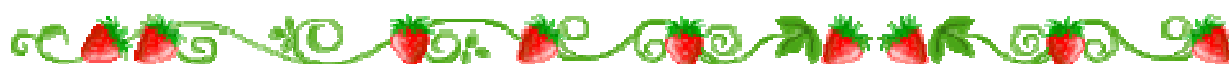


---

# 上机实验指导手册

重庆邮电大学通信与信息工程学院-电子信息工程系



# 基础知识：MATLAB 程序入门和基础应用

## 1.1 MATLAB 环境

### 一. MATLAB 的发展及其主要功能

MATLAB 是一套功能强大的工程技术人员必备的高端数学类科技应用软件。MATLAB 语言是由美国的 Clever Moler 博士于 1980 年开发的，设计者的初衷是为了解决“线性代数”课程的矩阵运算问题，MATLAB = matrix + laboratory，即矩阵实验室。1984，Clever Moler 博士与其合作者成立了 mathwork 公司，推出了 MATLAB 第 1 版(DOS 版)。此后，MATLAB 版本不断更新，从 1992 年的 4.0 版发展到 2004 年的 7.0 版，MATLAB 已发展成为由 MATLAB 语言、MATLAB 工作环境、MATLAB 图形处理系统、MATLAB 数学函数库和 MATLAB 应用程序接口五大部分组成的集数值计算、图形处理、程序开发为一体的功能强大的系统。

MATLAB 具有以下基本功能：

- 数值计算功能，矩阵运算功能
- 符号计算功能
- 图形处理集可视化功能
- 可视化建模集动态仿真功能

科学计算有数值计算和符号计算之分，MATLAB 的数值计算功能非常强大，它提高了丰富的数值计算函数，而且所采用的数值计算算法都是国际公认的、最先进的、可靠的算法。

### 1.数值计算功能，矩阵运算功能

MATLAB 提供了丰富的矩阵运算处理功能，是基于矩阵运算的处理工具。它的变量是矩阵，数值运算即为矩阵的运算。例如  $C = A + B$ ，A,B,C 都是矩阵，是矩阵的加运算。即使一个常数， $Y=5$ ，MATLAB 也看做是一个  $1 \times 1$  的矩阵。

### 2.符号运算功能

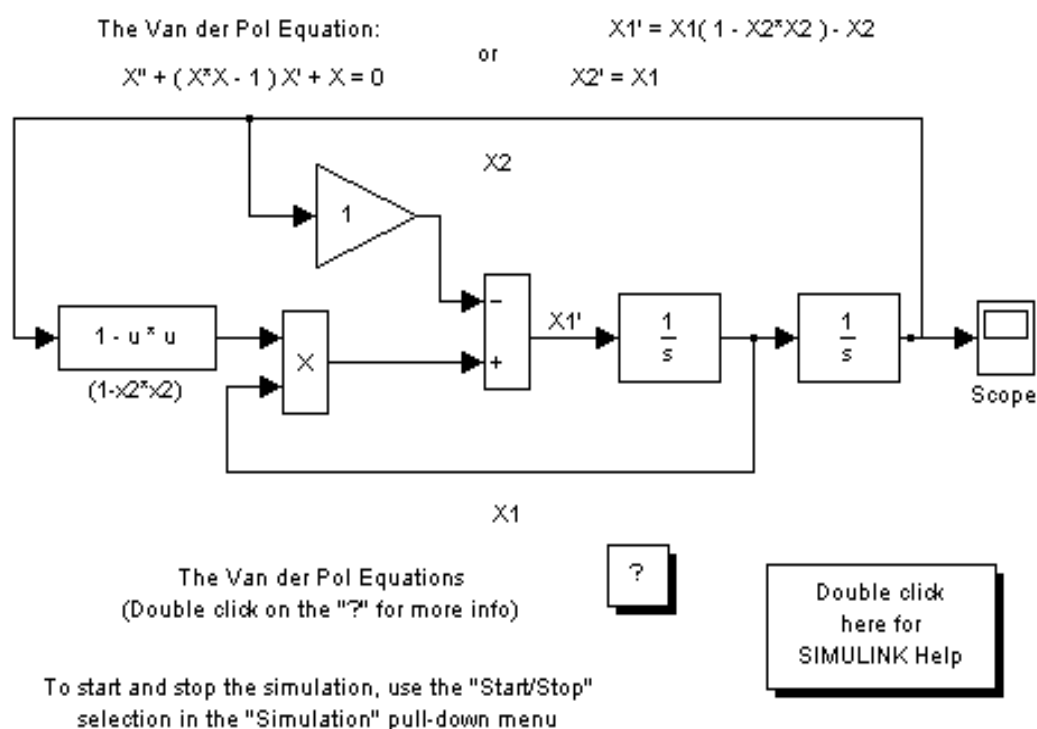
符号运算即用字符串进行数学分析，该运算允许变量不赋值而参与运算。符号运算用于解代数方程、微积分、复合导数、积分、二重积分、有理函数、微分方程、泰勒级数展开、寻优等等，可求得解析符号解。

### 3.丰富的绘图功能与计算结果的可视化

MATLAB 提供了两个层次的绘图操作：一种是对图形句柄进行的低层绘图操作，另一种是建立在低层绘图操作上的高层绘图操作，可以绘制两维、三维图形。

### 4.图形化程序编制功能

MATLAB 提供了利用动态系统进行建模、仿真和分析的软件包，利用此软件包实现结构图编程，而不用程序编程，只需拖几个方块、连几条线，即可实现编程功能。



### 5.丰富的 MATLAB 工具箱

MATLAB 包含两部分内容：基本部分和各种可选的工具箱。基本部分构成了 MATLAB 的核心内容，也是使用和构造工具箱的基础。MATLAB 工具箱分为两大类：功能性工具箱和学科性工具箱。功能性工具箱主要用来扩充其符号计算功能、可视建模仿真功能及文字处理功能等。学科性工具箱专业性比较强，如控制系统工具箱、信号处理工具箱、图象处理工具箱、神经网络工具箱、通讯工具箱等等。

MATLAB 功能强大、简单易学、编程效率高，所以广泛应用于工业研究与开发，数学教学（特别是线性代数），数值分析和科学计算方面的教学与研究，电

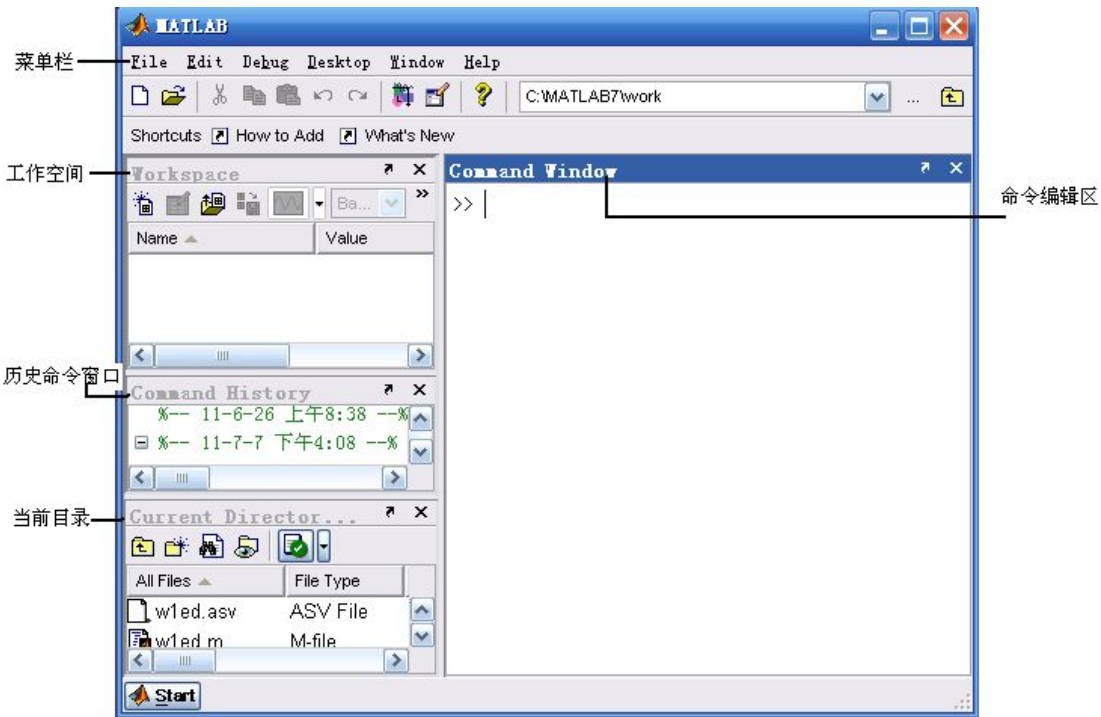
子学、控制理论和物理学等工程和科学学科方面的教学与研究，经济学、化学和生物学等计算问题的教学与研究等多个领域。

