

مقدمه ای بر برنامه نویسی با MATLAB

سعید سهیلی
استادیار گروه مکانیک
دانشگاه آزاد اسلامی مشهد

بسمہ اربعہ

فصل ششم

خصوصیات پیشرفته توابع تعریف شده توسط کاربر

Function Functions

- توابع `function` توابعی هستند که توابع دیگر را از طریق ورودی خود دریافت می کنند.

- برای مثال یک تابع `function` به نام `fzero` وجود دارد که صفرهای تابعی که به آن فرستاده می شود را محاسبه می کند.

```
» fzero('cos',[0 pi])
```

```
ans =
```

```
1.5708
```

```
» fzero('exp(x)-2',[0 1])
```

```
ans =
```

```
0.69311
```

- توابع `function` با استفاده از دو تابع خاص عمل می کنند.
- تابع `eval` رشته کاراکتری ورودی را مانند زمانی که در پنجره فرمان تایپ می شود، ارزیابی می کند.
- تابع `feval` مقدار یک تابع با نام را به ازای ورودی مشخص، محاسبه می کند.

```
eval(string)
```

```
» x = eval('sin(pi/4)')  
x =  
0.7071
```

- تابع `feval` یک تابع با نام را که به وسیله یک `m-file` ایجاد شده را در یک مقدار ورودی خاص برآورد می کند.

```
feval(fun,value)
```

```
» x = feval('sin',pi/4)  
x =  
0.7071
```

- برخی از توابع رایج function در جدول زیر آمده اند

Table 7-1 Common MATLAB Function Functions

Function Name	Description
fminbnd	Minimizes a function of one variable.
fzero	Finds a zero of a function of one variable.
quad	Numerically integrates a function.
ezplot	Easy to use function plotter.
fplot	Plots a function by name.

مثال) ایجاد یک تابع function

- تابع function ایجاد کنید که نمودار هر تابع matlab یک متغیره ای را در بین محدوده مشخص شده رسم نماید.
 - این تابع دارای دو آرگومان ورودی است. نام تابع و یک بردار ۲ عنصری شامل محدوده ترسیم.
1. بیان مساله

- تابعی ایجاد کنید که هر گونه تابع matlab یک متغیره را بین دو محدوده مشخص شده رسم نماید.

2. تعریف ورودی ها و خروجی ها

- ورودی یک) رشته کاراکتر شامل نام تابع
- ورودی دو) آرایه ای شامل مقادیر ابتدایی و انتهایی
- خروجی، نمودار تابع مشخص شده می باشد.

4. طراحی الگوریتم:

```
Check for a legal number of arguments  
Check that the second argument has two elements  
Calculate the value of the function between the  
start and stop points  
Plot and label the function
```

- ابتدا مجاز بودن تعداد آرگومان ها بررسی می شود.
- آرگومان دوم باید دارای دو عنصر باشد
- محاسبه مقدار تابع بین دو نقطه ابتدا و انتها
- رسم نمودار

- جزئیات کامل تر شبه کد

```
n_steps ← 100
step_size ← (xlim(2) - xlim(1)) / n_steps
x ← xlim(1):step_size:xlim(2)
y ← feval(fun,x)
plot(x,y)
title(['\bfPlot of function ' fun '(x)'])
xlabel('\bfx')
ylabel(['\bf' fun '(x)'])
```

4. تبدیل الگوریتم به عبارات matlab

• آزمایش برنامه

» `quickplot('sin')`

??? Error using ==> quickplot

Not enough input arguments.

