



苏州大学

SOOCHOW UNIVERSITY

电子信息学院



School of Electronic & Information Engineering

# 第5章 程序设计和M文件

5.1 程序控制

5.2 M文件结构

5.3 函数的使用

5.4 程序的调试

5.5 匿名函数和inline对象

5.6 函数绘图

5.7 数值分析



蘇州大學

SOOCHOW UNIVERSITY

## 5.1 程序控制

### 5.1.1 分支控制语句

#### 1. if 结构

- if 条件1
- 语句段1
- elseif 条件2
- 语句段2
- .....
- else
- 语句段n
- end

例：

$$\begin{cases} x^2 - 1 & x \geq 1 \\ 0 & -1 < x < 1 \\ -x^2 + 1 & x \leq -1 \end{cases}$$

```
>> x=2;  
>> if x>=1  
    y=x.^2-1  
elseif -1<x & x<1  
    y=0*x  
else  
    y=-x.^2+1  
end
```



蘇州大學

SOOCHOW UNIVERSITY

## 2. switch 结构

switch 表达式

case 值1

语句段1

case 值2

语句段2

...

otherwise

语句段n

end

说明:

1、将表达式依次与case后面的值进行比较，满足值的范围就执行相应的语句段，如果都不满足则执行otherwise后面的语句段；

2、表达式只能是标量或字符串；

3、case后面的值可以是标量、字符串或元胞数组，如果是元胞数组则将表达式与元胞数组的所有元素进行比较，只要某个元素与表达式相等，就执行其后的语句段；

4、switch和end必须配对使用。



蘇州大學

SOOCHOW UNIVERSITY

**例：**使用switch结构判断学生成绩的等级，90分以上为优，80～90为良，70～80为中，60～70为及格，60分以下为不及格。

```
>> score=98;  
>> s1=fix(score/10); %取十位数  
>> switch s1  
    case {9,10}  
        s='优'  
    case 8  
        s='良'  
    case 7  
        s='中'  
    case 6  
        s='及格'  
    otherwise  
        s='不及格'  
end
```



## 5.1.2 循环控制语句

### 1. for循环

```
for 循环变量=array  
    循环体  
end
```

说明：array可以是向量也可以是矩阵，循环执行的次数就是array的列数，每次循环中循环变量依次取array的各列并执行循环体，直到array所有列取完。

- *for n=1:5* %循环5次
- *for n=-1:0.1:1* %循环21次
- *for n=linspace(-2\*pi,2\*pi,5)* %循环5次
- *a=eye(2,3); for n=a* %循环3次，n为列向量



蘇州大學

SOOCHOW UNIVERSITY

**例：** 使用for循环将单位矩阵进行转换，转换为对角线上分别是1、2、3、4、5的矩阵。

```
>> x=eye(5);
```

```
>> len=length(x);
```

```
>> for n=1:len
```

```
  x(n,n)=n;
```

```
end
```



蘇州大學

SOOCHOW UNIVERSITY

## 2. while循环

**while** 条件表达式  
循环体

**end**

**例：** 使用while循环计算 $f(x)=\sin(x)$ ， $x<5$ 且每次 $x$ 递增 $\ln(x)$ 。

```
>> x=0.1;
```

```
>> while x<5
```

```
    f=sin(x)
```

```
    x=x+log(x)
```

```
end
```



蘇州大學

SOOCHOW UNIVERSITY

### 3. break和continue语句

#### (1) break语句

- break语句使包含break的最内层for或while循环强制终止，并立即跳出该循环结构，执行end后面的命令，break一般与if语句结合使用。

#### (2) continue语句

- continue语句与break不同的是continue只结束本次for或while循环，而继续进行下次循环，continue一般也与if语句结合使用。





蘇州大學

SOOCHOW UNIVERSITY

**例** 使用for循环将字符串中的数值取出，遇到非数值则跳过。

```
>> str='The result is 100.';
```

```
>> len=length(str);
```

```
>> s=[];
```

```
>> for n=1:len
```

```
    if str(n)>'9'|str(n)<'0'           %非数值时
```

```
        continue
```

```
    end
```

```
    s=[s str(n)];
```

```
end
```