## 二. 启动 MATLAB

启动 MATLAB 系统有 3 中常见方法：

1. MATLAB 系统启动程序以快捷方式放在 Windows 桌面上，用鼠标双击 MATLAB 图标即可启动。
- 2.单击任务栏上的“开始”按钮，选择“程序”菜单项，然后单击 MATLAB 菜单项中的 MATLAB 7.0 程序，即可启动。
- 3.运行 c:\matlab\bin\matlab.exe 启动程序 matlab.exe。

## 三. MATLAB 工作环境



在菜单栏下的大窗口就是 MATLAB 的主窗口，在大窗口里设置了 4 个小窗口 (这是桌面平台的默认设置)：“Command Windows”、“Workspace”、“Current Directory”、“Command History”。“Command Windows”命令编辑区用于输入命令和显示计算结果。在命令提示符》后输入命令并回车，MATLAB 就会解释执行所输入的命令，并在命令后面给出计算结果。“Workspace”为工作空间，用于显示计算机内存中当前时刻的变量情况。“Command History”为历史命令窗口，运行过的语句就会保存在历史命令窗口。“Current Directory”为当前目录窗口。

MATLAB 命令窗口的菜单栏包含 File、Edit、Debug、Desktop、Window 和 Help

等 6 个菜单项。File 功能包括新建、打开文件，导入数据，关闭命令窗，把工作区的所有变量保存为 MAT 文件，设置路径等等。Edit 的主要功能包括复制、粘贴、删除，刷新命令窗口、历史窗口、工作空间中的操作记录。Debug 为程序调试。Desktop 设置用户界面的窗口显示。Window 用来选择当前要进行操作的窗口。Help 为 MATLAB 的帮助系统。

例、用 MATLAB 求解线性方程。

$$3x_1 + x_2 - x_3 = 3.6$$

$$x_1 + 2x_2 + 4x_3 = 2.1$$

$$-x_1 + 4x_2 + 5x_3 = -1.4$$

程序：A=[3 1 -1;1 2 4;-1 4 5]; b=[3.6;2.1;-1.4]; x=A\b

计算结果为：x =

1.4818

-0.4606

0.3848

#### 四. MATLAB 的工作方式

交互式的命令行工作方式和 M 文件的编程工作方式。前者适用于运算过程简单，几条指令即可完成运算的情况，在这种情况下，只需在命令窗口中逐条输入命令按回车即出现运算结果，比如在命令窗口中输入如下指令，然后按回车键：

$$[(24+57)*39-88]/13$$


马上显示出答案：ans=236.2308

这是对于问题较简单的情况，但是当解决的问题较为复杂时，一次需要执行很多条指令，显然用交互式的指令工作方式直接逐条输入指令是不现实的，这就要用到后一种工作方式：M 文件的编程工作方式。

M 文件的编程工作方式是在命令窗口中调用 M 文件，从而执行文件中的多条指令，M 文件是由 MATLAB 命令行构成的文本文件，以.m 为后缀名。它又分为两种形式：命令 M 文件和函数 M 文件，分别简称为命令文件和函数文件。


##### 1. M 文件的建立

为建立新的 M 文件，启动 MATLAB 文本编辑器有 3 种方法：


- (1) 菜单操作。从 MATLAB 命令窗口的 File 菜单中选择 New 菜单项，再选择 M-file 命令。屏幕将出现 MATLAB Editor/Debugger 窗口。该窗口是一个集编辑与调试两种功能于一体的工具环境。利用它不仅可以完成基本的文本编辑操作，还可以对 M 文件进行调试。MATLAB 文本编辑器的操作界面与使用方法和其他 Windows 编辑器相似，这里不做详细介绍。
- (2) 命令操作。在 MATLAB 命令窗口输入命令 `edit`，启动 MATLAB 文本编辑器后，输入 M 文件的内容并存盘。
- (3) 命令操作按钮。单击 MATLAB 命令窗口工具栏上的新建命令按钮 ，启动 MATLAB 文本编辑器，输入 M 文件的内容并存盘。

## 2. 编辑已有的 M 文件

编辑已有的 M 文件，也有 3 种方法：

- (1) 菜单操作。从 MATLAB 命令窗口的 File 菜单中选择 Open M-file 命令，则屏幕出现 Open 对话框，在 Open 对话框中选中所需打开的 M 文件。在文档窗口可以对打开的 M 文件进行编辑修改。编辑完成后，将 M 文件存盘。
- (2) 命令操作。在 MATLAB 命令窗口输入命令：`edit 文件名`，则打开指定的 M 文件。
- (3) 命令按钮操作。单击 MATLAB 命令窗口工具栏上的打开命令按钮 ，再从弹出的对话框中选择所需打开的 M 文件。

## 3. M 文件的执行

M 文件的执行有 3 种方式：当用户在命令窗口中输入 M 文件的文件名按回车后，系统将搜索该文件并逐条执行该文件中的命令。第二种方法，保存 m 文件，点击 Debug 菜单下的 run 也可执行。第三种方法，点击菜单栏中的图标 。

## 4. M 文件的命名

保存 M 脚本文件时，文件名可以是字母、数字、下划线的组合，**但是必须以字母开头**，不允许使用中文命名，长度不超过 32 个字符，区分大小写，扩展名是 .m。

## 五. 常见操作系统命令

除了菜单操作之外, MATLAB 还提供了许多直接在命令窗口执行的操作命令, 以实施有关的操作。

命令	功能	命令	功能
dir	显示文件目录清单	more	使其后的显示内容分屏进行
cd	改变当前工作目录	copyfile	复制文件
mkdir	建立目录	what	显示当前目录下的 M 文件和 MAT 文件
delete	删除文件	clc	清除命令窗口显示的内容
type	显示文本文件内容	web	打开网络浏览器
clear	清除变量空间的内容	pause	运行中暂停按任意键继续

## 1.2 MATLAB 数据及运算

MATLAB 是由早期专门用于矩阵运算的软件发展而来。矩阵是 MATLAB 最基本、最重要的数据对象，MATLAB 的大部分运算或命令都是在矩阵运算的意义下执行的，而且这种运算定义在复数域上。

