

“KTHXBAI, MATLAB!”

Migrando a Academia para o Python

Melissa Weber Mendonça

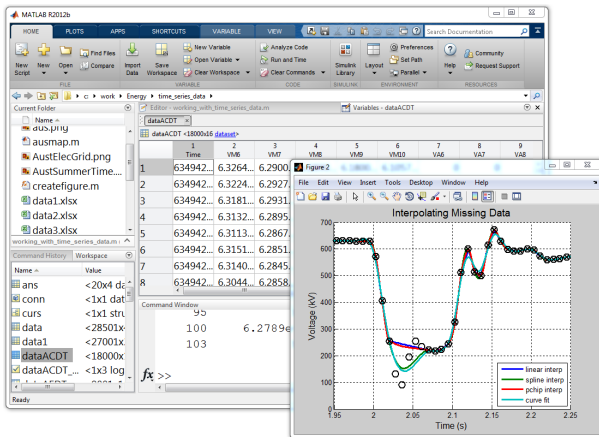
Quem (não) sou eu?

:)

O que estou fazendo aqui?

Pedindo ajuda!

Linguagem de alto nível e um ambiente interativo para computação numérica, visualização e programação.



Linguagem de alto nível e um ambiente interativo para computação numérica, visualização e programação.

```
A_aum = [A b;c' 0];
disp('Tableau 0:')
A_aum

% Custo relativo
[cmin,ind_entra] = min(A_aum(m+1,:));

iteracao = 1;
while ( cmin < 0 )
    % Primeiramente, vamos encontrar a menor razao entre o
    % lado direito e os elementos da coluna ind_entra.
    razao = 1e10;
    ind_sai = 0;
    for j = 1:m
        if ( A_aum(j,ind_entra) > 0 )
            if ( b(j)/A_aum(j,ind_entra) < razao )
                razao = b(j)/A_aum(j,ind_entra);
                ind_sai = j;
            end
        end
    end
    % razao
    % ind_sai
    % Agora, temos que pivotar no elemento A(ind_sai, ind_entra)
    if (ind_sai ~= 0)
        A_aum(ind_sai,:) = A_aum(ind_sai,+)/A_aum(ind_sai,ind_entra);
        for i = 1:m+1
            if (i ~= ind_sai)
                mult = A_aum(i,ind_entra);
                A_aum(i,:) = A_aum(i,:) - mult*A_aum(ind_sai,:);
            end
        end
    end
    cmin = A_aum(m+1,ind_sai);
    ind_entra = ind_sai;
end
```

Estado atual das coisas

- “Todo mundo usa MATLAB por que todo mundo usa MATLAB”

Estado atual das coisas

- “Todo mundo usa MATLAB por que todo mundo usa MATLAB”
- “Já que temos a licença, vamos usar”

Estado atual das coisas

- “Todo mundo usa MATLAB por que todo mundo usa MATLAB”
- “Já que temos a licença, vamos usar”
- “Não tenho tempo para aprender outra linguagem”

Por que se usa MATLAB?

- ***Stack científico completo vs.***
Python+NumPy+Scipy+matplotlib+ipython

Por que se usa MATLAB?

- *Stack* científico completo vs.
Python+NumPy+Scipy+matplotlib+ipython
- ***Toolboxes***

Por que se usa MATLAB?

- *Stack* científico completo vs. Python+NumPy+Scipy+matplotlib+ipython
- *Toolboxes*
- **Prototipagem rápida**

Por que se usa MATLAB?

- *Stack* científico completo vs.
Python+NumPy+Scipy+matplotlib+ipython
- *Toolboxes*
- Prototipagem rápida
- **Biblioteca enorme de rotinas básicas**

Por que se usa MATLAB?

- *Stack* científico completo vs. Python+NumPy+Scipy+matplotlib+ipython
- *Toolboxes*
- Prototipagem rápida
- Biblioteca enorme de rotinas básicas
- **Debugagem/controlado de variáveis muito fácil (e visual)**

Por que se usa MATLAB?

- *Stack* científico completo vs. Python+NumPy+Scipy+matplotlib+ipython
- *Toolboxes*
- Prototipagem rápida
- Biblioteca enorme de rotinas básicas
- Debugagem/controle de variáveis muito fácil (e visual)
- **Profiler, mlint, autocomplete, IDE: “Fácil”, “bonito” (?)**

Por que se usa MATLAB?

- *Stack* científico completo vs. Python+NumPy+Scipy+matplotlib+ipython
- *Toolboxes*
- Prototipagem rápida
- Biblioteca enorme de rotinas básicas
- Debugagem/controle de variáveis muito fácil (e visual)
- Profiler, mlint, autocomplete, IDE: “Fácil”, “bonito” (?)
- **GPU's, paralelização**

Por que se usa MATLAB?

