

Matlab 编写函数

函数编写规则

1、语法以及规则

a) 格式:

```
function [outputArg1,outputArg2] = untitled2(inputArg1,inputArg2)
    outputArg1 = inputArg1;
    outputArg2 = inputArg2;
end
```

解释: function代表函数 [outputArg1,outputArg2]表示输出。语法, 输出后面接等于号 untitled2为未命名的函数2, 也就是函数名, 可以自行更改。

inputArg1,inputArg2代表输入变量。

2、简单函数的编写:

a) 输入变量分类

i. 单个变量(数): 非常容易。不做过多阐述。

ii. 向量(矩阵):

1. 例子: 函数用于输出矩阵第2行2列的数与第2行2列的数的"+-×÷"运算

```
clc;
clear;
A=magic(3)
B=floor(20*rand(3,3))
fun_a(A,B,3)
function result=fun_a(A,B,way)
switch way
case 1
    result=A(2,2)+B(2,2);
case 2
    result=A(2,2)-B(2,2);
case 3
    result=A(2,2)*B(2,2);
case 4
    result=A(2,2)/B(2,2);
otherwise
    result=-1;
end
end
```

可见, 定义函数的输入变量时, 不需要定义类型(单个数还是矩阵或者向量)。和c语言不同。但要注意, 这样使用函数时要自我匹配。例如, 把A, B的规模都改成1×1, 这样就会报错。

```
A = 1  
B = 10
```

位置 1 处的索引超出数组边界(不能超出 1)。

出错 `practice_fun>fun_a (line 14)`
`result=A(2,2)*B(2,2);`

iii. 输入变量为函数

1. 函数句柄

由于想完成某种功能，需要输入表达式。

例如，求曲线在某点的曲率半径，这时候就需要输入函数表达式。

输入 $x(t)$ 和 $y(t)$ 以及点对应 t 的值，即可求解出曲率半径。

程序：

```
clc;  
clear;  
fxt=@(t)t^2;  
fyt=@(t)t^3;  
t=1;  
calc_qlbj(fxt,fyt,t)  
  
function z = calc_qlbj(x, y, t)  
dt=1e-7; %定义微分量  
dfx=@(t)(x(t+dt)-x(t))/dt; %x 对 t 的导数  
dfy=@(t)(y(t+dt)-y(t))/dt; %y 对 t 的导数  
dx=x(t+dt)-x(t); %dx 微分量  
dy=y(t+dt)-y(t); %dy 微分量  
ds=sqrt(dx^2+dy^2)%求解 ds  
  
theta=@(a)atan(dfy(a)/dfx(a)); %切线角对于 t 的函数  
d1=t+dt;  
d2=t;  
theta(d1)  
theta(d2)  
dtheta=theta(d1)-theta(d2) %手动求导（微分量）  
  
z=vpa(abs(ds/dtheta)); %曲率半径定义  
end
```

使用了函数句柄，由于表达式的需求。但这样做，在求导时需要进行手动求导（利用定义）。在函数中，还出现了嵌套函数。dfx时 x 对 t 的导数。

解释：函数中 x 就是一个函数，比如 $x(12)$ 可以得到144。

dfx是一个嵌套函数，dfx(12)可以得到24。dfx中先定义自变量为 t ，之后表达式定义为有关 $x(t)$ 函数。在数学中可以理解为 $f(g(t))$ 。

注意：不能用diff对这个 $x(t)$ 求导，因为这不是符号函数。

值得注意的是：编写函数时还是要注意匹配，如果输入与函数中的表达

式不符，还是会引起报错。

数组索引必须为正整数或逻辑值。

出错 `qlbjjisuan>calc_qlbj (line 15)`
`dx=x(t+dt)-x(t);`

这个错误出现在，把输入的fxt改为了任意一个向量导致的。

iv. 输入符号(函数)

