

# 数学建模算法与应用

# 附录A Matlab软件入门

## A.1 Matlab帮助的使用

### 1. help

**help**↵                    %帮助总揽

**help elfun**↵            %关于基本函数的帮助信息

**help exp**↵              %指数函数 exp 的详细信息

### 2.lookfor 指令

当要查找具有某种功能但又不知道准确名字的指令时，help 的能力就不够了，lookfor 可以根据用户提供的完整或不完整的关键词，去搜索出一组与之相关的指令。

**lookfor integral**↵      %查找有关积分的指令

**lookfor fourier**↵      %查找能进行傅里叶变换的指令

### 3.超文本格式的帮助用户

在 Matlab 中，关于一个函数的帮助信息可以用 `doc` 命令以超文本的方式给出，如

```
doc↵
```

```
doc doc↵
```

```
doc eig↵      %eig 求矩阵的特征值和特征向量
```

### 4.pdf 帮助文件

可从 MathWorks 网站上下载有关的 pdf 帮助文件。

## A.2 数据的输入

### 1. 简单矩阵的输入

(1) 要直接输入矩阵时，矩阵一行中的元素用空格或逗号分隔；矩阵行与行之间用分号“;”隔离，整个矩阵放在方括号“[ ]”里。

```
A=[1,2,3;4,5,6;7,8,9]↵
```

说明：指令执行后，矩阵 A 被保存在 Matlab 的工作间中，以备后用。如果用户不用 clear 指令清除它，或对它进行重新赋值，那么该矩阵会一直保存在工作间中，直到本次指令窗关闭为止。

## (2) 矩阵的分行输入

$A = \begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{bmatrix}$

## 2.特殊变量

<b>ans</b>	用于结果的缺省变量名
<b>pi</b>	圆周率
<b>eps</b>	计算机的最小数
<b>flops</b>	浮点运算次数
<b>inf</b>	无穷大 如 1/0
<b>NaN</b>	不定量 如 0/0
<b>i (j)</b>	$i=j=\sqrt{-1}$
<b>nargin</b>	所用函数的输入变量数目
<b>nargout</b>	所用函数的输出变量数目
<b>realmin</b>	最小可用正实数
<b>realmax</b>	最大可用正实数

### 3.特殊向量和特殊矩阵

#### (1) 特殊向量

`t=[0:0.1:10]`      %产生从 0 到 10 的行向量，元素之间间隔为 0.1

`t=linspace(n1,n2,n)`  
%产生 n1 和 n2 之间线性均匀分布的 n 个数 (缺省 n 时,产生 100 个数)

`t=logspace(n1,n2,n)` (缺省 n 时,产生 50 个数)  
%在 $10^{n1}$ 和 $10^{n2}$ 之间按照对数距离等间距产生 n 个数。



## (2) 特殊矩阵

### i) 单位矩阵

`eye(m)`,

`eye(m,n)` 可得到一个可允许的最大单位矩阵而其余处补 0,

`eye(size(a))` 可以得到与矩阵 `a` 同样大小的单位矩阵。

### ii) 所有元素为 1 的矩阵

`ones(n)`, `ones(size(a))`, `ones(m, n)`。

### iii) 所有元素为 0 的矩阵

`zeros(n)`, `zeros(m,n)`。

iv) 空矩阵是一个特殊矩阵,这在线性代数中是不存在的。例如

$$q=[]$$

矩阵  $q$  在工作空间之中,但它的大小为零。通过空矩阵的办法可以删除矩阵的行与列。例如

$$a(:,3)=[]$$

表示删除矩阵  $a$  的第 3 列。

## v ) 随机数矩阵

**rand(m,n)** 产生  $m \times n$  矩阵，其中的元素是服从  $[0,1]$  上均匀分布的随机数。

**normrnd(mu,sigma,m,n)** 产生  $m \times n$  矩阵，其中的元素是服从均值为  $\mu$ ，标准差为  $\sigma$  的正态分布的随机数。

**exprnd(mu,m,n)** 产生  $m \times n$  矩阵，其中的元素是服从均值为  $\mu$  的指数分布的随机数。

**poissrnd(mu,m,n)** 产生  $m \times n$  矩阵，其中的元素是服从均值为  $\mu$  的泊松 (Poisson) 分布的随机数。

**unifrnd(a,b,m,n)** 产生  $m \times n$  矩阵，其中的元素是服从区间  $[a,b]$  上均匀分布的随机数。

## vi) 随机置换

`randperm(n)`产生 1 到 n 的一个随机全排列。

`perms([1:n])`产生 1 到 n 的所有全排列。

## A.3 绘图命令

### A.3.1 二维绘图命令

二维绘图的基本命令有 `plot`, `loglog`, `semilogx`, `semilogy` 和 `polar`。它们的使用方法基本相同，其不同特点是在不同的坐标中绘制图形。`plot` 命令使用线性坐标空间绘制图形；`loglog` 命令在两个对数坐标空间中绘制图形；而 `semilogx`(或 `semilogy`)命令使用  $x$  轴（或  $y$  轴）为对数刻度，另外一个轴为线性刻度的坐标空间绘制图形；`polar` 使用极坐标空间绘制图形。