### 一. 变量与赋值

#### 1. 变量的命名

变量代表一个或若干个内存单元，为了对变量所对应的存储单元进行访问，需要给变量命名。在 MATLAB 中，变量以字母开头，可以由字母、数字和下划线混合组成，区分字母大、小写，字符长度不超过 31 个。在 MATLAB 工作空间中，还驻留几个系统本身定义的变量，它们有特定的含义，在使用时，应尽量避免对这些变量重新赋值。如  $\pi$ ， $i$ 、 $j$  表示虚数单位， $\infty$  表示无穷大， $\text{ans}$  表示计算结果的缺省赋值变量。

#### 2. 变量的赋值

MATLAB 赋值语句有两种格式：

(1) 变量=表达式

(2) 表达式

其中表达式是用运算符将有关运算量连接起来的式子，其结果是一个矩阵。在第一种语句形式下，MATLAB 将右边表达式的值赋给左边的变量，而在第二种语句形式下，将表达式的值赋给 MATLAB 的预定义变量  $\text{ans}$ 。

例 1. 计算表达式  $\frac{5 + \cos 47^\circ}{1 + \sqrt{7} - 2i}$  的值，并显示计算结果。

在 MATLAB 命令窗口输入命令：

```
x=(5+cos(47*pi/180))/(1+sqrt(7)-2*i);
```

输出结果是：

```
x =
```

```
1.1980+ 0.6572i
```

#### 3. 数据的表示

采用十进制技术，可带小数点和负号并可采用科学计数法，最多大约保持有效数字 16 位。一般情况下，MATLAB 内部每一个数据元素都是用双精度来表示和



存储的。数据输出时用户可以用 `format` 命令设置或改变数据输出格式。如果输出矩阵的每个元素为纯整数，**MATLAB** 就用不加小数点的纯整数格式显示结果，只要矩阵中有一个元素不是纯整数，**MATLAB** 将按当前的输出格式显示计算结果，缺省的输出格式是 `short` 格式。`short` 格式输出小数点后 4 为，最多不超过 7 位有效数字。

## 二. 矩阵

矩阵是 `matlab` 进行数据处理和运算的基本元素，大部分运算和命令都是在矩阵的意义进行的，仅有一行或一列的矩阵称为向量。

### 1. 矩阵的建立

#### (1) 直接输入法

最简单的建立矩阵的方法是从键盘直接输入矩阵的元素。具体方法如下：将矩阵的元素用方括号括起来，按矩阵行的顺序输入各元素，**同一行的各元素之间用空格或逗号分隔，不同行的元素之间用分号分隔。**

#### (2) 利用 M 文件建立矩阵

对于比较大且比较复杂的矩阵，可以为它专门建立一个 `M` 文件。下面通过一个简单例子来说明如何利用 `M` 文件创建矩阵。

例 2. 新建 `m` 文件命名为 `ma.m`，在该文件中输入

```
X=[1 2;3 4];
```

在命令窗口输入 `ma`，即运行该文件，就会自动建立一个名为 `ma` 的矩阵，可供以后使用。

#### (3) 利用 **MATLAB** 函数建立矩阵

**MATLAB** 提供了许多产生特殊矩阵的函数，可以利用它们建立矩阵。常用的产生通用特殊矩阵的函数有：

`zeros`: 产生全 0 矩阵(零矩阵)。

`ones`: 产生全 1 矩阵(幺矩阵)。

`eye`: 产生单位矩阵。

`rand`: 产生 0~1 间均匀分布的随机矩阵。

`randn`: 产生均值为 0，方差为 1 的标准正态分布随机矩阵。

#### (4) 建立大矩阵

大矩阵可由方括号中的小矩阵或向量建立起来。

## 2. 冒号表达式

MATLAB 中，可利用冒号表达式建立行向量。一般格式是：

`e1:e2:e3`

其中 `e1` 为初始值，`e2` 为步长，`e3` 为终止值。

在 MATLAB 中，还可以用 `linspace` 函数产生行向量。其调用格式为：

`linspace(a,b,n)`

其中 `a` 和 `b` 是生成向量的第一个和最后一个元素，`n` 是元素总数。显然，`linspace(a,b,n)` 与 `a:(b-a)/(n-1):b` 等价。

## 3. 矩阵元素

MATLAB 允许用户对一个矩阵的单个元素进行赋值和操作。如 `A(3,2)` 表示 `A` 矩阵中第 3 行第 2 列的元素。也可采用矩阵元素的序号来引用矩阵元素，矩阵元素按列编号，先第一列，再第二列，依次类推。如 `A=[1 2 3;4 5 6]`, `A(2,1)=4`, `A(2)=4`。

# 三. MATLAB 运算

## 1. 算术运算

### (1) 基本算术运算

MATLAB 的基本算术运算有：`+`(加)、`-`(减)、`*`(乘)、`/`(右除)、`\`(左除)、`^`(乘方)。注意，运算是在矩阵意义下进行的，单个数据的算术运算只是一种特例。

**矩阵加减运算：**假定有两个矩阵 `A` 和 `B`，则可以由 `A+B` 和 `A-B` 实现矩阵的加减运算。运算规则是：**若 `A` 和 `B` 矩阵的维数相同**，则可以执行矩阵的加减运算，`A` 和 `B` 矩阵的相应元素相加减。如果 `A` 与 `B` 的维数不相同，则 MATLAB 将给出错误信息，提示用户两个矩阵的维数不匹配。

例：`A=B=[1 1 1;2 2 2;3 3 3];`

`A+B`

`ans=2 2 2`

`4 4 4`

`6 6 6`

`A-B`

`ans=0 0 0`

`0 0 0`

```

0 0 0
A+2
ans=3 3 3
4 4 4
5 5 5

```

**矩阵乘法：**假定有两个矩阵 A 和 B，若 A 为  $m \times n$  矩阵，B 为  $n \times p$  矩阵，则  $C=A*B$  为  $m \times p$  矩阵。

例：A=B=[1 1 1;2 2 2;3 3 3];

```

A*B
ans= 6      6      6
     12     12     12
     18     18     18

```

**矩阵除法：**在 MATLAB 中，有两种矩阵除法运算：\和/，分别表示左除和右除。如果 A 矩阵是非奇异方阵，则 A\B 和 B/A 运算可以实现。A\B 等效于 A 的逆左乘 B 矩阵，也就是  $\text{inv}(A)*B$ ，而 B/A 等效于 A 矩阵的逆右乘 B 矩阵，也就是  $B*\text{inv}(A)$ 。对于含有标量的运算，两种除法运算的结果相同，如 3/4 和 4\3 有相同的值，都等于 0.75。又如，设 a=[10.5,25]，则 a/5=5\a=[2.1000 5.0000]。对于矩阵来说，左除和右除表示两种不同的除数矩阵和被除数矩阵的关系。对于矩阵运算，一般  $A \setminus B \neq B/A$ 。

例：A=[1 2 3;4 2 6;7 4 9]; B=[4 3 2;7 5 1;12 7 92]

```

A/B
ans= 7.9623   -4.2453   -0.0943
     -4.5472    2.9434    0.1321
     -5.1321    3.6415    0.1698

```

```

A\B
ans= 0.5000   -0.5000   44.5000
     1.0000    0.0000   46.0000
     0.5000    1.1667  -44.8333

```

**矩阵乘方：** $A^B$ ，矩阵 A、B 均为标量时，表示 A 的 B 次方幂；A 为方阵，B

为正整数时，表示矩阵 A 的 B 次乘积；A、B 均为矩阵则无定义。

例：A=[1 2 3;4 5 6;7 8 0]