» `quickplot('sin',[-2*pi 2*pi],3)`

??? Error using ==> quickplot

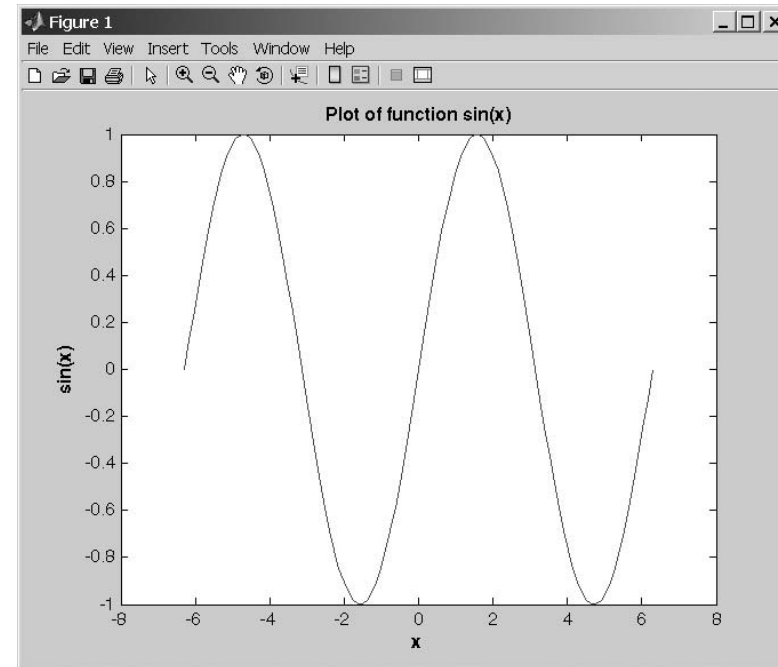
Too many input arguments.

» `quickplot('sin',-2*pi)`

??? Error using ==> quickplot

Incorrect number of elements in xlim.

