

AULA 1

Introdução ao curso. Regras de avaliação.
Materiais de estudo. Revisão de conceitos
básicos de programação em python.

Laboratório Numérico 2020

Teóricas: Pedro Miranda pmmiranda@fc.ul.pt, (C8, 8.3.38)

Práticas: Fernando Soares, Pedro Mateus, Pedro Miranda

Objectivos

Resolução de problemas em Física Aplicada e Engenharia com utilização de métodos numéricos

Linguagem: python

Avaliação

Trabalho de grupo: 60% (*Projectos + Apresentações*) **obrigatório**

Exame final: 40%, nota mínima 8

(Para quem queira manter a nota prática anterior, será

Prática=40%, Exame=60% nota mínima 8)

Calendário 2020

20PL21 (4ª13)
19PL22 (2ª9:30)
20PL23 (2ª9:30)
20PL24 (4ª11)
0PL25 (4ª13)
8NO

← PL25: Quarta 13:00-15:00

Práticas

S	T	Q	Q	S	S	T	Q	Q	S	S	T	Q	Q	S
17-Feb	18-Feb	19-Feb	20-Feb	21-Feb	24-Feb	25-Feb	26-Feb	27-Feb	28-Feb	02-Mar	03-Mar	04-Mar	05-Mar	06-Mar
	T1		T2					T3		P0	T4		T5	
09-Mar	10-Mar	11-Mar	12-Mar	13-Mar	16-Mar	17-Mar	18-Mar	19-Mar	20-Mar	23-Mar	24-Mar	25-Mar	26-Mar	27-Mar
P1	T6		T7		P2	T8		T9		P3			T10	
30-Mar	31-Mar	01-Apr	02-Apr	03-Apr	06-Apr	07-Apr	08-Apr	09-Apr	10-Apr	13-Apr	14-Apr	15-Apr	16-Apr	17-Apr
P4	T11		T12			T13							T14/E1	
20-Apr	21-Apr	22-Apr	23-Apr	24-Apr	27-Apr	28-Apr	29-Apr	30-Apr	01-May	04-May	05-May	06-May	07-May	08-May
P5			T15		P6	T16		T17		P7	T18		T19	
11-May	12-May	13-May	14-May	15-May	18-May	19-May	20-May	21-May	22-May	25-May	26-May	27-May	28-May	29-May
P8	T20		T21		P9	T22		E2		P10				

22 teóricas

11 práticas: 0 (introdução) 1-4 (Projeto 1) 5 (Apresentação) 6-9 (Projeto 2) 10 (Apresentação)

Envio dos projetos: 24:00 dos dias E1, E2

Projetos

Projeto 1

Competências de programação: variáveis, estruturas de controlo, funções

Input/output básico

Gráficos 1D

Métodos numéricos: regressão linear, equações não lineares,
interpolação, sistemas de equações

Projecto 2

Gestão de dados em multidimensões

Integração

Equações diferenciais

Séries temporais

Gráficos 2D e cartografia

Trabalho de grupo

Grupos de 2 membros

2 **projetos** por grupo (A,B)

Os projetos deverão ser enviados nas datas E1, E2 (file **PL21_G01_A.py** comentada) e defendidos nas aulas P5 e P10

O trabalho prático (projetos) é **obrigatório**

Nas apresentações deverão falar os dois membros do grupo em igual tempo. A ordem será alternada entre o Projeto A e o Projeto B. Na apresentação será entregue um ppt/pdf.

Importante

O trabalho é em grupo mas a avaliação é individual

Grupos de ≤ 2

Será valorizada a **originalidade** do Código e das apresentações

Há marcação de presenças nas aulas práticas: máximo de 3 faltas nas aulas P2-P11

Em caso de **força maior** as apresentações poderão ser apresentadas no horário de outra turma. O mesmo se aplica a presenças.

Estudo

Notas do curso e ppts das aulas (progressivamente...)

fenix

Help: google “python anything”

python

Existem 2 versões (python 2 e python 3) não totalmente compatíveis... Quando se utiliza código já escrito pode ser necessário ter os 2.

Vamos usar a versão **3.6**, na distribuição **anaconda**, recorrendo à interface gráfica **spyder**

A versão 3.6 é mais moderna e consistente, mas existem aplicações que só correm na 2.7.

spyder

Interface gráfica para o python: inclui editor, linha de comandos (ipython), visualizador de variáveis (debug)

Pode ser necessário limpar a memória do spyder, de vez em quando:

```
reset #apaga tudo
```

```
plt.close('all') #apaga todas as figuras
```

Nota: os scripts *.py são ficheiros de texto e podem ser corridos na linha de comandos (sem acumular lixo na memória):

```
python script.py
```

Objetos

O python processa **objetos**

Cada objeto é identificado por um nome:

`X, x, xis, a22, b_2`

Notar que $x \neq X$

Cada objeto, consoante a **classe** a que pertença, pode conter conjuntos de **números**, **textos** e **código**

Os objetos mais simples são **escalares**: contêm um único número. Esse número é de um dado **tipo**

Tipos numéricos

Plain integer (32 bit)

Long integer (n bit)

Floating point (64 bit)

Complex (2 x 64bit)

Boolean (0/1)

Listas

As listas são objetos constituídos por **conjuntos** de outros objetos

Exemplos:

```
x=[0,1]
```

```
Y=[0,10.,'texto']
```

```
z=[x,Y]
```

Um elemento de uma lista é acedido na forma:

```
Y0=Y[0] # (==0)
```

```
Y1=Y[1] # (==10.)
```

arrays

O modulo **numpy** permite utilização de objetos multidimensionais que podem ser utilizados em operações algébricas de forma eficiente. Por exemplo:

```
X=np.array([0.,10.,20.,30.,40.])
```

define um **vetor** X do tipo **float**, a partir da lista indicada.

A instrução:

```
X2=X*2
```

Define um novo vetor X2, onde cada elemento é o dobro do elemento correspondente de x, i.e.

```
X[k]=x[k]*2, k=[0,1,2,3,4]
```

Módulos

```
from sympy import Symbol
import numpy as np
import math
Y=Symbol('y')
X=np.array([1,2,3])
Pi=math.pi
```

A instrução **import** dá acesso a bibliotecas de código já disponíveis na instalação. Por vezes pode ser necessário fazer o download de uma nova biblioteca:

Na linha de comandos (admin) na pasta *scripts*: **conda install biblioteca**