A^2

```
ans= 30    36    15
      66    81    42
      39    54    69
```

## (2) 点运算

在 MATLAB 中，有一种特殊的运算，因为其运算符是在有关算术运算符前面加点，所以称为点运算。点运算符有.\*、./、.\和.^。如果 A、B 矩阵的维数相同，A.\*B 表示 A 和 B 单个元素之间的对应相乘。A./B 表示 A 矩阵除以 B 矩阵的对应元素，B.\A 等价于 A./B。

例：A=B=[1 1 1;2 2 2;3 3 3];

A.\*B

```
ans = 1      1      1
      4      4      4
      9      9      9
```

例：A = [1 2 3;4 2 6;7 4 9]; B= [4 3 2;7 5 1;12 7 92]

A./B

```
ans = 0.2500    0.6667    1.5000
      0.5714    0.4000    6.0000
      0.5833    0.5714    0.0978
```

A.\B

```
ans = 4.0000    1.5000    0.6667
      1.7500    2.5000    0.1667
      1.7143    1.7500   10.2222
```

例：A=[1 2 3;4 5 6;7 8 0]

A.^2

```
ans= 1      4      9
      16     25     36
      49     64      0
```

### (3) MATLAB 常用数学函数

MATLAB 提供了许多数学函数，函数的自变量规定为矩阵变量，**运算法则是将函数逐项作用于矩阵的元素上，因而运算结果是一个与自变量同维数的矩阵。**

函数名	含义	函数名	含义
sin	正弦函数	exp	自然指数函数
cos	余弦函数	sqrt	平方根函数
tan	正切函数	log	自然对数函数
asin	反正弦函数	log10	常用对数函数
abs	取模函数	real	复数的实部
angle	复数的幅角	imag	复数的虚部
conj	求复数的共轭	max	求矩阵元素的最大值
sinc	抽样函数	sign	符号函数

## 2. 关系运算

MATLAB 提供了 6 种关系运算符： $>$ （大于）、 $<$ （小于）、 $==$ （等于）、 $>=$ （大于或等于）、 $<=$ （小于或等于）、 $\sim$ （不等于）。当两个比较量为标量时，直接比较两数的大小。若关系成立，表达式结果为 1，否则为 0。

## 3. 逻辑运算

MATLAB 提供了 3 种逻辑运算符： $\&$ （与）、 $|$ （或）、 $\sim$ （非）。

### 分号的作用：

结尾加分号：将计算结果存入内存，但不显示在屏幕上；结尾不加分号：将计算结果存入内存，同时显示在屏幕上。

## 四. MATLAB 符号计算

在数学运算中除了数值计算以外，在数学、物理、应用工程和科学方面的抽象运算，即计算式中带有  $x$ 、 $y$  等符号变量、表达式的运算，也占有相当大的比例。符号计算可以获得比数值计算更一般的结果。需要注意的是，在符号运算的整个过程中，所有的运算均是以符号进行的，即使以数字形式出现的量也是字符量。举一个简单的例子，在命令窗口中输入如下符号表达式按回车：

```
f=sym('sin(x/2)');
```

```
dfdx=diff(f)
```

显示如下结果：
$$\text{dfdx} = 1/2*\cos(1/2*x)$$

上式是对  $\sin(x/2)$  求导的过程，一切都是由符号变量和符号表达式完成，没有涉及到具体的数值运算，其中  $1/2$  也被当作是字符常量。

## 1. 建立符号变量

使用符号变量前先要进行定义，定义语句是：`sym` 或 `syms` 变量名列表。前者定义一个单一的符号变量，后者可以一次定义多个符号变量。如下例：

`a=sym('a')`      定义 a 为符号变量

`syms a b c`      定义 a,b,c 均为符号变量

符号表达式是由符号变量组成的一个表达式，符号方程是将一个符号表达式通过等号给一个符号变量。凡是用到 `sym` 命令的时候，表达式和方程式对空格都是敏感的，因此不要随意添加空格符到式中。

## 2. 符号表达式的创建

用 `sym` 命令直接创建符号表达式，这种创建方式不需要在前面有任何说明，使用快捷方便。例如：

`f=sym('a+b+c')`

在命令窗口中输入上述语句按回车键，出现以下结果：`f= a+b+c`。说明已成功将符号表达式 `a+b+c` 赋给变量 `f`。

按照普通书写方式创建符号表达式。它需要在具体创建一个符号表达式之前，将这个表达式所包含的全部符号变量创建完毕。例如：

`syms a b c`

`f=a+b+c`

在命令窗口中输入上述语句后，也出现和上面相同的结果，创建符号表达式成功。

## 1.3 MATLAB 绘图

MATLAB 语言丰富的图形表现方法，使得数学计算结果可以方便地、多样性地实现了可视化，这是其它语言所不能比拟的。用户不需要过多考虑绘图细节，只需给出一些基本参数就能得到所需图形，这一类函数称为高层绘图函数。除此之外，MATLAB 还提供了直接对图形句柄进行操作的低层绘图操作。这类操作将图形的每个图形元素（如坐标轴、曲线、文字）看作一个独立的对象，系统给每个图形对象分配一个句柄，以后可以通过该句柄对该图形元素进行操作，而不影响图形的其它部分。高层绘图操作简单明了、方便高效，是用户最常使用的绘图方法，而低层绘图操作控制和表现图形的能力更强，为用户更加自主地绘制图形提供了条件。

### 一. 二维绘图

在 MATLAB 中，最基本且应用最广泛的绘图函数为 `plot` 函数，利用它可以在二维平面上绘制出不同的曲线。

#### 1. `plot` 函数的基本用法

`plot` 命令自动打开一个图形窗口 **Figure**，绘制  $x$ - $y$  平面上的线性坐标曲线图，因此需提供一组  $x$  坐标以及与各点  $x$  坐标对应的  $y$  坐标，这样就可以绘制分别以  $x$ 、 $y$  坐标为横纵坐标的二维曲线。它用直线连接相邻两数据点来绘制图形，根据图形坐标大小自动缩扩坐标轴，将数据标尺及单位标注自动加到两个坐标轴上，可以把  $x$ 、 $y$  轴用对数坐标表示。如果已经存在一个图形窗口，`plot` 命令则清除当前图形，绘制新图形；可单窗口单曲线绘图；可单窗口多曲线绘图，可单窗口多曲线分图绘图；可多窗口绘图；可任意设定曲线颜色和线型；可给图形加坐标网线和图形加注功能。`plot` 函数的基本调用格式为：

- `plot(x)` —— 缺省自变量绘图格式， $x$  为向量，以  $x$  元素值为纵坐标，以相应元素下标为横坐标绘图。
- `plot(x,y)` —— 基本格式，以  $y(x)$  的函数关系作出直角坐标图，如果  $y$  为  $n \times m$  的矩阵，则以  $x$  为自变量，作出  $m$  条曲线。
- `plot(x1,y1,x2,y2)` —— 多条曲线绘图格式。

■ `plot(x,y,'s')` —— 开关格式，开关量字符串 `s` 设定曲线颜色和绘图方式，使用颜色字符串的前 1~3 个字母，如 `yellow—yel` 表示等；或  
`plot(x1,y1,'s1',x2,y2,'s2', ...)`，`s` 的标准设定值如下：