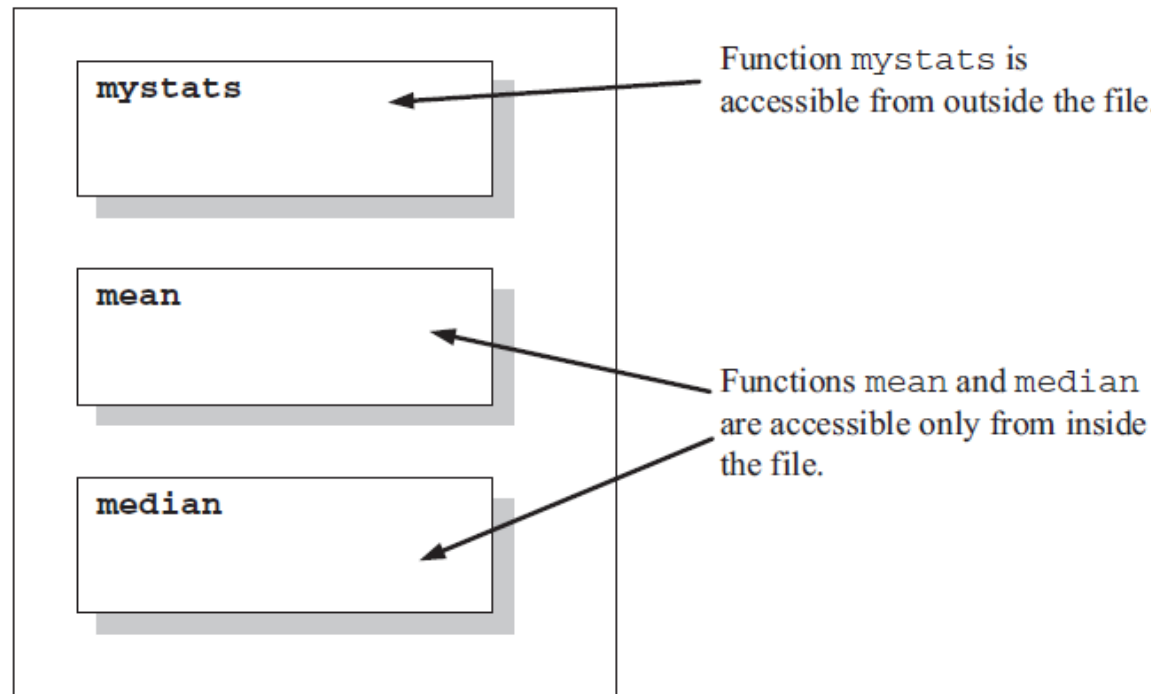
» `quickplot('sin',[-2*pi 2*pi])`



زیر توابع و توابع خصوصی

- امکان قرار دادن بیش از یک تابع درون یک فایل وجود دارد.
- زیر توابع فقط توسط توابع درون همان فایل قابل دسترس هستند.
- در مثال زیر تابع `mystats` تابع اصلی و دو تابع `mean`, `median` زیر تابع هستند که فقط می توان توسط توابع درون فایل اصلی فراخوانی نمود.

File `mystats.m`



```
function [avg, med] = mystats(u)  
%MYSTATS Find mean and median with internal  
functions.  
% Function MYSTATS calculates the average and median  
% of a data set using subfunctions.  
n = length(u);  
avg = mean(u,n);  
med = median(u,n);
```

```
function a = mean(v,n)  
% Subfunction to calculate average.  
a = sum(v)/n;
```

```
function m = median(v,n)  
% Subfunction to calculate median.  
w = sort(v);  
if rem(n,2) == 1  
m = w((n+1)/2);  
else  
m = (w(n/2)+w(n/2+1))/2;  
end
```

توابع خصوصی

- توابعی هستند که درون زیر شاخه ای با نام `private` واقع شده اند.
- این توابع تنها توسط توابع درون شاخه اصلی قابل رویت هستند.
- از توابع خصوصی برای پنهان نمودن توابع با کاربرد ویژه استفاده می شود.

ترتیب اجرای توابع

- در برنامه های بزرگ امکان وجود توابع با نام های مشابه وجود دارد.
- روال زیر ترتیب فراخوانی توابع توسط matlab را نشان می دهد.

1. وجود زیر تابع با اسم معین شده چک می شود
2. وجود تابع در شاخه private را چک می کند
3. در شاخه جاری اسم تابع را دنبال می کند
4. در مسیرهای معرفی شده (matlab path) دنبال تابع می گردد

Function Handles دستگیره تابع

- function handle یک نوع داده matlab است که اطلاعاتی را برای ارجاع دادن به یک تابع نگه می دارد
- وقتی دستگیره ایجاد شد از آن می توان برای اجرای تابع در هر زمانی استفاده نمود.
- دستگیره تابع در واقع دسترسی به تابع را از طریق یک متغیر فراهم می کند.
- دستگیره تابع به دو شیوه ایجاد می شود.
 - اپراتور @
 - تابع str2func
- برای استفاده از اپراتور @ کافی است اپراتور قبل از نام تابع به کار برده شود.
- در روش دوم اسم تابع به صورت رشته فراخوانی می شود

- تابع `my_func` به صورت زیر تعریف شده است

```
function res = my_func(x)
res = x.^2 - 2*x + 1;
```

- دو روش بیان شده به صورت زیر استفاده می شوند.

```
hndl = @my_func
hndl = str2func('my_func');
```

- با قرار دادن متغیرهای ورودی در دستگیره تابع تعریف شده، می توان تابع را اجرا نمود.

- نتیجه مشابه وقتی است که از خود تابع استفاده می کنیم

```
>> hndl = @my_func
hndl =
    @my_func
>> hndl(4)
ans =
     9
>> my_func(4)
ans =
     9
```

- از تابع `feval` نیز می توان برای مقداردهی استفاده نمود و تابع `func2str` نام تابع را بر می گرداند.

```
» feval(hndl,4)
```

```
ans =  
      9
```

```
» func2str(hndl)
```

```
ans =  
my_func
```