蘇州大學

SOOCHOW UNIVERSITY

## 5. 1. 3 错误控制语句

- **try**
- **语句段1**
- **catch**
- **语句段2**
- **end**



蘇州大學

SOOCHOW UNIVERSITY

## 说明:

- 先试探地执行语句段1，如果出现错误则将错误信息赋给保留的lasterr变量，并放弃语句段1转而执行语句段2中的语句；如果语句段2正确则结束；
- 如果语句段1正确则不执行语句段2就结束；
- 当语句段1和语句段2都错误，则程序出错。
- 当错误控制结构运行结束后，可以调用lasterr函数查询出错信息，调用方法是“[lastmsg,lastid]=lasterr”，其中lastmsg保存出错信息，lastid保存错误类型。



蘇州大學

SOOCHOW UNIVERSITY

- 例 使用错误控制语句查看 $a*b$ 的运算

```
a=[1 3 5];
```

```
b=[1 2 3];
```

```
try
```

```
    c=a*b
```

```
catch
```

```
    c=a.*b
```

```
end
```

```
[lastmsg,lastid]=lasterr
```



蘇州大學

SOOCHOW UNIVERSITY

## 5.1.4 流程控制命令

### 1. return命令

- **return命令**用于提前结束程序的执行，并立即返回到上一级调用函数或等待键盘输入命令，一般用于遇到特殊情况需要立即退出程序或终止键盘方式。
- 应注意当程序进入死循环时，则按 **Ctrl+break**键来终止程序的运行。



蘇州大學

SOOCHOW UNIVERSITY

## 2. keyboard命令

- **keyboard命令**用来使程序暂停运行，等待键盘命令，命令窗口出现“K>>”提示符，当键盘输入“return”后，程序才继续运行。**keyboard命令**可以用来在程序调试或程序执行时



蘇州大學

SOOCHOW UNIVERSITY

- 例 使用keyboard命令输入变量b

```
a=[1 3 5];
```

```
>> keyboard
```

```
K>> b=[1 2 4];
```

%等待键盘输入

```
K>> return
```

%终止键盘输入 dbcont

```
K>> try
```

```
c=a.*b'
```

```
catch
```

```
    c=a.*b
```

```
end
```



蘇州大學

SOOCHOW UNIVERSITY

### 3. input命令

- `r=input('str','s')` %从键盘中输入数据保存到变量r

### 4. disp命令

- `disp`命令是较常用的显示命令，常用来显示字符串型的信息提示。

### 5. pause命令

- `pause`命令用来使程序暂停运行，当用户按任意键才继续执行。常用于程序调试或查看中间结果，也可以用来控制执行的速度。`pause`的命令格式如下：
  - `pause(n)` %暂停n秒





## 5.2 M文件结构

- MATLAB 7.3的程序如果要保存则使用扩展名是“.m”的M文件，M文件有两种即M脚本文件（Script File）和M函数文件（Function File）。

```
1 function c=ex5_9(a,b)
2 %EX5_9 a multiply b is c
3 % c is the product
4 % a is the multiplier
5 % b is the multiplicand
6
7 % copyright 2007-10-01
8
9 - try
10 - c=a.*b';
11 - catch
12 - c=a.*b;
13 - end
```



蘇州大學

SOOCHOW UNIVERSITY

## 5.2.1 M文件的一般结构

### 1. 函数声明行

- 函数声明行是在M函数文件的第一行，只有M函数文件必须有，以“function”引导并指定函数名、输入和输出参数，M脚本文件没有函数声明行。

### 2. H1行

- H1行是帮助文字的第一行。

### 3. 帮助文本

### 4. 程序代码

- 程序代码由MATLAB语句和注释语句构成。



蘇州大學

SOOCHOW UNIVERSITY

## 5. 2. 2 M脚本文件和M函数文件

### 1. M脚本文件

M脚本文件的说明如下：

(1) MATLAB在运行脚本文件时，只是简单地按顺序从文件中读取一条条命令，送到MATLAB命令窗口中去执行；



蘇州大學

SOOCHOW UNIVERSITY

(2) M脚本文件运行产生的变量都驻留在MATLAB的工作空间中，可以很方便地查看变量，在命令窗口中运行的命令都可以使用这些变量；

(3) 脚本文件的命令可以访问工作空间的所有数据，因此要注意避免工作空间和脚本文件中的同名变量相互覆盖，一般在M脚本文件的开头使用“clear”命令清除工作空间的变量。