二维绘图命令 `plot` 为了适应各种绘图需要，提供了用于控制线色、数据点和线型的 3 组基本参数。它的使用格式如下：

`plot(x,y,'color_point_linestyle')`

该命令是绘制 `y` 对应 `x` 的轨迹的命令。`y` 与 `x` 均为向量，且具有相同的元素个数。用字符串 '`color_point_linestyle`' 完成对上面 3 个参数的设置。线色 (`r-red`, `g-green`, `b-blue`, `w-white`, `k-black`, `i-invisible`, `y-yellow`), 数据点 (`.`, `o`, `x`, `+`, `*`, `S`, `H`, `D`, `V`, `^`, `>`, `<`, `p`) 与线型 (`-`, `-.`, `--`, `:`) 都可以根据需要适当选择。

当 `plot(x,y)` 中的 `x` 和 `y` 均为  $m \times n$  矩阵时，`plot` 命令将绘制  $n$  条曲线。

`plot(t,[x1,x2,x3])` 在同一坐标轴内同时绘制三条曲线。

如果多重曲线对应不同的向量绘制，可使用命令 `plot(t1,x1,t2,x2,t3,x3)`

式中 `x1` 对应 `t1`，`x2` 对应 `t2` 等等。在这种情况下，`t1`，`t2` 和 `t3` 可以具有不同的元素个数，但要求 `x1`，`x2` 和 `x3` 必须分别与 `t1`，`t2` 和 `t3` 具有相同的元素个数。

**subplot** 命令使得在一个屏幕上可以分开显示  $n$  个不同坐标系，且可分别在每一个坐标系中绘制曲线。其命令格式如下

**subplot(r,c,p)**

该命令将屏幕分成  $r \times c$  个子窗口，而  $p$  表示激活第  $p$  个子窗口。窗口的排号是从左到右，自上而下。



在图形绘制完毕后，执行如下命令可以再在图中加入标题、标号、说明和分格线等。这些命令有 `title`, `xlabel`, `ylabel`, `text`, `gtext` 等。它们的命令格式如下

```
title('My Title'), xlabel('My X-axis Label'), ylabel('My  
Y-axis Label'),
```

```
text(x,y,'Text for annotation'),
```

```
gtext('Text for annotation'), grid
```

`gtext` 命令是使用鼠标器定位的文字注释命令。当输入命令后，可以在屏幕上得到一个光标，然后使用鼠标器控制它的位置。按鼠标器的左键，即可确定文字设定的位置。

**hold on** 是图形保持命令，可以把当前图形保持在屏幕上不变，同时在这个坐标系内绘制另外一个图形。**hold** 命令是一个交替转换命令，即执行一次，转变一个状态（相当于**hold on**、**hold off**）。

### A.3.2 显函数，符号函数或隐函数的绘图

**fplot(fun,lims)**绘制由字符串 **fun** 指定函数名的函数在 **x** 轴区间为 **lims=[xmin, xmax]**的函数图。若 **lims=[xmin,xmax,ymin,ymax]**，则 **y** 轴也被限制。

**例 A.1** 画  $f(x) = \begin{cases} x+1, & x < 1 \\ 1+\frac{1}{x}, & x \geq 1 \end{cases}$  的图形。

解 (1) 首先用 M 文件 Afun1.m 定义函数  $f(x)$   
如下

```
function y=Afun1(x);  
if x<1  
    y=x+1;  
else  
    y=1+1/x;  
end
```

在 matlab 命令窗口输入

```
fplot('Afun1',[-3,3])
```

就可画出函数  $f(x)$  的图形。

(2) 这里也可以使用匿名函数，编写程序如下

```
Afun2=@(x) (x+1)*(x<1)+(1+1/x)*(x>=1);  
fplot(Afun2,[-3,3])
```

**ezplot(f)**绘制符号函数  $f$  的图形，横轴的近似范围为  $[-2\pi, 2\pi]$ 。

**ezplot(f,[xmin,xmax])**使用输入参数来代替默认横坐标范围  $[-2\pi, 2\pi]$ 。

**例 A.2** 画出函数  $y = \tan x$  的图形

**解** `ezplot('tan(x)')`

**ezplot** 也可以绘制隐函数的图形。

**例 A.3** 画出椭圆  $x^2 + \frac{y^2}{4} = 1$  的图形。

**解** `ezplot('x^2+y^2/4=1')`



### A.3.3 三维图形

在实际工程计算中，最常用的三维绘图是三维曲线图、三维网格图和三维曲面图 3 种基本类型。与此对应，Matlab 也提供了一些三维基本绘图命令，如三维曲线命令 `plot3`，三维网格图命令 `mesh` 和三维表面图命令 `surf`。

