

# L<sup>A</sup>T<sub>E</sub>X: COMO FAZER EXAMES ESCRITOS INDIVIDUAIS E CORRIGIR SEM TRABALHO

FILIPPE GAMA FREIRE

1. Introdução	1
2. Incorporação de Python em L <sup>A</sup> T <sub>E</sub> X	1
2.1. Instalação do PythonTex	2
2.2. O PythonTex	2
3. Um pequeno exemplo	4
4. Conclusão	5
References	5

Contents

## 1. INTRODUÇÃO

Com o necessário aumento do número de alunos por docente no ensino superior e com as mesmas salas de aula, pequenas, disponíveis é necessário fazer diversas versões dos exames de modo a evitar:

- o abuso da utilização de colegas para os vigiar
- a cópia quase inevitável

Por outro lado, como acho que os exames são um dos melhores momentos para ensinar, a publicação da resolução do exame, nas suas diversas versões, torna-se um processo fastidioso.

O objectivo principal deste exercício é mostrar que com L<sup>A</sup>T<sub>E</sub>X e o auxílio de uma linguagem matemática, por exemplo Python, é possível não só tornar automática a produção de enunciados individuais mas também a sua resolução automática.

## 2. INCORPORAÇÃO DE PYTHON EM L<sup>A</sup>T<sub>E</sub>X

Há a possibilidade de fazer com que o Python escreva um documento em L<sup>A</sup>T<sub>E</sub>X, contudo, para o propósito, acho menos complicado inserir no L<sup>A</sup>T<sub>E</sub>X código de Python.

Existem diversos pacotes para o L<sup>A</sup>T<sub>E</sub>X utilizar códigos externos, dos quais os que uso com mais frequência são:

- [PythonTex](#), que é o pacote que vou aqui utilizar

- [Python](#)
- [SageTeX](#)

**2.1. Instalação do PythonTeX.** A instalação do  $\text{\LaTeX}$  nas diversas plataformas (OS's) está bem documentada na internet.

A instalação de pacotes de  $\text{\LaTeX}$  e Python também dependem um pouco do OS e aqui estou apenas interessado no Apple Max OS 10.11 ou superior. Para outras versões do OS 10, estas instruções não variam e para outras plataformas há documentação disponível.

A distribuição de  $\text{\LaTeX}$ , [MacTeX](#).

Para instalar pacotes do  $\text{\LaTeX}$  eu prefiro o uso de um gestor, [TeX Live Utility](#), normalmente incluído na distribuição.

O PythonTeX pode ser instalado deste modo.

É claro que é necessário possuir uma instalação de Python. Para isso eu utilizo o gestor de pacotes [Homebrew](#).

Resumindo:

Instalar:

- Python
- $\text{\LaTeX}$
- PythonTeX

**2.2. O PythonTeX.** O PythonTeX tem documentação disponível. Para aqui o que interessa é a declaração de comandos Python no  $\text{\LaTeX}$ , que é feito no bloco

```
.  
.   
\usepackage{pythontex}  
.   
.   
\begin{document}  
\begin{pcode}  
.  
.  
.  
\end{pcode}  
\end{document}
```

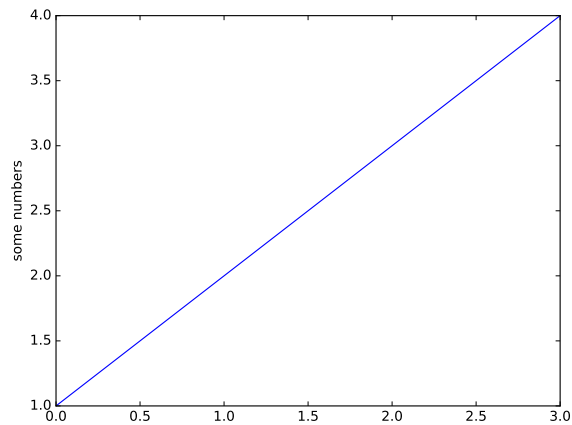
Nestes blocos, não só podemos fazer cálculos mas também, por exemplo, fazer e pôr no texto uma figura.

O bloco

```
\begin{pycode}
from matplotlib.pyplot import *
plot([1,2,3,4])
ylabel('some numbers')
figure(figsize=(5, 3))
close()
savefig("myplot.pdf", bbox_inches="tight")

print(r"\begin{center}"
      + r"\includegraphics[width=\textwidth]{myplot.pdf}"
      + r"\end{center}")
\end{pycode}
```

produz:



Outra instrução importante é

`\py{}`

Por exemplo, o bloco

```
\begin{pycode}
ver=1
if ver==1:
    A=1
else:
    A=2
```

```
\end{pycode}
A versão do exame é \py{A}
```

produz:

A versão do exame é a 1

Para uma ideia mais completa é conveniente ler as [instruções do PythonTex](#) ou talvez melhor [a sua introdução](#).

### 3. UM PEQUENO EXEMPLO

De seguida apresenta-se um pequeno exemplo que é a melhor e a mais curta forma de expôr a ideia.

```
\begin{pycode}
from sympy import *

C=Symbol('C')
t=Symbol('t')
if ver==1:
    n=1
    CAo=1
elif ver==2:
    n=2
    CAo=2
sol=integrate(1/(-0.1*C**n),(C,CAo,C))
sol=solve(t-sol,C)[0]
\end{pycode}
```

A reacção  $A \rightarrow B$  de ordem  $\text{\py{n}}$  ocorre num reactor batch. A concentração inicial era de  $\text{\py{CAo}}$  M. Escreva a variação de  $C(t)$  e diga qual a concentração ao fim de 1 min.

**Solução**

$\left[\frac{dC}{dt}=kC^{\text{\py{n}}}\right]$   $\text{\quad com } C_{Ao}=\text{\py{CAo}} \text{ M}$   
se  $k=-0,1$  então  $C(t)=\text{\py{latex(sol)}}$

Ao fim de 1 min teremos  $\text{\py{round(sol.subs(t,1),3)}}$  M.

Produz o seguinte:

- A versão 1:

A reacção  $A \rightarrow B$  de ordem 1 ocorre num reactor batch. A concentração inicial era de 1 M. Escreva a variação de  $C(t)$  e diga qual a concentração ao fim de 1 min.

**Solução**

$$\frac{dC}{dt} = kC^1 \quad \text{com } C_{Ao} = 1 \text{ M}$$

se  $k=-0,1$  então  $C(t) = e^{-0.1t}$

Ao fim de 1 min teremos 0.905 M.

- A versão 2:

A reacção  $A \rightarrow B$  de ordem 2 ocorre num reactor batch. A concentração inicial era de 2 M. Escreva a variação de  $C(t)$  e diga qual a concentração ao fim de 1 min.

**Solução**

$$\frac{dC}{dt} = kC^2 \quad \text{com } C_{Ao} = 2 \text{ M}$$

se  $k=-0,1$  então  $C(t) = \frac{10.0}{t+5.0}$

Ao fim de 1 min teremos 1.667 M.

#### 4. CONCLUSÃO

Parece evidente que os objectivos estabelecidos foram atingidos, dependendo agora o sucesso da implementação, não apenas dos conhecimentos do professor, mas também da disciplina que lecciona.

Por fim resta acrescentar que a produção de questionários online com verificação automática de pontuação pode ser a evolução natural do processo aqui exposto.

#### REFERENCES

- [1] Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.
- [2] Geoffrey M Poore, *PythonTeX: reproducible documents with LaTeX, Python, and more*, Computational Science & Discovery, 8,(1) 2015
- [3] Google, *A todos os que ajudam - Obrigado!* 2016