蘇州大學

SOOCHOW UNIVERSITY

- **例** 在M文件编辑 / 调试器窗口中编写M脚本文件得出10-50范围内的所有素数。
- **%EX5\_8 计算10-50内的素数**
- ***clear***
- ***y1=primes(50);***                   ***%获得50以内的所有素数***
- ***n=length(primes(10));*** ***%获得10以内素数的个数***
- ***y=y1(n+1:end)***



蘇州大學

SOOCHOW UNIVERSITY

在命令窗口中输入：

```
>> ex5_8
```

```
y = 11 13 17 19 23 29 31 37 41 43 47
```

在工作空间中就可以查看到变量y1、n和y，并可以修改和使用这些变量。



蘇州大學

SOOCHOW UNIVERSITY

## 2. M函数文件

M函数文件的说明如下：

- (1) M函数文件中的函数声明行是必不可少的；
- (2) M函数文件在运行过程中产生的变量都存放在函数本身的工作空间中，函数的工作空间是独立的、临时的，随具体的M函数文件调用而产生并随调用结束而删除，在MATLAB运行过程中如果运行多个函数则产生多个临时的函数空间；
- (3) 当文件执行完最后一条命令或遇到“return”命令时就结束函数文件的运行，同时函数工作空间的变量被清除；
- (4) 一个M函数文件至少要定义一个函数。



蘇州大學

SOOCHOW UNIVERSITY

函数声明行的格式如下：

**function [输出参数列表] = 函数名(输入参数列表)**

**说明：**

函数名是函数的名称，保存时最好函数名与文件名一致，当不一致时，MATLAB以文件名为准；

输入参数列表是函数接收的输入参数，多个参数间用“,”分隔；

输出参数列表是函数运算的结果，多个参数间用“,”分隔





蘇州大學

SOOCHOW UNIVERSITY

**例** 将上例的计算行向量乘积的运算使用M函数文件保存。

```
function c=ex5_9(a,b)
```

```
% EX5_9(a,b) returns the product of a and b
```

```
% c is the product
```

```
% a is the multiplier
```

```
% b is the multiplicand
```

```
% copyright 2022-11-01
```

```
try
```

```
    c=a.*b';
```

```
catch
```

```
    c=a.*b;
```

```
end
```

调用该函数：

```
>> z=ex5_9([1 2 3],[4 5 6])
```

```
z =
```

```
    4    10    18
```



蘇州大學

SOOCHOW UNIVERSITY

## 5.3 函数的使用

### 5.3.1 主函数和子函数

#### 1. 主函数

一个M函数文件中可以包含一个或多个函数，主函数是出现在文件最上方的函数，即第一行声明的函数，一个M文件只能有一个主函数，通常主函数名与M函数文件名相同。



蘇州大學

SOOCHOW UNIVERSITY

## 2. 子函数

- (1) 子函数的次序无任何限制；
- (2) 子函数只能被同一文件中的函数（主函数或子函数）调用，不能被其它文件的函数调用；
- (3) 同一文件的主函数和子函数运行时的  
工作空间是相互独立的。



蘇州大學

SOOCHOW UNIVERSITY

## 5.3.2 函数的输入输出参数

### 1. 参数的传递

函数的参数传递是将主调函数中的变量值传给被调函数的输入参数

- (1) 函数参数传递的是数值
- (2) 被调函数的输入参数是存放在函数的工作空间中，与MATLAB的工作空间是独立的，当调用结束时函数的工作空间被清除，输入参数也被清除。



蘇州大學

SOOCHOW UNIVERSITY

## 2. 输入输出参数的个数

### (1) nargin和nargout函数

nargin('fun')            %获取函数fun的输入参数个数

nargout('fun')           %获取函数fun的输出参数个数

- 说明：fun是函数名，可以省略，当nargin和nargout函数在函数体内时fun可省略，在函数外时fun不省略。



蘇州大學

SOOCHOW UNIVERSITY

- **例** 当输入参数个数变化时使用nargin函数绘制不同线型的曲线。

```
function n=ex5_11(s1,s2)
x=0:10;
y=nargin*ones(11,1);
hold on
if nargin ==0
plot(x,y)
elseif nargin == 1
plot(x,y,s1)
else
plot(x,y,[s1,s2])
end
```

在命令窗口中输入调用命令：

```
>> ex5_11
>> ex5_11('r')
>> ex5_11('k','o')
>> nargin('ex5_11')
```