## 1. 三维曲线

`plot3(x,y,z)`通过描点连线画出曲线，这里  $x,y,z$  都是  $n$  维向量，分别表示该曲线上点集的横坐标、纵坐标、竖坐标。

**例 A.4** 在区间 $[0,10\pi]$ 画出参数曲线  $x = \sin(t)$ ,  $y = \cos(t)$ ,  $z = t$ ，并分别标注。

## 2. 网格图

命令 `mesh(x,y,z)`画网格曲面。这里 `x,y,z` 是三个同维数的数据矩阵,分别表示数据点的横坐标、纵坐标、竖坐标,命令 `mesh(x,y,z)`将该数据点在空间中描出,并连成网格。

**例 A.5** 绘制二元函数

$$z = \frac{\sin(xy)}{xy}$$

的三维网格图。

### 3. 表面图

命令 `surf(x,y,z)` 画三维表面图，这里 `x,y,z` 是三个同维数的数据矩阵，分别表示数据点的横坐标、纵坐标、竖坐标。

例 A.6 绘制二元函数

$$z = \frac{\sin(xy)}{xy}$$

的三维表面图。

## 4. 旋转曲面

例A.7 画出

$$y = 30e^{-\frac{x}{400}} \sin\left(\frac{1}{100}(x + 25\pi)\right) + 130, \quad x \in [0, 600]$$

绕  $x$  轴旋转一周形成的旋转曲面。

**例 A.8** 画出  $x^2 + (y - 5)^2 = 16$  绕  $x$  轴旋转一周所形成的旋转曲面。

**解** 因为这里的函数是隐函数，化成显函数后有两支，必须使用参数方程，旋转面的参数方程为

$$x = 4 \cos \alpha,$$

$$y = (5 + 4 \sin \alpha) \cos \beta$$

$$z = (5 + 4 \sin \alpha) \sin \beta$$

其中  $\alpha, \beta \in [0, 2\pi]$ 。(程序略)

## 5. 其它二次曲面

**Matlab 中使用绘图命令 `ezmesh` 或 `ezsurf` 也很方便, 只需要把曲面方程写成两个变量的显函数方程或参数方程即可。**

### 例 A.9 画出下列曲面的图形

(1) 旋转单叶双曲面  $\frac{x^2 + y^2}{9} - \frac{z^2}{4} = 1;$

(2) 旋转双叶双曲面  $\frac{x^2}{9} - \frac{y^2 + z^2}{4} = 1;$

(3) 抛物柱面  $y^2 = x;$

(4) 椭圆锥面  $\frac{x^2}{9} + \frac{y^2}{4} = z^2;$

(5) 椭球面  $\frac{x^2}{9} + \frac{y^2}{4} + \frac{z^2}{6} = 1;$

(6) 马鞍面  $z = xy;$

(7) 椭圆柱面  $\frac{x^2}{9} + \frac{y^2}{4} = 1.$



### A.3.4 3-D可视化图形

给出 Matlab 帮助中的一个例子。

#### 例 A.10

```
[x,y,z,v] = flow;  
isosurface(x,y,z,v);
```

**例 A.11** 画出  $v = x^2 y(z + 1)$  的示意图。

## A.4 Matlab在高等数学中应用

### A.4.1 求极限

**Matlab 求极限的命令为**

**limit(expr, x, a)**

**limit(expr, a)**

**limit(expr)**

**limit(expr, x, a, 'left')**

**limit(expr, x, a, 'right')**

其中 **limit(expr, x, a)**表示求符号表达式 **expr** 关于符号变量 **x** 趋近于 **a** 时的极限，**limit(expr)**求表示缺省变量趋近于 0 时的极限。

**例 A.12** 求下列表达式的极限

$$(1) \lim_{x \rightarrow 0} \frac{\sqrt{1+x^2} - 1}{1 - \cos x}, \quad (2) \lim_{x \rightarrow +\infty} \left(1 + \frac{a}{x}\right)^x.$$

## A.4.2 求导数

### 建模与求解

Matlab 的求导数命令为

`diff(expr)`

`diff(expr, v)`

`diff(expr, sym('v'))`

`diff(expr, n)`

`diff(expr, v, n)`

`diff(expr, n, v)`

其中 `diff(expr)`表示求表达式 `expr` 关于默认变量的 1 阶导数，`diff(expr, v, n)`和 `diff(expr, n, v)`都表示求表达式 `expr` 关于符号变量 `v` 的 `n` 阶导数。

