# Software Livre científico: O que temos?

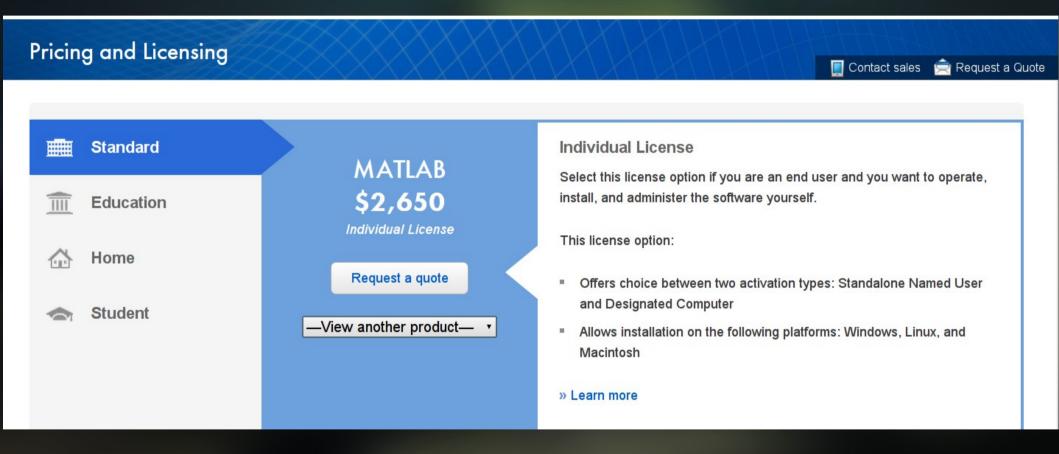
(De onde viemos? Para onde vamos?)

Renan Birck Pinheiro [renan.ee.ufsm@gmail.com]

#### O autor

- Formado em Engenharia Elétrica pela UFSM;
  - Curso onde o MATLAB é quem manda, na prática
- Hardware/software/sistemas embarcados;
- Tarefas diversas de simulação, análise de dados etc...;

### Problemas das Ferramentas Proprietárias



US\$ 2650 x dólar a R\$ 3,80 =  $\mathbb{R}$ \$ 10200 (e isso pelo MATLAB sem nenhuma *toolbox*)

#### Outros problemas

#### Além do custo:

- Cada ferramenta usa uma linguagem proprietária;
- Ferramentas criadas especificamente para aplicações científicas, dificultando outras tarefas comuns (ex.: criação de GUIs ou servidores, conexão com banco de dados etc...);
- Difícil (senão impossível) usar como backend de uma ferramenta mais complexa;
- Algoritmos fechados prejudicam a reprodutibilidade (necessária ao método científico).

#### Algumas ferramentas

- Octave, Scilab
- Sage
- Python + NumPy + SciPy + matplotlib + outras

#### Octave / Scilab

- http://www.octave.org E http://www.scilab.org
- Duas ferramentas que tentam se aproximar do MATLAB
  - Vantagem: compatibilidade de código
  - Porém:
    - Linguagem obsoleta (pouco suporte a OO, namespaces etc...);
    - Tiveram que copiar todos os vícios e bugs do MATLAB;
    - Na prática são clones incompletos.

#### Sage

- http://www.sagemath.org;
- Vários pacotes matemáticos na mesma ferramenta, com uma interface notebook;
- Grande, confuso e ruim de instalar;
- Voltado à matemática pura/aplicada.

### Então, vamos ao Python!

#### Por quê?

- Bibliotecas abrangentes para diversas aplicações (científicas ou não);
- Linguagem simples e moderna;
- Comunidade forte (Stack Overflow, GitHub, redes sociais, etc...).

- Dica: Anaconda
  - http://continuum.io/downloads
  - Instalação fácil do ambiente Python + bibliotecas de uso científico
    - Mesmo que você não seja root
    - Mesmo que você esteja usando Windows
    - Não afeta o Python do SO (ótimo para distros stable).

#### Por que não?

- Problemas com desempenho;
- Ambiente menos integrado;
- Falta de bibliotecas para aplicações específicas.

#### NumPy

 Permite trabalhar com matrizes, números complexos etc... de forma nativa, além de fornecer outras funções

#### SciPy

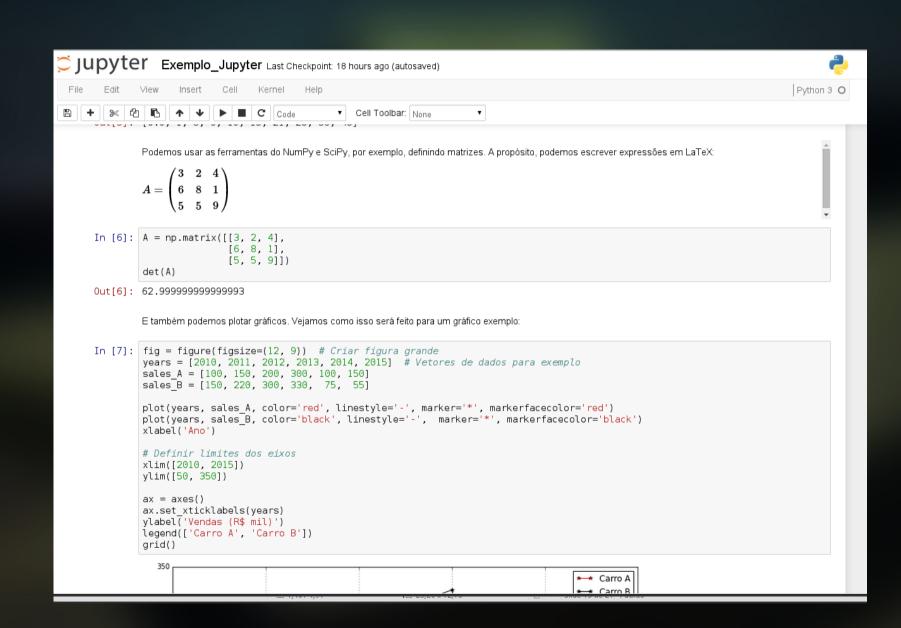
- Funções para integração numérica, processamento de sinais e de imagens, otimização etc...
- Usa a NumPy como framework

#### matplotlib

- Criação de gráficos 2D/3D;
- Duas formas de uso:
  - Programação imperativa (similar ao MATLAB)
  - OO (cria-se um objeto gráfico e daí se constrói a imagem)

#### ipython / Jupyter

- Permite trabalhar com Python (e com diversas outras linguagens) de forma interativa, dentro do browser.
- Na prática é um "editor de texto" no qual se mistura texto e código/resultados.



#### pandas

 Trabalhar com arquivos CSV, Excel etc... de forma simples e eficiente.

## Alguns exemplos de aplicação

#### Exemplo 1: FFT

- Transformada rápida de Fourier
  - Passar sinal do tempo para frequência
  - Uma das ferramentas mais usadas em EE, processamento de sinais etc...

#### Octave

# Python

#### Exemplo 2: filtrar sinal

- Sinal (senoide) com mistura de frequências
- Queremos pegar apenas uma delas

#### Octave

# Python

# OBRIGADO PELA ATENÇÃO!

Slides e materiais dessa apresentação: http://github.com/renanbirck/sfdsm-2015

#### Contatos:

- http://github.com/renanbirck
- http://fb.com/renanbirck
- http://twitter.com/birckrenan