

Computação Científica com MATLAB

Melissa Weber Mendonça
`melissa.mendonca@ufsc.br`

Aula passada...

- ▶ `if`; “e”; “ou”;
- ▶ Células
- ▶ Estruturas (*structs*)
- ▶ Importação de dados

cell2struct

A função `cell2struct` cria uma estrutura a partir dos dados contidos na célula:

Se

```
>> c = {'florianopolis', 23.5;  
        'sao jose', 22.1;  
        'joinville', 26.9}
```

e

```
>> f = {'cidade', 'temperatura'}
```

então

```
>> s = cell2struct(c, f, 2)
```

resulta em uma estrutura `s` que podemos consultar da seguinte forma:

```
>> s(1)
```

Importar dados

Para importarmos dados, o método mais fácil é utilizar a interface gráfica do MATLAB, selecionando

File → Import Data

Para verificar os tipos de arquivo suportados e as funções disponíveis, consulte o Help.

xlsread

Para ler arquivos de planilha gerados pelo Microsoft Excel, usamos

```
>> [dados,texto,resto] = xlsread(arquivo)
```

Para ler os dados de uma planilha específica do arquivo, usamos

```
>> [dados,texto,resto] = xlsread(arquivo,planilha)
```

Em sistemas com o Microsoft Excel instalado, pode-se usar

```
>> [dados,texto,resto] = xlsread(arquivo,-1)
```

para abrir uma janela do Excel e selecionar os dados a serem importados interativamente.

importdata

Para importarmos dados de maneira automática, podemos usar o comando

```
>> importdata(arquivo, separador, ncabecalho)
```

Leitura de dados numéricos: load

Para lermos um arquivo com dados **numéricos** chamado `dados.txt`, usamos o comando

```
>> load dados.txt;
```

Em seguida, na variável `dados` estarão contidos os valores obtidos do arquivo `dados.txt`.

Se quisermos também podemos usar a sintaxe

```
>> A = load('dados.txt')
```

Escrita em arquivos: save

Para salvarmos alguma variável em um arquivo, podemos usar o comando

```
>> save('arquivo.txt','variavel')
```

Porém, este comando salva o arquivo no formato MAT, que é um formato próprio do MATLAB, ilegível para humanos. Assim, para salvarmos em um arquivo texto simples, acrescentamos a opção '-ascii'.

Exemplo:

```
>> dados = rand(3,4);  
>> save('dadosout.txt','dados','-ascii')
```


Abrir e fechar um arquivo

Para abrir um arquivo chamado `nome.txt`, usamos o comando

```
arquivo = fopen('nome.txt')
```

Sempre que abrimos um arquivo, precisamos fechá-lo antes de sair do nosso programa. Para isso, usamos o comando

```
fclose(arquivo)
```

Comandos: leitura

Para ler dados de um arquivo, precisamos indicar que tipo de informação estamos procurando. Isto é feito através dos *formatos* abaixo:

- ▶ Números inteiros: %d ou %u
- ▶ Números reais: %f (notação decimal) ou %e (notação científica)
- ▶ Texto com espaços: %c
- ▶ Texto sem espaços: %s
- ▶ Nova linha: \n (sinaliza o fim de uma linha de dados)

Para lermos dados de um arquivo em uma célula, usamos

```
C = textscan(arquivo, '%d')
```

Para lermos dados de um arquivo em uma matrix, usamos

```
A = fscanf(arquivo, '%d')
```

Exemplo

1. Crie um arquivo chamado

`info.txt`

no mesmo diretório em que está salvando seus programas, com um número inteiro dentro.

2. No console, faça

```
>> arquivo = fopen('info.txt')  
>> a = fscanf(arquivo,'%d')  
>> fclose(arquivo)
```

Verifique que a variável `a` vale o mesmo que seu inteiro no arquivo.

Exercício

Repita o exercício anterior (ler UM dado de um arquivo), agora com um número real:

```
>> real = fscanf(arquivo,'%f')
```

e depois repita com um texto:

```
>> texto = fscanf(arquivo,'%s')
```

Tente colocar o formato errado e observe o erro produzido.

Lista de dados

Se quisermos ler uma lista de números inteiros, por exemplo, devemos informar o *padrão* dos dados.

Exemplo: se no arquivo temos

1 2 3 4 5

precisamos usar o comando

```
v = fscanf(arquivo,'%d')
```

v será um vetor *coluna*.

Exercício

Ler uma matriz 3×3 de dados de um arquivo .txt

Cuidado com a ordem dos dados!

matrix.txt

```
1 2 3  
4 5 6  
7 8 9
```

```
arquivo = fopen('matriz.txt');  
A = fscanf(arquivo,'%f');  
fclose(arquivo);  
A
```