



Capítulo 13

Gerenciamento de arquivos e de diretórios

Armazenamento de variáveis

- O comando

```
save filename lista de variáveis
```

salva as variáveis do espaço de trabalho do **MATLAB** especificadas no arquivo de nome *filename.mat*.

- Se o nome do arquivo for omitido, é usado o nome padrão *matlab.mat*.
- Se não for especificada uma lista de variáveis, todas as variáveis do espaço de trabalho são salvas.

- Lembrando da dualidade comando/função podemos usar *save* como função. Suponha que

lista de variáveis = {var1, var2, var3},

então

```
save('filename','var1','var2','var3')
```

é equivalente ao comando anterior;
(útil quando o nome do arquivo está em uma string.)

Armazenamento de variáveis

- Algumas observações a respeito do armazenamento:
 - As variáveis são salvas com precisão dupla e no formato binário do **MATLAB**.
 - É possível salvar as variáveis no formato de texto padrão ASCII. Veja documentação *on line* para maiores detalhes desta e de outras opções da função *save*.
 - As variáveis salvas em uma plataforma (Windows, Linux, etc) podem ser abertas em outra plataforma sem requerer tratamento especial.

Recuperação de dados

- A função *load* recupera os dados armazenados em um arquivo, carregando-os no espaço de trabalho do **MATLAB**. Sua sintaxe mais geral, na forma de comando, é

```
load filename lista de variáveis
```

- O arquivo de nome *filename.mat* é aberto e os conteúdos das variáveis especificadas recuperado.
- Se o nome do arquivo for omitido, o arquivo procurado é o *matlab.mat*.
- Se a lista de variáveis for omitida, todas as variáveis do arquivo serão carregadas no espaço de trabalho.
- Se houver variáveis de mesmo nome em uso no espaço de trabalho, o novo valor, recuperado do arquivo, substitui o valor anterior.

Recuperação de dados

- Para que uma variável existente no espaço de trabalho não tenha seu conteúdo perdido quando carregamos o conteúdo de um arquivo, podemos redirecionar a saída de dados para uma estrutura. Exemplo:

```
>> var1 = 10; var2 = 20;
>> save fi le var1 var2
>> vnew=load('fi le','var1', 'var2')
vnew =
  var2: 20
  var1: 10

>> disp(class(vnew))      >> disp(vnew.var1)      >> disp(vnew.var2)
struct                    10                    20
```

- Mais informações sobre *load* no *help on line*.

Algumas funções úteis

- Para saber se um arquivo existe podemos usar a função

```
exist('matlab.mat','file')
```

que retorna **0** se o arquivo não existe e **2** se o arquivo existe.

- A função *exist* possui uma variedade de saídas e opções. Veja o *help on line* para maiores detalhes.

- O comando

```
whos -file matlab.mat
```

retorna a saída padrão do comando *whos*, mas referindo-se ao conteúdo do arquivo *matlab.mat*.

- *delete filename.ext* remove o arquivo *filename.ext* do disco.

Formatos de arquivos

- Já vimos que o **MATLAB** trabalha com arquivos no seu formato original (arquivos `.mat`) e formato texto padrão (ASCII).
- O **MATLAB** aceita uma variedade de formatos padrão da indústria. Alguns são restritos apenas à leitura e outros à escrita.
- Imagens que aparecem na janela *Figure*:
 - podem ser salvas no formato padrão do **MATLAB** (`.fig`) usando a opção **Save** do menu **File**.
 - podem ser exportadas usando um dos formatos disponíveis no submenu **Export** do menu **File**.
 - alternativamente podemos usar o comando `saveas` na janela *Command* (veja *help on line*).
- O comando `help fileformats` retorna uma lista completa dos formatos de arquivo que o **MATLAB** aceita.

Funções de baixo nível para I/O

Consulte Help → **MATLAB** Help → **MATLAB** → Development Environment → Importing and Exporting Data → Using Low-level I/O Functions

- Há uma grande variedade de funções para a leitura e escrita de dados em arquivos binários ou ASCII.
- A maioria dessas funções é similar a funções equivalentes em ANSI C.

Operações em diretórios

Consulte Help → **MATLAB** Help → **MATLAB** → Workspace, Search Path and File Operations → File Operations

- A janela *Current Directory* é a forma mais fácil de se abrir arquivos, percorrer diretórios, e verificar várias propriedades de arquivos.
- Entretanto, é possível fazer o mesmo usando a janela de comandos. Essa era a forma padrão de se operar em diretórios antes do advento da janela *Current Directory*. Essas funções são úteis em funções que manipulem arquivos e diretórios.
- Operações simples são *cd*, *pwd*, *ls*, *dir*, *copyfile*, *delete*.
- Funções que combinam operações em cadeias de caracteres (strings) e diretórios são: *path*, *addpath*, *matlabroot*, etc.