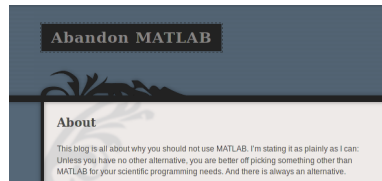
- *Stack* científico completo vs. Python+NumPy+Scipy+matplotlib+ipython
- *Toolboxes*
- Prototipagem rápida
- Biblioteca enorme de rotinas básicas
- Debugagem/controle de variáveis muito fácil (e visual)
- Profiler, mlint, autocomplete, IDE: “Fácil”, “bonito” (?)
- GPU's, paralelização
- **Gráficos: visualizar (e editar!) gráficos no momento da execução**

Por que se usa MATLAB?

- *Stack* científico completo vs. Python+NumPy+Scipy+matplotlib+ipython
- *Toolboxes*
- Prototipagem rápida
- Biblioteca enorme de rotinas básicas
- Debugagem/controle de variáveis muito fácil (e visual)
- Profiler, mlint, autocomplete, IDE: “Fácil”, “bonito” (?)
- GPU's, paralelização
- Gráficos: visualizar (e editar!) gráficos no momento da execução
- **Documentação: Help**

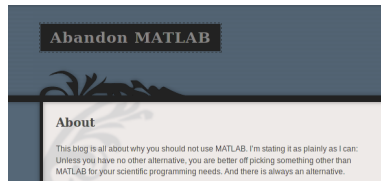
Por que *não* usar MATLAB?

- Código fechado



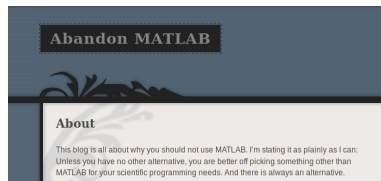
Por que *não* usar MATLAB?

- Código fechado
- Caro



Por que *não* usar MATLAB?

- Código fechado
- Caro
- Não é uma linguagem de uso geral



Por que *não* usar MATLAB?

```
>> x = int8(100)
```

```
x =
```

```
100
```

```
>> y = 400
```

```
y =
```

```
400
```

```
>> z = x+y;
```

```
>> z
```

```
z =
```

```
127
```

WAT?

Detalhes (nem tão pequenos)

- **() vs. []: Permite visualizar a diferença entre acessar elementos e fazer uma chamada de função.**

Detalhes (nem tão pequenos)

- () vs. []: Permite visualizar a diferença entre acessar elementos e fazer uma chamada de função.
- **Funções e classes podem ser definidas em qualquer lugar, inclusive na linha de comando (no MATLAB, precisam estar em arquivos isolados)**

Detalhes (nem tão pequenos)

- () vs. []: Permite visualizar a diferença entre acessar elementos e fazer uma chamada de função.
- Funções e classes podem ser definidas em qualquer lugar, inclusive na linha de comando (no MATLAB, precisam estar em arquivos isolados)
- **Legibilidade**

Detalhes (nem tão pequenos)

- () vs. []: Permite visualizar a diferença entre acessar elementos e fazer uma chamada de função.
- Funções e classes podem ser definidas em qualquer lugar, inclusive na linha de comando (no MATLAB, precisam estar em arquivos isolados)
- Legibilidade
- **Python tem estruturas de dados melhores: dicionários**

Detalhes (nem tão pequenos)

- `()` vs. `[]`: Permite visualizar a diferença entre acessar elementos e fazer uma chamada de função.
- Funções e classes podem ser definidas em qualquer lugar, inclusive na linha de comando (no MATLAB, precisam estar em arquivos isolados)
- Legibilidade
- Python tem estruturas de dados melhores: dicionários
- **Namespaces**; `import`

Broadcasting

Se A é $20 \times 1 \times 15$, B é $20 \times 12 \times 1$, C é $1 \times 12 \times 15$, no MATLAB teríamos

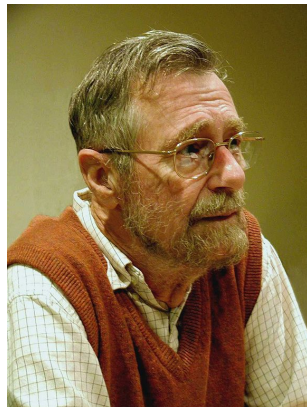
```
D = bsxfun(@times, bsxfun(@minus,B,C), A);
```

e em Python

```
D = A*(B-C)
```

Índices

- “A Case against the GO TO Statement”
- Odiava BASIC
- **“Why numbering should start at zero”**



Dijkstra

MEX files

```
subroutine mexfunction(nlhs, plhs, nrhs, prhs)
implicit none
integer*8 mxGetPr, mxCreateDoubleMatrix
integer*8 mxGetM
integer*8 nlhs, nrhs, plhs(*), prhs(*)
integer*8 poly_ptr, value_ptr, point_ptr
integer*8 p, n
real*8 poly(463203), point(463203), value
character*100 message

message=' '

poly_ptr = mxGetPr(prhs(1))
p = mxGetM(prhs(1))
point_ptr = mxGetPr(prhs(2))
n = mxGetM(prhs(2))

write(message, FMT='(E8.1)') 0.000
CALL mexPrintf(message)

plhs(1) = mxCreateDoubleMatrix(1_8, 1_8, 0)
value_ptr = mxGetPr(plhs(1))

call mxCopyPtrToReal8(poly_ptr,poly,p)
call mxCopyPtrToReal8(point_ptr,point,n)
call mxCopyPtrToReal8(value_ptr,value,1_8)

call comp_compute_poly(value,poly,p,point,n,message)

call mxCopyReal8ToPtr(value,value_ptr,1_8)

return
end
```