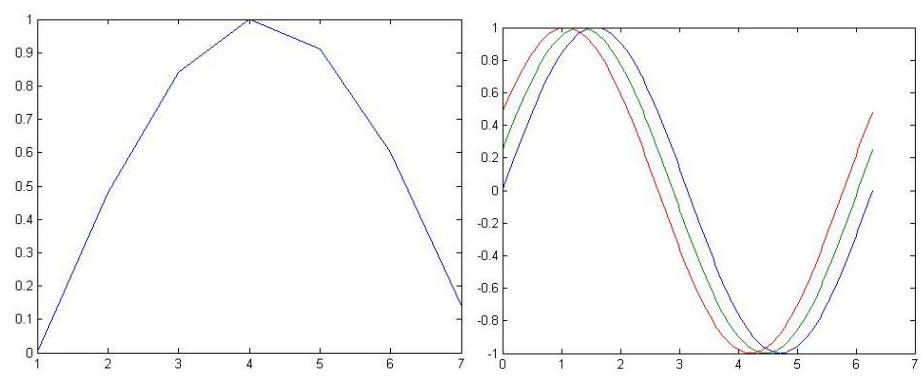
颜色选项		标记符号选项	
字母	颜色	标记符号	说明
y	黄色	.	点线
m	粉红	o	圈线
r	大红	x	x线
g	绿色	+	加号线
b	蓝色	*	星号线
w	白色	—	实线
k	黑色	:	虚线

## 2. 二维曲线的绘制

### (1) 单窗口单曲线绘图

例 1: `x=[0,0.48,0.84,1,0.91,0.6,0.14]`

`plot(x)`



单窗口单曲线

单窗口多曲线

### (2) 单窗口多曲线绘图

例 2: `t=0:pi/100:2*pi;`

`y=sin(t);y1=sin(t+0.25);y2=sin(t+0.5);`

`plot(t,y,t,y1,t,y2)`

### (3) 单窗口多曲线分图绘图

在实际应用中，经常需要在一个图形窗口绘制若干个独立的图形，这就需要对



图形窗口进行分割。分割后的图形窗口由若干个绘图区组成，每一个绘图区可以建立独立的坐标系并绘制图形。同一个图形窗口中的不同图形称为子图。

`subplot` 函数，用来将当前图形窗口分割成  $m \times n$  个绘图区，即每行  $n$  个，共  $m$  行，区号按行优先排号，且选定  $p$  区位当前活动区。

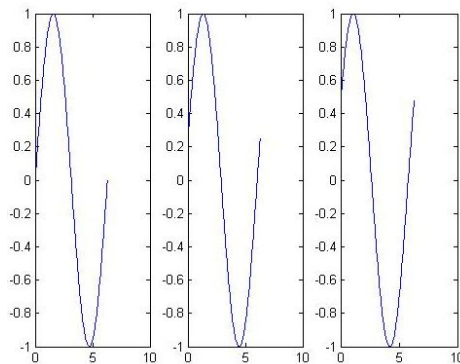
**调用格式：**`subplot(m,n,p)`

例 3、把例 2 中的三个曲线分图绘制

```
subplot(1,3,1); plot(t,y)
```

```
subplot(1,3,2); plot(t,y1)
```

```
subplot(1,3,3); plot(t,y2)
```



单窗口多曲线分图绘图

#### (4) 多窗口绘图

`figure(n)` —— 创建窗口函数， $n$  为窗口顺序号。

例 3.  $t=0:\pi/100:2\pi$ ;

```
y=sin(t);y1=sin(t+0.25);y2=sin(t+0.5);
```

```
plot(t,y); %自动出现第一个窗口
```

```
figure(2); plot(t,y1); %在第二窗口绘图
```

```
figure(3); plot(t,y2); %在第三窗口绘图
```

程序执行结果就是将上述的三个正弦波形分窗口绘制。

### 3. 图形加注功能

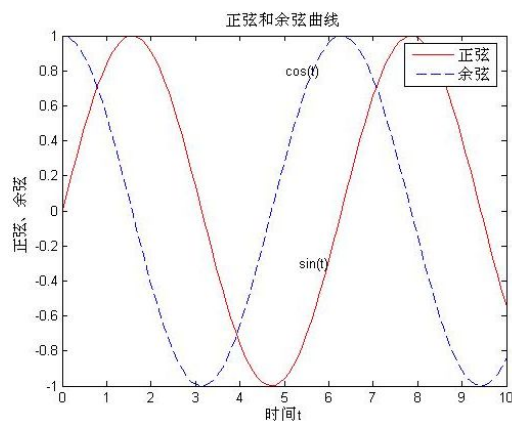
绘制完图形后，可能还需要对图形进行一些辅助操作，以使图形意义更加明确，可读性更强。MATLAB 允许将标题、坐标轴标记、网格线及文字注释加注到图形上，这些函数为：

`title` —— 给图形加标题

xlabel —— 给 x 轴加标注  
ylabel —— 给 y 轴加标注  
text —— 在图形指定位置加标注  
gtext —— 将标注加到图形任意位置  
grid on(off) —— 打开、关闭坐标网格线  
legend —— 添加图例  
axis —— 控制坐标轴的刻度

例 4.  $t=0:0.1:10$

```
y1=sin(t);y2=cos(t);plot(t,y1,'r',t,y2,'b--');  
x=[1.7*pi;1.6*pi];  
y=[-0.3;0.8];  
s=['sin(t)';'cos(t)'];  
text(x,y,s);  
title('正弦和余弦曲线');  
legend('正弦','余弦')  
xlabel('时间 t'),ylabel('正弦、余弦')
```



#### 4. 绘制二维图形的其它函数

(1) 在线性直角坐标系中，其它形式的图形有条形图、阶梯图、杆图和填充图，所采用的函数分别是：

bar(x,y,选项)    stairs(x,y,选项)    stem(x,y,选项)    fill(x1,y1,选项 x2,y2,选项.....)

前 3 个函数的用法与 plot 函数相似，只是没有多输入变量形式。fill 函数按元素下标渐增次序用直线段连接 x，y 对应元素定义的数据点。

例 5.分别以条形图、填充图、阶梯图和杆图绘制曲线  $y = 2e^{-0.5x}$

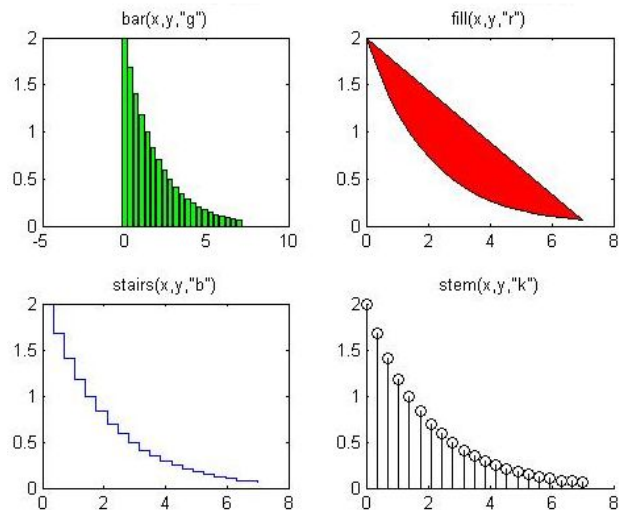
程序:  $x=0:0.35:7;y=2*\exp(-0.5*x);$

```
subplot(2,2,1);bar(x,y,'g');title('bar(x,y,"g")');
```

```
subplot(2,2,2);fill(x,y,'r');title('fill(x,y,"r")')
```

```
subplot(2,2,3);stairs(x,y,'b');title('stairs(x,y,"b")');
```

```
subplot(2,2,4);stem(x,y,'k');title('stem(x,y,"k")');
```