蘇州大學

SOOCHOW UNIVERSITY

## (2) varargin和varargout函数

- varargin和varargout函数将函数调用时实际传递的参数构成元胞数组，通过访问元胞数组中各元素内容来获得输入输出变量。
- **function y = fun(varargin)**  
    %输入参数为varargin的函数fun
- **function varargout = fun(x)**  
    %输出参数为varargout的函数fun



蘇州大學

SOOCHOW UNIVERSITY

- **例** 根据输入参数的个数将上例中参数个数使用 `varargin` 和 `varargout` 函数，绘制不同线型的曲线。

```
function varargout=ex5_12(varargin)
x=0:10;
lin=length(varargin); %取输入参数个数
y=lin*ones(11,1);
hold on
if lin==0
plot(x,y)
elseif lin==1
plot(x,y,varargin{1})
else
plot(x,y,[varargin{1} varargin{2}])
end
varargout{1}=lin
```

在命令窗口中输入调用命令：

```
>> y=ex5_12('y','o')
```

```
varargout =
    {[2]}
```

```
y =
    2
```

```
>> ex5_12('r','o')
```

```
varargout =
    {[2]}
```

```
ans =
    2
```





蘇州大學

SOOCHOW UNIVERSITY

## 5.3.3 局部变量、全局变量和静态变量

### 1. 局部变量

局部变量 (Local Variables) 的作用范围只能在函数内部

### 2. 全局变量

- 全局变量在使用前必须用“global”声明，而且每个要共享全局变量的函数和工作空间，都必须逐个用“global”对该变量加以声明
- 要清除全局变量可以使用clear命令，命令格式如下：
- clear global 变量名    %清除某个全局变量
- clear global            %清除所有的全局变量



蘇州大學

SOOCHOW UNIVERSITY

## 5.3.4 嵌套函数、私有函数和重载函数

### 1. 嵌套函数

- 在MATLAB中一个函数的内部还可以定义一个或多个函数，这种定义在其他函数内部的函数就称为嵌套函数。

### 2. 私有函数

- 私有函数是限制访问权限的函数，私有函数存放在“private”子目录中，只能被其直接父目录的M函数文件所调用。

### 3. 重载函数

- 重载函数是指两个函数使用相同的名称，处理的功能相似，但参数类型或个数不同，重载函数通常放在不同的文件夹下，文件夹名称以“@”开头后面跟一个数据类型名。



蘇州大學

SOOCHOW UNIVERSITY

## 5.3.5 函数的工作过程和P码文件

### 1. 函数的搜索过程

- 当在MATLAB中输入一个函数名时，首先确认不是变量名后，函数搜索的顺序如下：
- 检查是否是本M函数文件内部的子函数；
- 检查是否是“private”目录下的私有函数；
- 检查是否在当前路径中；
- 检查是否在搜索路径中。



蘇州大學

SOOCHOW UNIVERSITY

## 2. P码文件

- P码就是伪代码（Pseudocode），一个M文件第一次被调用时，MATLAB就将其进行编译并生成P码文件存放在内存中，
- 生成的P码文件与原M文件名相同，其扩展名为“.p”，P码文件的保密性好，
- `pcode File1.m,File2.m..... -inplace` %生成p文件
- `dir` % 列出文件夹内容

## 3. 函数的工作空间

- 每一个M函数运行时都有一个内存区，称为函数的工作空间



蘇州大學

SOOCHOW UNIVERSITY

## 5.4 程序的调试

### 5.4.1 直接检测

(1) 对于需要检测的变量可以通过删除语句行末尾的分号，或在程序的适当位置加显示变量值的语句，将结果显示在命令窗口中；

(2) 调试函数时，可以将该函数的第一句函数声明行前加“%”，并给输入参数赋值，就可以以脚本文件的方式来执行该函数，避免多个函数在一起运行时查错不方便；

(3) 在程序的适当位置添加“keyboard”语句，当程序运行至此句会暂停运行，并在命令窗口显示“k>>”提示符，这时就可以在命令窗口查看和修改各变量的内容。



蘇州大學

SOOCHOW UNIVERSITY

## 5. 4. 2 M文件编辑 / 调试器窗口

### 1. “Debug”菜单

- (1) Step (F10)
- (2) Step in (F11) 和Step out (shift+F11)
- (3) Run/Continue (F5)
- (4) Set/Clear Breakpoints (F12)
- (5) Set/Modify Conditional Breakpoint...
- (6) Stop if Errors/Warnings...
- (7) Exit Debug Mode



