

Universidade Federal do Paraná
Programa de Pós-Graduação em Geologia

GEOL7048: Tópicos Especiais em Geologia Exploratória II

Métodos semiquantitativos

Saulo P. Oliveira

Departamento de Matemática, Universidade Federal do Paraná



Aula 6

- Manipulação de arquivos
- Gráficos bidimensionais

Manipulação de arquivos



Até agora vimos somente a leitura/gravação de arquivos .m e de figuras:

```
>> x = linspace(0,2*pi,1000);  
>> y = sin(x);  
>> plot(x,y)  
>> print -dpng grafico.png
```

Até agora vimos somente a leitura/gravação de arquivos .m e de figuras:

```
>> x = linspace(0,2*pi,1000);  
>> y = sin(x);  
>> plot(x,y)  
>> print -dpng grafico.png
```

Hoje veremos comandos que leem/gravam as **variáveis** em arquivos.

Manipulação de arquivos



Até agora vimos somente a leitura/gravação de arquivos .m e de figuras:

```
>> x = linspace(0,2*pi,1000);  
>> y = sin(x);  
>> plot(x,y)  
>> print -dpng grafico.png
```

Hoje veremos comandos que leem/gravam as **variáveis** em arquivos. Começamos pelo par load/save:

```
>> save aula.mat  
>> exit
```

Até agora vimos somente a leitura/gravação de arquivos .m e de figuras:

```
>> x = linspace(0,2*pi,1000);  
>> y = sin(x);  
>> plot(x,y)  
>> print -dpng grafico.png
```

Hoje veremos comandos que leem/gravam as **variáveis** em arquivos. Começamos pelo par load/save:

```
>> save aula.mat  
>> exit  
>> x  
>>load aula.mat
```

Até agora vimos somente a leitura/gravação de arquivos .m e de figuras:

```
>> x = linspace(0,2*pi,1000);  
>> y = sin(x);  
>> plot(x,y)  
>> print -dpng grafico.png
```

Hoje veremos comandos que leem/gravam as **variáveis** em arquivos. Começamos pelo par load/save:

```
>> save aula.mat  
>> exit  
>> x  
>>load aula.mat
```

Podemos usar estes comandos para ler/gravar uma única variável, em formato binário ou ASCII.

Manipulação de arquivos



O próximo par de comandos é o par `dlmread/dlmwrite`, que permite trabalhar, por exemplo, com arquivos CSV:

```
>> y = sin(x);  
>> A = [x',y'];  
>> dlmwrite('xy.csv',A);
```

Manipulação de arquivos



O próximo par de comandos é o par `dlmread/dlmwrite`, que permite trabalhar, por exemplo, com arquivos CSV:

```
>> y = sin(x);  
>> A = [x',y'];  
>> dlmwrite('xy.csv',A);  
>> dlmwrite('xy.csv',A,'delimiter','\t','precision',10);
```

Manipulação de arquivos



O próximo par de comandos é o par `dlmread/dlmwrite`, que permite trabalhar, por exemplo, com arquivos CSV:

```
>> y = sin(x);  
>> A = [x',y'];  
>> dlmwrite('xy.csv',A);  
>> dlmwrite('xy.csv',A,'delimiter','\t','precision',10);
```

OBS: o comando `format` altera o formato de saída na tela

Manipulação de arquivos



O próximo par de comandos é o par `dlmread/dlmwrite`, que permite trabalhar, por exemplo, com arquivos CSV:

```
>> y = sin(x);  
>> A = [x',y'];  
>> dlmwrite('xy.csv',A);  
>> dlmwrite('xy.csv',A,'delimiter','\t','precision',10);
```

OBS: o comando `format` altera o formato de saída na tela

Para incluir um cabeçalho:

```
>> dlmwrite('xy.csv','xy');  
>> dlmwrite('xy.csv',A,'-append');
```

Manipulação de arquivos



O próximo par de comandos é o par `dlmread/dlmwrite`, que permite trabalhar, por exemplo, com arquivos CSV:

```
>> y = sin(x);  
>> A = [x',y'];  
>> dlmwrite('xy.csv',A);  
>> dlmwrite('xy.csv',A,'delimiter','\t','precision',10);
```

OBS: o comando `format` altera o formato de saída na tela

Para incluir um cabeçalho:

```
>> dlmwrite('xy.csv','xy');  
>> dlmwrite('xy.csv',A,'-append');
```

Para ler um arquivo:

```
>> x2=dlmread('xy.csv');
```

Manipulação de arquivos



O próximo par de comandos é o par `dlmread/dlmwrite`, que permite trabalhar, por exemplo, com arquivos CSV:

```
>> y = sin(x);  
>> A = [x',y'];  
>> dlmwrite('xy.csv',A);  
>> dlmwrite('xy.csv',A,'delimiter','\t','precision',10);
```

OBS: o comando `format` altera o formato de saída na tela

Para incluir um cabeçalho:

```
>> dlmwrite('xy.csv','xy');  
>> dlmwrite('xy.csv',A,'-append');
```