## (2) ezplot ——符号函数的简易绘图函数

ezplot 的调用格式:

ezplot(f) —这里 f 为包含单个符号变量 x 的符号表达式, 在 x 轴的默认范围  $[-2\pi, 2\pi]$  内绘制 f(x) 的函数图

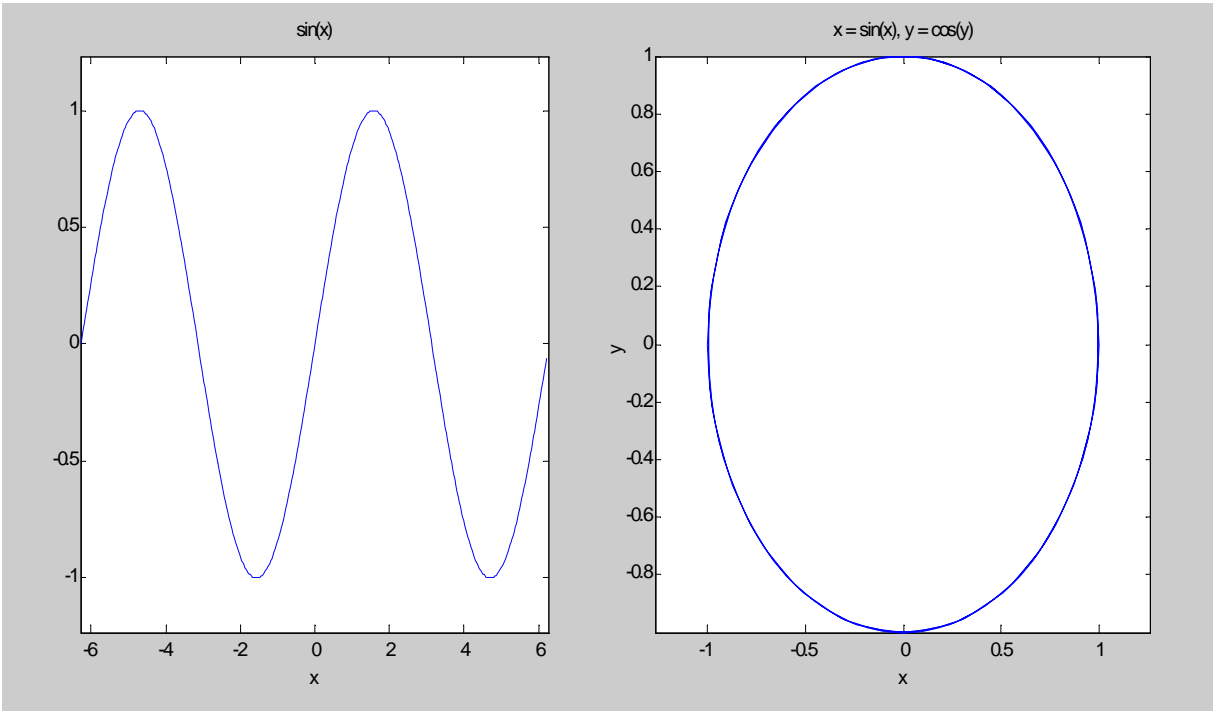
ezplot(f,[xmin,xmax]) — x 的给定区间绘制

ezplot(x,y)—绘制  $x=x(t), y=y(t)$  的二维曲线,  $t \in [-2\pi, 2\pi]$

例 6.绘制  $\sin(x), f(x,y)=(\sin x, \cos y)$

```
ezplot('sin(x)')
```

```
ezplot('sin(x)','cos(y)',[-4*pi 4*pi],figure(2))
```



## 1.4 信号与系统常用 MATLAB 函数简介

为了方便实验时查阅，下面列出信号与系统实验中用到的几个函数。

### 1. subs 函数

功能：实现符号变量的替换

subs 函数实现在一个符号解或在一个符号表达式中，将一些符号变量替换成数字或其他符号。

subs(s,new)用新变量 new 替换 s 中的默认变量。

subs(s,new,old)用新变量 new 替换 s 中的指定变量 old。

### 2. flipr 函数

功能：实现矩阵的反折

a=flipr(b)将产生维数与矩阵 b 相同的矩阵 a，其元素是由矩阵 b 的元素按列的方向进行反折而得。

### 3. length 函数

功能：计算矩阵的长度（列数）

a=length(b)将矩阵 b 的列数赋给变量 a。

### 4. conv 函数

功能：求卷积

f=conv(f1,f2)f1,f2 为参与卷积运算的两个序列，f 为卷积的结果，f 的长度为 length(f1)+length(f2)-1。

### 5. impulse 函数

功能：求连续系统的冲激响应并绘出图形

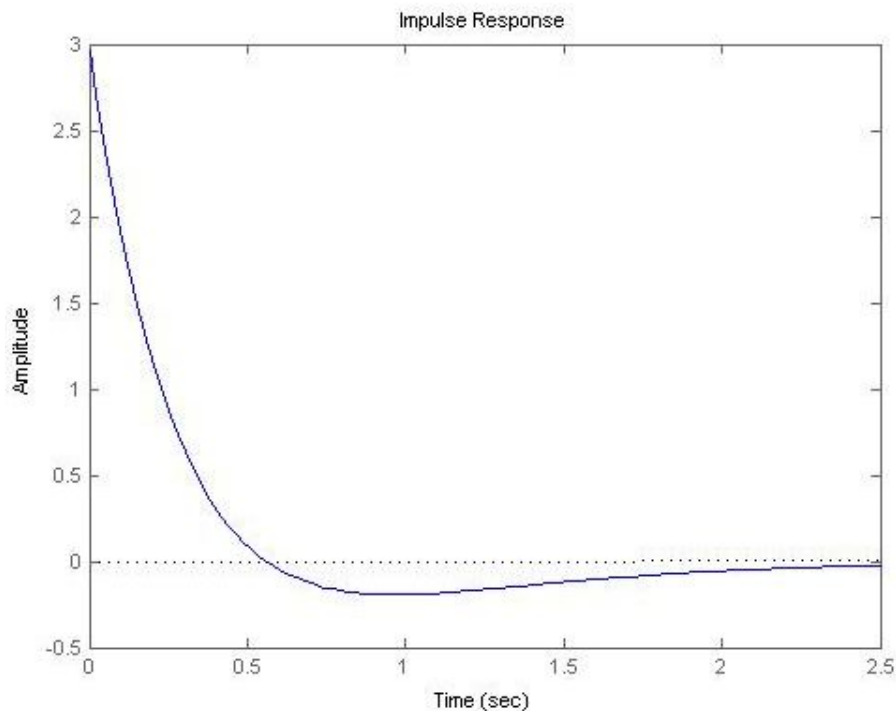
格式：impulse(b,a)，其中 a 为响应的系数行向量，b 为激励的系数行向量