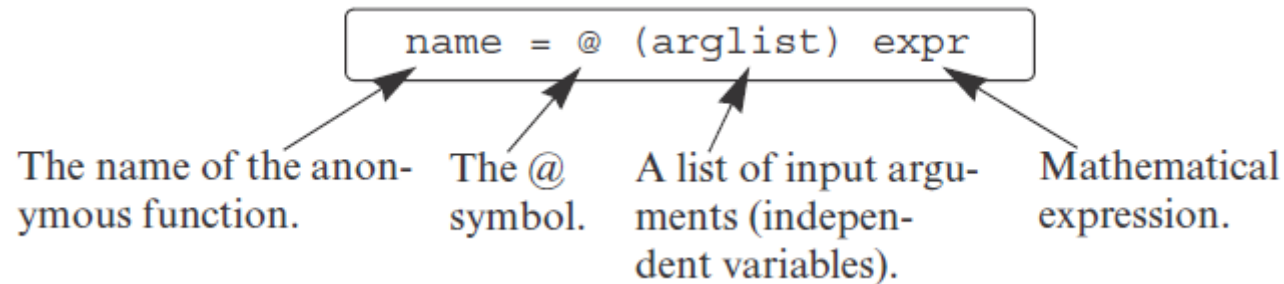
- به عنوان مثال برنامه زیر یک `function handle` را دریافت و آن را در بازه تعیین شده رسم می کند.

```
plotfunc(@sin,[-2*pi:pi/10:2*pi])
```

- دستگیره تابع به طور خاص برای حل معادلات دیفرانسیل کاربرد دارد

توابع بی نام

- توابع بدون نامی هستند که یک دستگیره تابع ایجاد می کند تا از آن برای برآورد مقدار استفاده نمود.



$$f(x) = x^2 - 2x - 2 ;$$

```
myfunc = @(x) x.^2 - 2*x - 2
```

```
>> myfunc(2)  
ans =  
    -2
```

The function $f(x) = \frac{e^{x^2}}{\sqrt{x^2+5}}$ can be defined (in the Command Window) as an anonymous function for x as a scalar by:

```
>> FA = @ (x) exp(x^2)/sqrt(x^2+5)
FA =
    @(x) exp(x^2)/sqrt(x^2+5)
```

```
>> FA(2)
ans =
    18.1994
>> z = FA(3)
z =
    2.1656e+003
```

The function $f(x, y) = x^2 - 4xy + y^2$ can be defined as an anonymous function by:

```
>> HA = @ (x,y) 2*x^2 - 4*x*y + y^2  
HA =  
    @(x,y) 2*x^2-4*x*y+y^2
```

```
>> HA(2,3)  
ans =  
    -7
```

توابع تکراری

- یک تابع وقتی تکراری است که خودش را فراخوانی کند
- به عنوان مثال تابع فاکتوریل در نظر گرفته می شود.

$$n! = \begin{cases} 1 & n = 0 \\ n \times (n-1) \times (n-2) \times \dots \times 2 \times 1 & n > 0 \end{cases}$$

$$n! = \begin{cases} 1 & n = 0 \\ (n-1)! & n > 0 \end{cases}$$

```

function result = fact(n)
%FACT Calculate the factorial function
% Function FACT calculates the factorial function
% by recursively calling itself.
% Define variables:
% n -- Non-negative integer input
%
% Record of revisions:
% Date Engineer Description of change
% =====
=====
% 07/07/10 S. J. Chapman Original code
% Check for a legal number of input arguments.
msg = nargchk(1,1,nargin);
error(msg);
% Calculate function
if n == 0
result = 1;
else
result = n * fact(n-1);
end

```

رسم توابع

- برای رسم مستقیم توابع از دو دستور `ezplot` و `fplot` می توان استفاده نمود.

```
ezplot(fun);  
ezplot(fun, [xmin xmax]);  
ezplot(fun, [xmin xmax], figure);
```

- `fun` می تواند دستگیره تابع، نام یک تابع `m-file` و یا یک رشته حرفی باشد.

```
ezplot('sin(x)/x',[-4*pi 4*pi]);  
title('Plot of sin x / x');  
grid on;
```