蘇州大學

SOOCHOW UNIVERSITY

## 5.5 匿名函数、inline对象

### 5.5.1 匿名函数

- 匿名函数是面向命令行代码的函数形式，通常只有一句很简单的语句。
- `fhandle=@(arg1,arg2,...)(expr)` %创建匿名函数
- 说明：`fhandle`是函数句柄；`arg1,arg2,...`是参数列表，也可以省略；`expr`是函数表达式。
- 例：
  - `>> fhnd1=@(x)(1+exp(-x));` %创建匿名函数
  - `>> rf1=fhnd1(2)` %调用匿名函数
  - `rf1 =`
  - `1.1353`



创建inline对象就是使用inline函数将字符串转换成inline对象。

• `inline_fun=inline('string',arg1,arg2,...)`      %创建inline对象

例:

`>> f=inline('sin(x)*exp(-z*x)','x','z')`      %创建inline对象f

`>> class(f)`      %查看的类型

`ans =`

`inline`

`>> y1=f(1,2)`      %调用inline对象f

`y1 =`

`0.1139`

`>> y2=feval(f,1,2)`      %执行inline对象f

`y2 =`

`0.1139`





蘇州大學

SOOCHOW UNIVERSITY

## 5.5.3 函数句柄

- 函数句柄 (Function\_Handle) 包含了函数的路径、函数名、类型以及可能存在的重载方法，即函数是否为内部函数、M或P文件、子函数、私有函数等。

### 1. 创建函数句柄

- `fhandle=@fun`                      %创建函数句柄

### 2. 函数句柄的调用

- `[y1,y2,...]=fhandle(arg1,arg2...) %调用函数句柄`  
`fhandle`
- `[y1,y2,...]=feval(fhandle,arg1,arg2...)`
- `[y1,y2,...]=feval('fun',arg1,arg2...)`



蘇州大學

SOOCHOW UNIVERSITY

## 5.6 函数绘图

### 5.6.1 fplot命令

- fplot命令可以绘制函数的曲线。
- fplot(fun,limits,tol,Linespec) %绘制函数fun的曲线
- fplot(fun,limits,n)

说明：fun是函数句柄或函数名；limits是自变量的取值范围[xmin xmax]或[xmin xmax ymin ymax]；tol是相对误差度，默认为2e-3，可省略；Linespec是线型，与plot命令的设置相同，可省略；n是绘制的点数，当 $n \geq 1$ 时至少绘制n+1个点。

例：>> fhnd=@(x)(exp(-x).\*sin(x));

>> fplot(fhnd,[0 20],'b-.\*')



蘇州大學

SOOCHOW UNIVERSITY

## 5.6.2 ezplot命令

- ezplot命令也是用于绘制函数在某一自变量区域内的图形，ezplot命令格式如下：
- ezplot(fun,[min,max]) %绘制函数fun的曲线
- ezplot(x,y,[tmin,tmax])

说明：fun是函数句柄或函数名；[min,max]是自变量的范围，可省略，省略时默认为 $[-2\pi, 2\pi]$ ，当fun是二元函数，则范围是[xmin xmax ymin ymax]；x和y是指x(t)和y(t)；[tmin,tmax]是t的范围，可省略，省略时范围是 $[0, 2\pi]$ 。

例：>> ezplot('x^2+y^2-1',[-3,3,-3,3])



蘇州大學

SOOCHOW UNIVERSITY

## 5.7 数值分析

### 5.7.1 求最小值和过零点

#### 1. 一元函数的最小值

- `fminbnd`函数可以获得一元函数在给定区间内的最小值。
- `x=fminbnd(fun,x1,x2)` %寻找最小值

**例：**使用`fminbnd`函数获得 $\sin(x)$ 和匿名函数 $f1(x)=x^2-5x$ 的在 $[0,10]$ 范围内的最小值

- `>> x=fminbnd(@sin,0,10)` %计算正弦函数最小值的横坐标
- `>> fhndl=@(x)(x^2-5*x);`
- `>> [x,y1]=fminbnd(fhndl,0,10)` %计算匿名函数的最小值
- `>> fplot(fhndl,[0,10])`