例：描述连续系统的微分方程为： $y''(t) + 5y'(t) + 6y(t) = 3f''(t) + 2f(t)$

运行如下程序：

```
a=[1 5 6];b=[3 2]; impulse(b,a);
```

则绘出该连续系统的冲激响应波形，如下图：



`impulse(b,a,t)`绘出向量 **a** 和 **b** 定义的连续系统在 0~t 时间范围内冲激响应的时域波形。

## 6. step 函数

功能：求连续系统的阶跃响应并绘出图形

`step(b,a)` 绘出向量 **a** 和 **b** 定义的连续系统的阶跃响应的时域波形。

`step(b,a,t)`绘出向量 **a** 和 **b** 定义的连续系统在 0~t 时间范围内阶跃响应的时域波形。

## 7. lsim 函数

功能：计算连续系统的冲激响应及零状态响应并绘出相应图形

格式：`lsim(b,a,x,t)`，其中 **a** 为响应的系数行向量，**b** 为激励的系数行向量，**x** 为激励，**t** 为时间范围。

例：描述连续系统的微分方程为： $y''(t) + 2y'(t) + y(t) = f'(t) + 2f(t)$ ，激励

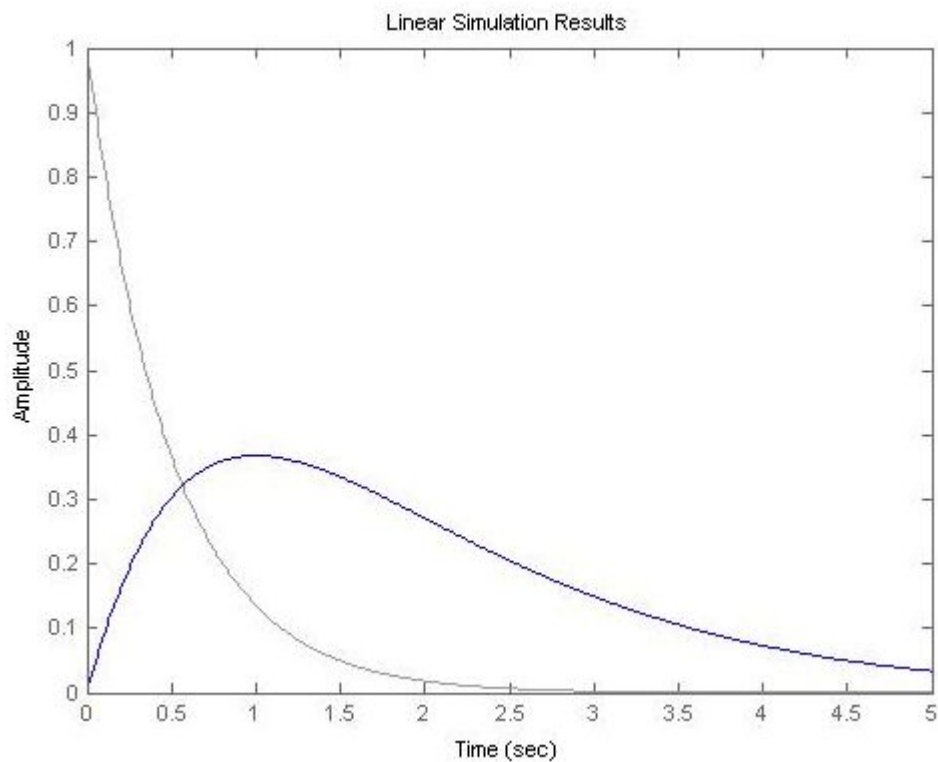
$f(t) = e^{-2t}\varepsilon(t)$ ，求零状态响应。

程序如下：`a=[1 2 1];b=[1 2];`

`t=0:0.01:5;x=exp(-2*t);`

`lsim(b,a,x,t)`

执行结果如下：



图中蓝色曲线为零状态响应，灰色曲线为冲激响应。

## 8. impz 函数

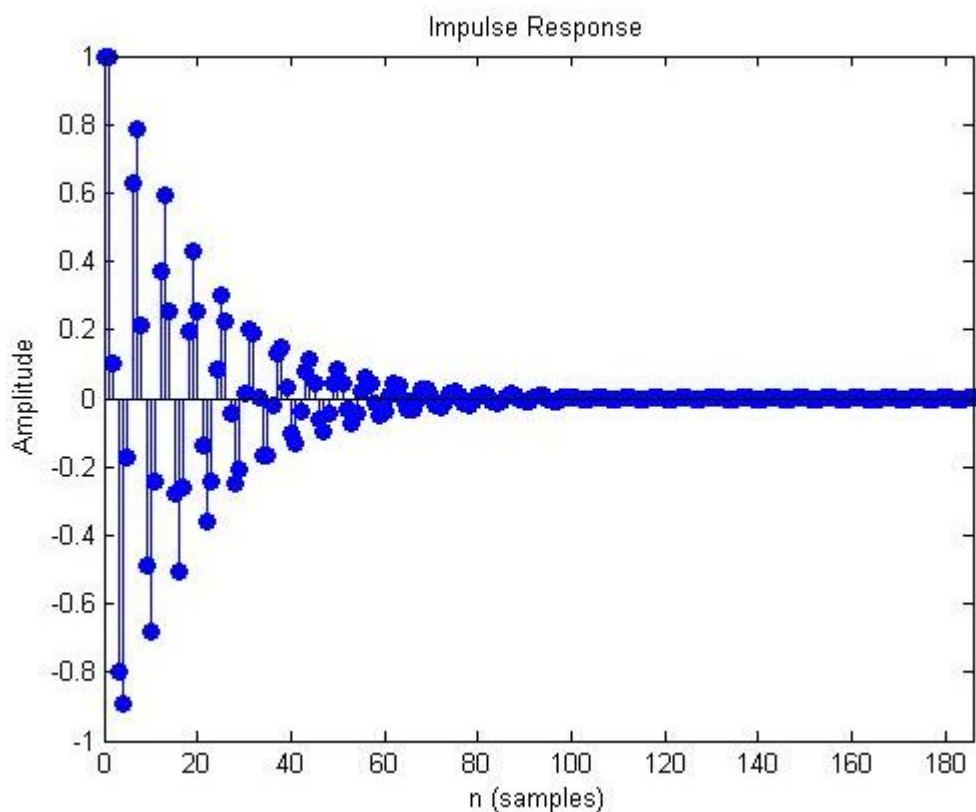
功能：求离散系统的单位响应并绘出图形

格式：impz(b,a) 其中 a 为响应的系数行向量，b 为激励的系数行向量

例：描述离散系统的差分方程为：  $y(k) - y(k-1) + 0.9y(k-2) = f(k)$

运行如下程序：a=[1 -1 0.9];b=[1];impz(b,a);

离散系统的单位响应如下图：



`impz(b,a,n)`绘出由向量  $a$  和  $b$  定义的离散系统在  $0 \sim n$  时间范围内单位响应的时域波形。

## 9. `dlsim` 函数

功能：计算离散系统的冲激响应及零状态响应并绘出相应图形

格式：`dlsim(b,a,f)` 其中  $a$  为响应的系数行向量， $b$  为激励的系数行向量， $f$  为激励序列。

## 10. `filter` 函数

功能：计算离散系统的零状态响应

格式：`filter(b,a,x)`，其中  $a$  为响应的系数行向量， $b$  为激励的系数行向量， $x$  为激励序列。

例：描述离散系统的差分方程为： $y(k) + y(k-1) + 0.25y(k-2) = f(k)$ ，激励  $f(k) = R_{16}(n)$ ，求零状态响应。