Para ler um arquivo:

```
>> x2=dlmread('xy.csv');  
>> x2=dlmread('xy.csv',' ',',',1,0); (exclui a primeira linha)
```

Gráficos bidimensionais



Para entender alguns componentes gráficos, vamos inserir dois gráficos em uma figura:

```
>> x=linspace(0,2*pi,1000);  
>> subplot(311)  
>> plot(x,sin(x),x,0.5+0*x)
```

Gráficos bidimensionais



Para entender alguns componentes gráficos, vamos inserir dois gráficos em uma figura:

```
>> x=linspace(0,2*pi,1000);  
>> subplot(311)  
>> plot(x,sin(x),x,0.5+0*x)  
>> subplot(313), plot(x,cos(x))
```

Gráficos bidimensionais



Para entender alguns componentes gráficos, vamos inserir dois gráficos em uma figura:

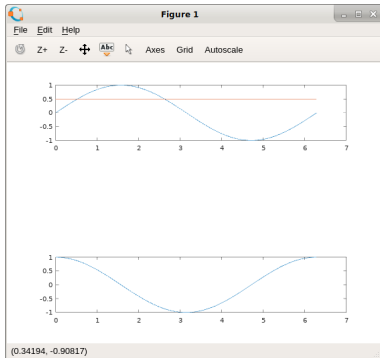
```
>> x=linspace(0,2*pi,1000);  
>> subplot(311)  
>> plot(x,sin(x),x,0.5+0*x)  
>> subplot(313), plot(x,cos(x))
```

Figura (figure): toda a janela →

Primeiro gráfico (axes) →
(possui uma curva e uma reta)

Último gráfico (axes) →

OBS: um ou mais eixos: **axis**



Gráficos bidimensionais



No matlab/octave, a manipulação gráfica segue o paradigma da programação orientada a objetos

No matlab/octave, a manipulação gráfica segue o paradigma da **programação orientada a objetos**

objeto = “super-variável”

classe = “super-tipo”

No matlab/octave, a manipulação gráfica segue o paradigma da **programação orientada a objetos**

objeto = “super-variável”

classe = “super-tipo”

As classes podem formar uma hierarquia: a classe dos gráficos é uma sub-classe da classe das figuras

No matlab/octave, a manipulação gráfica segue o paradigma da **programação orientada a objetos**

objeto = “super-variável”

classe = “super-tipo”

As classes podem formar uma hierarquia: a classe dos gráficos é uma sub-classe da classe das figuras

Um objeto pode ter múltiplos atributos (ao contrário das variáveis, que possuem um só).

No matlab/octave, a manipulação gráfica segue o paradigma da **programação orientada a objetos**

objeto = “super-variável”

classe = “super-tipo”

As classes podem formar uma hierarquia: a classe dos gráficos é uma sub-classe da classe das figuras

Um objeto pode ter múltiplos atributos (ao contrário das variáveis, que possuem um só).

Ojetos mais importantes: figuras, gráficos (axes), elementos gráficos e subelementos gráficos

No matlab/octave, a manipulação gráfica segue o paradigma da **programação orientada a objetos**

objeto = “super-variável”

classe = “super-tipo”

As classes podem formar uma hierarquia: a classe dos gráficos é uma sub-classe da classe das figuras

Um objeto pode ter múltiplos atributos (ao contrário das variáveis, que possuem um só).

Ojetos mais importantes: figuras, gráficos (axes), elementos gráficos e subelementos gráficos

Elementos gráficos: curvas, superfícies, mapas, pontos, textos, ...

Subelementos gráficos: títulos, eixos (axis), legendas,...

Seleção do endereço do objeto atual: gcf, gca, gco

Gráficos bidimensionais



Seleção do endereço do objeto atual: `gcf`, `gca`, `gco`

Para visualizar/editar atributos de um objeto: `get/set`

Seleção do endereço do objeto atual: `gcf`, `gca`, `gco`

Para visualizar/editar atributos de um objeto: `get/set`

Exemplo: clique sobre uma das curvas e digite

```
>> set(gca,'linewidth',2);
```

Seleção do endereço do objeto atual: `gcf`, `gca`, `gco`

Para visualizar/editar atributos de um objeto: `get/set`

Exemplo: clique sobre uma das curvas e digite

```
>> set(gca,'linewidth',2);
```

clique de novo sobre a mesma curva e digite

```
>> set(gco,'linewidth',2);
```

Seleção do endereço do objeto atual: `gcf`, `gca`, `gco`

Para visualizar/editar atributos de um objeto: `get/set`

Exemplo: clique sobre uma das curvas e digite

```
>> set(gca,'linewidth',2);
```

clique de novo sobre a mesma curva e digite

```
>> set(gco,'linewidth',2);
```

Comandos úteis que dispensam o `set`: `axis`, `xlim`, `xlabel`, `title`