
 - (iii) O nome do ficheiro .zip a entregar deve seguir a seguinte convenção para o seu nome,

NumeroAluno-PrimeiroNome-Apelido-21180-efA.zip

Por exemplo, um aluno com número 327555 e nome Paulo ... Costa, deverá dar o seguinte nome ao ficheiro,

"327555-Paulo-Costa-21180-efA.zip"(sem acentos).

- O ficheiro deve ser única e exclusivamente entregue através do recurso "E-fólio A" disponibilizado na plataforma (Nota: apenas é visível para os alunos inscritos em avaliação contínua), não sendo aceites trabalhos enviados por outras vias, como por exemplo por e-mail.

- Esta é uma prova de avaliação **individual** e não "um trabalho de grupo". A sua resolução deve provir unicamente do conhecimento adquirido e trabalho **original** desenvolvido pelo próprio aluno. Cumpra estritamente as normas de realização individual, como se estivesse num exame com consulta, onde pode consultar a documentação da UC mas não pode falar com ninguém nem consultar agentes de IA (ex. ChatGPT) na resolução do trabalho.
- Os alunos deverão saber distinguir claramente entre discutir os conteúdos abordados na unidade curricular (permitido) e discutir ou solicitar a resolução específica do e-fólio (não permitido).
- No caso de dúvidas de interpretação do enunciado, utilize o fórum de avaliação para pedidos de esclarecimento.

Grupo I [4 valores]

1. Considere a equação $f(x) = 0$ com $f(x) = -x + 20 \ln(1 + x^2) - a$. Pretende-se neste trabalho verificar experimentalmente a convergência do método de Newton utilizando como exemplo a equação dada.

Para um método iterativo, onde $\lim_{k \rightarrow \infty} x_k = r$, sendo r uma raiz da equação,

$$\lim_{k \rightarrow \infty} C_k = \lim_{k \rightarrow \infty} \frac{|r - x_{k+1}|}{|r - x_k|^p} = C, \quad C \neq 0$$

onde p é a ordem de convergência do método e C a constante de convergência assintótica. Para o método de Newton tem-se que,

$$p = 2$$

$$C = \left| \frac{f''(r)}{2f'(r)} \right|$$

- 1.1. [0.75] Dimensione o parâmetro a de modo a que $r = 4$, ou seja, $f(4) = 0$. Defina em linguagem Octave em ficheiros individuais de nomes fx.m, dfx.m e d2fx.m respectivamente as funções $y = f(x)$, $y = f'(x)$ e $y = f''(x)$. As funções devem admitir como argumento um vetor x , efetuar os cálculos vetorialmente e retornar um vetor y tal que $y(i) = f(x(i))$.
- 1.2. [0.75] Escreva um script em Octave de nome efa1.m que utilizando as funções definidas na alínea anterior gera em três janelas diferentes gráficos das funções $f(x)$, $f'(x)$ e $f''(x)$ para $x \in [-50, 200]$ com pelo menos 200 pontos, título, etiqueta nas abcissas e grelha. Inclua os gráficos obtidos no seu relatório. Dica: utilize o comando "figure(n)" para criar/selecionar previamente a janela em que vai gerar o gráfico.

- 1.3. [1.5] Escreva em Octave uma função que implementa o método de Newton com os seguintes argumentos de entrada e saída,

```
[r, e, n, x] = alg_newton(f, df, x0, emax, kmax)
```

A função deve ser implementada no ficheiro `alg_newton.m` e os argumentos de entrada são respectivamente uma referência para uma função genérica $f(x)$ e a sua função derivada, uma estimativa inicial da raiz x_0 , erro absoluto máximo e_{\max} desejado para a raiz e o número máximo iterações k_{\max} permitido.

As variáveis de saída são respectivamente a estimativa da raiz r e o respectivo erro absoluto ϵ , o vetor x com x_0 e todos os valores das iterações efetuadas x_0, x_1, \dots, x_n , sendo $n \leq k_{\max}$ o número de iterações efetuadas. Note que $x(1) = x_0$ e que o vetor x tem dimensão $n + 1$.

- 1.4. [1] Para demonstrar o funcionamento da função `alg_newton()` elabore um script de nome `efa2.m` que:

- Aplique a função `alg_newton()` à resolução da equação $f(x) = 0$ para 2 casos de valor inicial $x_0 = \{1, 10\}$, obtendo estimativas da raiz com erro $\leq 10^{-9}$;
- Imprima uma mensagem com a estimativa da raiz, o erro ϵ da estimativa da raiz, o intervalo em que se encontra a raiz (tendo em conta o erro determinado);
- Imprima o valor teórico de C e uma tabela com três colunas, $k \geq 0$, x_k e C_k , para todas as iterações efetuadas, excepto C_k para a última iteração;
- Inclua os dados de saída do script `efa2` no seu relatório. Comente a convergência do método e os valores de C_k obtidos.

Todos os ficheiros com código devem estar identificados com um cabeçalho similar ao seguinte,

```
/*  
** UC: 21180 - Computação Numérica  
** e-fólio A 2024-25  
**  
** Aluno: 327555 - Paulo Costa  
*/
```

Critérios de correção

- Não é permitida a utilização de variáveis globais (uso da palavra chave "global") nem da função `fplot()`.
- Não é permitida a utilização de funções já existentes no Octave análogas às que se pretendem desenvolver nem a utilização de funções de pacotes de software (packages) adicionais à instalação base do Octave.

- As funções pedidas devem implementar rigorosamente a interface de argumentos de entrada e de saída indicados. Não devem pedir dados ao utilizador nem imprimir dados ou gráficos não solicitados no enunciado.
- Os programas devem empregar sempre que possível operações vetoriais e/ou matriciais (ex. produto interno de vetores, etc) em detrimento de ciclos que manipulam simples escalares.
- O código dos programas deve estar correta e uniformemente indentado de modo a permitir a sua leitura fácil.
- Os programas devem estar estruturados, comentados e em conjunto com o relatório explicados de modo à fácil compreensão da sua estrutura e funcionamento.
- Os programas que não funcionem corretamente ou não cumpram todas as especificações ou sejam demasiado complexos \Rightarrow de 0 a 100% valores, sendo cada programa avaliado como um todo e tendo em conta a implementação das características pedidas.

Nota ética: Nunca é de mais referir que o código a apresentar como solução para este e-fólio deve ser 100% **original** do aluno. A probabilidade de duas pessoas apresentarem programas “quase iguais” é considerada nula. Isto é válido para qualquer par de alunos (cópia direta), assim como entre um aluno e qualquer outra fonte de informação (cópia indireta), em particular através da Internet (plágio), onde existem inúmeras soluções e código para os mais variados problemas, em sites, fóruns, blogs, simuladores, agentes de IA, etc.

FIM