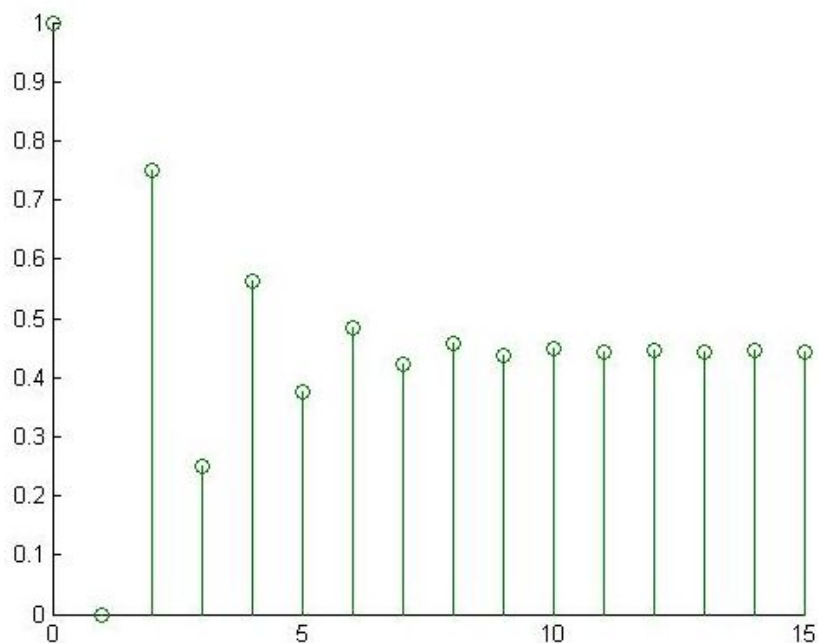
程序如下：`a=[1 1 0.25];b=[1];`

`x=ones(16);`%产生一个包含 16 个 1 的行向量  $[1 \ 1 \ 1 \dots 1 \ 1]$ 。

`y=filter(b,a,x);k=0:15;stem(k,y);`



执行结果：



## 11.fourier 函数

功能：计算连续信号的傅立叶变换

格式：F=fourier(f),即计算符号函数 f 的傅立叶变换，返回值 F 是 w 的函数。

F=fourier(f,v) 对 f(t)进行傅里叶变换，其结果为 F(v)

F=fourier(f,u,v) 对 f(u)进行傅里叶变换，其结果为 F(v)

## 12.ifourier 函数

功能：计算连续信号的傅立叶反变换

格式：f=ifourier(F,v,u) 对 F(v)进行傅里叶反变换，其结果为 f(u)

f=ifourier(F,u) 对 F(w)进行傅里叶反变换，其结果为 f(u)

f=ifourier(F) 对 F(w)进行傅里叶反变换，其结果为 f(x)

### 注意：

(1)在调用函数 fourier() 及 ifourier() 之前,要用 syms 命令对所有需要用到的变量(如 t,u,v,w) 等进行说明,即要将这些变量定义成符号变量。对 fourier() 中的 f 及 ifourier() 中的 F 也要用符号定义符 sym 将其定义为符号表达式。

(2) 采用 fourier() 及 ifourier() 得到的返回函数, 仍然为符号表达式。在对其作图时要用 ezplot() 函数, 而不能用 plot() 函数。

(3) `fourier()`及 `fourier()`函数的应用有很多局限性, 如果在返回函数中含有  $\delta(\omega)$ 等函数, 则 `ezplot()`函数也无法作出图来。另外, 在用 `fourier()`函数对某些信号进行变换时, 其返回函数如果包含一些不能直接表达的式子, 则此时当然也就无法作图了。这是 `fourier()`函数的一个局限。另一个局限是在很多场合, 尽管原时间信号  $f(t)$ 是连续的, 但却不能表示成符号表达式, 此时只能应用下面介绍的数值算法来进行傅氏变换了, 当然, 大多数情况下, 用数值算法所求的频谱函数只是一种近似值。

### 13.freqs 函数

功能: 由系统函数或微分方程得到连续系统的频率响应特性

格式:

(1) `h=freqs(b,a,w)`

该调用格式中, `a`为响应的系数行向量, `b`为激励的系数行向量, `w`为形如 `w1:p:w2`的冒号运算定义的系统频率响应的频率范围, `w1`为频率起始值, `w2`为频率终止值, `p` 为频率取样间隔。向量`h`为返回在向量`w`所定义的频率点上, 系统频率响应的样值。

(2) `[h,w]=freqs(b,a)`

该调用格式将计算默认频率范围内200个频率点的系统频率响应的样值, 并赋值给返回变量`h`, 200个频率点记录在`w`中。

(3) `[h,w]=freqs(b,a,n)`

该调用格式将计算默认频率范围内`n`个频率点上系统频率响应的样值, 并赋值给返回变量`h`, `n`个频率点记录在`w`中。

(4) `freqs(b,a)`

该格式并不返回系统频率响应的样值, 而是以对数坐标的方式绘出系统的幅频响应和相频响应曲线。

### 14.roots函数

功能: 求方程的根, 可利用此函数来求系统函数的零点和极点

格式: `p = roots(A)`, `A`为方程的系数构成的行向量, 返回向量`p`则是包含该多项

式所有根的列向量。

例:  $A(z) = z^2 + 3z + 4$

则求该多项式根的MATLAB命令为:

$A=[1 \ 3 \ 4];$

$p=\text{roots}(A)$

运行结果为:

$p=$

$-1.5000 + 1.3229i$

$-1.5000 - 1.3229i$

注意: 求离散系统函数零极点时, 系统函数有两种形式, 一种是分子和分母多项式均按未知数 $z$ 的降幂次序排列, 如式(1)所示; 另一种是分子和分母多项式均按未知数的 $z^{-1}$ 的升幂次序排列, 如式(2)所示。上述两种方式在构造多项式系数向量时稍有不同。

$$H(z) = \frac{z^3 + 2z}{z^4 + 3z^3 + 2z^2 + 2z + 1} \quad (1)$$

$$H(z) = \frac{1 + z^{-1}}{1 + 0.5z^{-1} + 0.25z^{-2}} \quad (2)$$

若 $H(z)$ 是以 $z$ 的降幂形式排列, 则系数向量一定要由多项式的最高幂次开始, 一直到常数项, 缺项要用0补齐。例如对式(1)所示的系统函数,

分子多项式的系数向量为:  $b = [1 \ 0 \ 2 \ 0]$  (缺项用0补齐)

分母多项式的系数向量为:  $a = [1 \ 3 \ 2 \ 2 \ 1]$

若 $H(z)$ 是以 $z^{-1}$ 的升幂形式排列, 则分子和分母多项式系数向量的维数一定要相同, 不足的要补0, 否则 $z=0$ 的零点或极点就可能被漏掉。例如对式(2)所示的系统函数

分子多项式系数向量应为  $b = [1 \ 1 \ 0]$  (缺项用0补齐, 以保证分子分母系数向量维数相同)

分母多项式系数向量应为  $a = [1 \ 0.5 \ 0.25]$

用  $\text{roots}()$  函数求得系统函数  $H(z)$  的零极点, 就可以用  $\text{plot}$  命令在复平面上绘制

出系统函数的零极点图，方法是在零点位置标以符号“×”，而在极点位置标以符号“o”。

## 15.tf2zp函数

功能：求系统函数的零点和极点

格式： $[q, p, c] = \text{tf2zp}(b, a)$ ，b和a分别表示H(z)的分子和分母多项式的系数向量，q和p分别为求得的零点和极点，c为H(z)中的常数。

例：用tf2zp函数求系统  $H(z) = \frac{z^{-1} + 2z^{-2} + z^{-3}}{1 - 0.5z^{-1} - 0.005z^{-2} + 0.3z^{-3}}$  的零极点。

$$H(z) = \frac{z^{-1} + 2z^{-2} + z^{-3}}{1 - 0.5z^{-1} - 0.005z^{-2} + 0.3z^{-3}} = \frac{z^2 + 2z + 1}{z^3 - 0.5z^2 - 0.005z + 0.3}$$

程序如下：

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

```
a=[1 -0.5 -0.005 0.3];
```

```
[q,p,c] = tf2zp(b,a)
```

程序运行结果为：

```
q = -1    -1
```

```
p = 0.5198 + 0.5346i    0.5198 - 0.5346i    -0.5396
```

```
c = 1
```

## 16.zplane函数

功能：得到系统函数H(z)的零极点分布图

格式： $\text{zplane}(b, a)$ ，其中b和a分别表示H(z)的分子和分母多项式的系数向量，函数的作用是在z平面上画出单位圆，零点和极点。