MEX files

```
subroutine mexfunction(nlhs, plhs, nrhs, prhs)
implicit none
integer*8 mxGetPr, mxCreateDoubleMatrix
integer*8 mxGetM
integer*8 nlhs, nrhs, plhs(*), prhs(*)
integer*8 poly_ptr, value_ptr, point_ptr
integer*8 p, n
real*8 poly(463203), point(463203), value
character*100 message

message=''

poly_ptr = mxGetPr(prhs(1))
p = mxGetM(prhs(1))
point_ptr = mxGetPr(prhs(2))
n = mxGetM(prhs(2))

write(message, FMT='(E8.1)') 0.000
CALL mexPrintf(message)

plhs(1) = mxCreateDoubleMatrix(1_8, 1_8, 0)
value_ptr = mxGetPr(plhs(1))

call mxCopyPtrToReal8(poly_ptr,poly,p)
call mxCopyPtrToReal8(point_ptr,point,n)
call mxCopyPtrToReal8(value_ptr,value,1_8)

call comp_compute_poly(value,poly,p,point,n,message)

call mxCopyReal8ToPtr(value,value_ptr,1_8)

return
end
```

f2py

Desempenho

- C, C++, Fortran (sim, eu uso Fortran :))
- **Python tem um desempenho melhor naturalmente!**

Comunidade

~~Vocês!~~

Nós :)

Muito além da técnica...

- **Open Science**

Muito além da técnica...

- Open Science
- **Reprodutibilidade**

Muito além da técnica...

- Open Science
- Reprodutibilidade
- **Universidades Públicas**

?

Impedimentos

- Técnicos (bem poucos!)

Impedimentos

- Técnicos (bem poucos!)
- **Humanos**

Cultura Acadêmica

- Na academia a teoria não é o problema; o conservadorismo é pior.

Cultura Acadêmica

- Na academia a teoria não é o problema; o conservadorismo é pior.
- Publicações usando Python

Cultura Acadêmica

- Na academia a teoria não é o problema; o conservadorismo é pior.
- Publicações usando Python
- **Converter alunos e depois converter professores!**

Matemática

- **Purismo**

Matemática

- Purismo
- **Isolamento**

Matemática

- Purismo
- Isolamento
- **Código raramente compartilhado (às vezes, escondido de propósito!)**

Matemática

- Purismo
- Isolamento
- Código raramente compartilhado (às vezes, escondido de propósito!)
- **“Implementação é detalhe”**

“Introdução à Computação Científica”

Ciência da Computação
(só computação, sem matemática)

VS.


“Computação Científica”
(só matemática, sem computação)

Casos de sucesso

MITOPENCOURSEWARE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Subscribe to the
OCW Newsletter

Enter Email

 Courses ▾ About ▾ Donate ▾ Featured Sites ▾ Search

Home » Courses » Electrical Engineering and Computer Science » Introduction to Computer Science and Programming

Introduction to Computer Science and Programming

OCW Scholar

COURSE HOME <

SYLLABUS

SOFTWARE


REFERENCES

UNIT 1

UNIT 2

UNIT 3

DOWNLOAD COURSE MATERIALS



Instructor(s)
Prof. John Guttag

Level
Undergraduate

CITE THIS COURSE

Many of the problem sets focus on specific topics, such as virus population dynamics, word games, optimizing routes, or simulating the movement of a Roomba. (Roomba photograph courtesy of Stephanie Booth on Flickr; virus image courtesy of the CDC; Boggle photograph courtesy of Angelina on Flickr; MIT campus map image courtesy of RahulG on Flickr.)

21 / 25

Casos de sucesso

Home UiO > The Faculty of Mathematics and Natural Sciences

UiO : Faculty of Mathematics and Natural Sciences

Home Research Studies Student life Services and tools About People

About

Collaboration

CSE - Computing in Science Education

- In the media
- News
- portal with resources
- Publications
- Teaching resources

CSE - Computing in Science Education

The goal of the CSE-project is to include computing as a natural tool for all science and engineering students from the first semester of their undergraduate studies. Not as a substitute for more traditional approaches, but as an extension of the classical toolbox.




Foto: Colourbox

- [Why Computing in Science Education?](#)
- [How we do it](#)
- [Background and history](#)

Do que precisamos?

- **Material didático**

Do que precisamos?

- Material didático
- **Troca de experiências**

Do que precisamos?

- Material didático
- Troca de experiências
- **Documentação**

Do que precisamos?

- Material didático
- Troca de experiências
- Documentação
- **Ambientes amigáveis multiplataformas**

Exemplos

- “Janelinha branca”



- StackOverflow

... tudo em inglês!

**Ajudem-me a migrar a academia
para o Python!**

Contato

@melissawm
www.mtm.ufsc.br/~melissa