蘇州大學

SOOCHOW UNIVERSITY

## 2. 多元函数的最小值

- `fminsearch`函数可以获得多元函数的最小值。
- `[x,fval]=fminsearch(fun,x0)` %寻找最小值

## 3. 一元函数的过零点

- `x=fzero(fun,x0)` %获得`fun`在`x0`附近的过零点
- 一元函数`f(x)`的过零点求解可以使用`fzero`函数来实现。



蘇州大學

SOOCHOW UNIVERSITY

## 5.7.2 数值积分

- 一元函数的数值积分，采用自适应的Simpson方法。
- `q=quad(fun,a,b,tol,trace)`

例：

```
>> fhnd=@(x)(exp(-x.^2));
```

```
>> q1=quad(fhnd,-1,1,2e-3,1) %数值积分并跟踪展示迭代过程
```

```
9 -1.00000000000 5.43160000e-001 0.3198710950
```

```
11 -0.4568400000 9.13680000e-001 0.8538774475
```

```
13 0.4568400000 5.43160000e-001 0.3198710950
```

```
q1 =
```

```
1.4936
```



蘇州大學

SOOCHOW UNIVERSITY

### 5.7.3 微分方程组数值求解

- MATLAB 7.3为解常微分方程提供了7种数值求解的方法，包括ode45、ode23、ode113、ode15s、ode23s、ode23t和ode23tb 函数，各函数的命令格式如下：

- `[t,y]=ode45(fun,ts,y0,options)`      %解常微分方程

说明：fun是函数句柄或函数名；ts是自变量范围，可以是范围[t0,tf]，也可以是向量[t0,.....tf]；y0是初始值，y0应该是和y具有同样长度的列向量；options是设定微分方程解法器的参数，可省略，可以由odeset函数来获得。



蘇州大學

SOOCHOW UNIVERSITY

- **例** 使用ode45函数解微分方程。

- 先将二阶微分方程式  $\frac{d^2y(t)}{dt^2} + 1.414 \frac{dy(t)}{dt} + y(t) = 1$  变换成一阶微分方程组：

- 创建M函数文件ex5\_25.m，虽然t参数不用，但微分方程的函数必须有时间t变量：

- *function ex5\_25()*

- *ts=[0,20];*                    %自变量范围

- *y0=[0;0];*                    %初始条件

- *[t,y]=ode45(@ex5\_25\_1,ts,y0);*

- *plot(t,y(:,1),'r',t,y(:,2),'g--')*

- *function yp=ex5\_25\_1(t,y)*

- *%EX5\_25 使用ode45函数解微分方程*

- *yp=[y(2);-1.414\*y(2)-y(1)+1];*





蘇州大學

SOOCHOW UNIVERSITY

## 练习：

- 运行以下命令：
- `>> a=eye(5);`
- `>> for n=a(2:end,:)`
- `.....`
- 则for循环的循环次数是\_\_A\_\_。
- A. 5                      B. 4
- C. 3                      D. 1





- 运行以下命令则 C
- `>> a=[1 2 3]`
- `>> keyboard`
- `K>> a=[1 2 4];`
- `K>> return`
- A. `a=[1 2 3]`                      B. `a=[1 2 4]`
- C. 命令窗口的提示符为“K>>”
- D. 出错





蘇州大學

SOOCHOW UNIVERSITY

## 练习：

- 编写M脚本文件，分别使用for和while循环语句计算  $sum = \sum_{i=1}^{10} i^i$ ，当sum>1000时终止程序。

答案：

```
clear
sum=0
for n=1:1000
    sum=n^n+sum
    if sum>1000
        break
    end
end
```

```
clear
sum=0;n=1;
while sum<1000
    sum=n^n+sum
    n=n+1
end
```





蘇州大學

SOOCHOW UNIVERSITY

## 练习：

- 求  $y(x) = -e^{-x} |\sin(\sin(x))|$  在  $x=0$  附近的最小值。

答案：

```
fhnd=@(x)(-exp(-x)*abs(sin(sin(x))))  
y=fminbnd(fhnd,0,10)
```





蘇州大學

SOOCHOW UNIVERSITY

## 练习：

- 解微分方程， $y_0=1$ ，并绘制曲线

$$\frac{dy}{dt} + y \tan y = \cos y$$

```
>> f=inline('sin(x)*exp(-z*x)','x','z')    %创建  
inline对象f  
>> class(f)
```

答案：

```
fhnd=@(t,y)(y*tan(y)-cos(y));  
[t,y]=ode45(fhnd,[0,20],1);  
plot(t,y)
```