符号函数

1、基本用法: `syms x y ... / syms x(t)`

解释: `x`, `y` 为单独的符号变量, `1×1sym`, 而 `x(t)` 为符号函数, `1×1symfun`

2、优点: 符号对象是一种数据结构, 用来存储代表符号对象的字符串的复杂数据结构。这样有很多遍历的用处: 求导, 积分等等。

3、用法举例:

```
syms x(t)
x(t)=t^3+t^2+t+1;
y(t)=diff(x(t),t) %求导数
g(t)=int(x(t),t) %求不定积分
y(2) %求值
g(6)
fplot(x(t)) %绘图
```

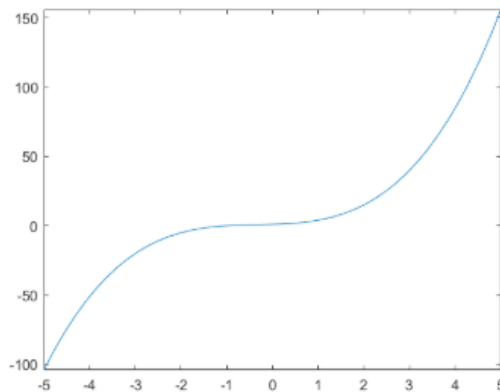
$$y(t) = 3t^2 + 2t + 1$$

$$g(t) =$$

$$\frac{t^4}{4} + \frac{t^3}{3} + \frac{t^2}{2} + t$$

$$\text{ans} = 17$$

$$\text{ans} = 420$$



解释: 定义 `x(t)` 为 `symfun`, 则求导时, `y(t)` 代表 `x(t)` 的导数是关于 `t` 的一个方程, 而 `y(t)` 就是这个方程, 为 `symfun`。但如果 `y` 后不加括号, 则表示的是 `y` 是一个 `sym` 变量, `y` 和 `t` 这个 `sym` 变量是有一定的变量关系, 所以在求值时候, 不能使用 `y(2...)`, 因为编译器默认为: `y` 是一个 `m×n` 的 `sym` 变量 (本例中为 `1×1`), 所以, `y(2...)` 会越界导致报错。

索引超出数组元素的数目(1)。

出错 `sym/subsref (line 900)`
`R_tilde = builtin('subsref',L_tilde,I`

上述思想在编写函数时，传递的变量若为符号函数，则不要带有括号（例如： $x(t)$ ），这样在传递参数时，会导致 $y=x(t)$ ，这时， y 就是一个 `sym` 类型；若要 y 为 `symfun` 类型，则不要带有括号。

例子：求解曲率半径

```
clear;
clc;
syms x(t) y(t) t           %定义符号变量和符号函数
x(t)=t^2;
y(t)=t^3;
calc_qlbj(x,y,1);           %调用求解曲率半径函数
                             %注意不要在 x 后面带括号，这里的 x 就是一个
                             %Symfun 数据类型。
```

```
function z=calc_qlbj(x,y,p)
%计算 ds
syms dx(t) dy(t) t
dx(t)=diff(x);             %直接求导
dy(t)=diff(y);
dt=1e-12;                   %定义微分量
class(x)                    %查看 x 类型，为 symfun
ds=vpa(sqrt((x(p+dt)-x(p))^2+(y(p+dt)-y(p))^2));
                             %近似，求解 ds
alpha(t)=atan(dy(t)/dx(t)); %符号函数，切线仰角函数
dalphi=vpa(alpha(p+dt)-alpha(p));
                             %求解 dalphi
z=vpa(abs(ds/dalphi))        %根据定义计算曲率半径
vpa(13/6*sqrt(13))          %与理论值进行对比
end
```

结果如图：

$z = 7.8120277635173297077952266842856$

$ans = 7.8120277635053101350916460795194$

上面为求解的值，下面为理论值，误差非常小。

4、复合符号函数

```
syms x(t) z(t)
z(t)=t^2;
f1(t)=x(z(t)) %符合简单
f2(t)=simplify(compose(x,y)) %符合，代表 x(y(t))，并且化简

f1(t) =  $t^6 + t^4 + t^2 + 1$ 
f2(t) =
 $27 t^6 + 54 t^5 + 72 t^4 + 56 t^3 + 34 t^2 + 12 t + 4$ 
```