**例 A.13**    (1) 求函数  $y = \ln \frac{x+2}{1-x}$  的三阶导数;  
(2) 求向量  $a = [0 \quad 0.5 \quad 2 \quad 4]$  的一阶向前差分。

### A.4.3 求极值

**例 A.14** 求函数  $f(x) = x^3 + 6x^2 + 8x - 1$  的极值点，并画出函数的图形。

## A.4.4 求积分

### 1. 求不定积分

Matlab 求符号函数不定积分的命令为

**int(expr)**

**int(expr, v)**



**例 A.15 求不定积分**

$$\int \frac{1}{1 + \sqrt{1 - x^2}} dx.$$

**解**    `syms x`

`I=int(1/(1+sqrt(1-x^2)))`

`pretty(I)`

## 2. 求定积分

### (1) 求定积分的符号解

Matlab 求符号函数的定积分命令为

`int(expr, a, b)`

`int(expr, v, a, b)`

**例 A.16** 求定积分

$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos x \cos 2x dx$$

## (2) 求定积分的数值解

**例 A.17** 求下列积分的数值解

i )  $\int_2^{+\infty} \frac{dx}{x \cdot \sqrt[3]{x^2 - 3x + 2}};$

ii )  $\iint_D \sqrt{1 - x^2 - y^2} d\sigma, \quad D = \{(x, y) \mid x^2 + y^2 \leq x\};$

iii )  $\iiint_{\Omega} \frac{z^2 \ln(x^2 + y^2 + z^2 + 1)}{x^2 + y^2 + z^2 + 1} dv, \quad ,$

$$\Omega = \left\{ (x, y, z) \mid 0 \leq z \leq \sqrt{1 - x^2 - y^2} \right\}.$$

## A.4.5 级数求和

**Matlab 级数求和的命令为**

**`r = symsum(expr, v)`**

**`r = symsum(expr, v, a, b)`**

**其中 `expr` 为级数的通项表达式，`v` 是求和变量，`a` 和 `b` 分别为求和变量的起始点和终止点，若没有指明 `a` 和 `b`，`a` 的默认值为 0，`b` 的默认值为 `v-1`。**

**例 A.18** 求如下级数的和

$$(1) \sum_{n=1}^{\infty} \frac{2n-1}{2^n}, \quad (2) \sum_{n=1}^{\infty} \frac{1}{n^2}.$$

## A.5 Matlab在线性代数中的应用

### A.5.1 向量组的线性相关性

求列向量组  $A$  的一个最大线性无关组可用命令 `rref(A)` 将  $A$  化成行最简形，其中单位向量对应的列向量即为最大线性无关组所含向量，其它列向量的坐标即为其对应向量用最大线性无关组线性表示的系数。

**例 A.19** 求下列矩阵列向量组的一个最大无关组。

$$A = \begin{bmatrix} 1 & -2 & -1 & 0 & 2 \\ -2 & 4 & 2 & 6 & -6 \\ 2 & -1 & 0 & 2 & 3 \\ 3 & 3 & 3 & 3 & 4 \end{bmatrix}$$

**例 A.20** 设  $A = [a_1, a_2, a_3] = \begin{bmatrix} 2 & 2 & -1 \\ 2 & -1 & 2 \\ -1 & 2 & 2 \end{bmatrix}$ ,

$$B = [b_1, b_2] = \begin{bmatrix} 1 & 4 \\ 0 & 3 \\ -4 & 2 \end{bmatrix},$$

验证 $a_1, a_2, a_3$ 是 $R^3$ 的一个基, 并把 $b_1, b_2$ 用这个基线性表示。



## A.5.2 齐次线性方程组

在 Matlab 中，函数 `null` 用来求解零空间，即满足  $Ax = 0$  的解空间，实际上是求出解空间的一组基（基础解系）。格式如下

`z=null(A)`    %z 的列向量为方程组的正交规范基，满足  $z^T z = E$ 。

`z=null(A,'r')`    %z 的列向量是方程  $Ax = 0$  的有理基。

**例 A.21** 求方程组的通解

$$\begin{cases} x_1 + 2x_2 + 2x_3 + x_4 = 0, \\ 2x_1 + x_2 - 2x_3 - 2x_4 = 0, \\ x_1 - x_2 - 4x_3 - 3x_4 = 0. \end{cases}$$

### A.5.3 非齐次线性方程组

**Matlab** 中解非齐次线性方程组可以使用“`\`”。虽然表面上只是一个简简单单的符号，而它的内部却包含许许多多的自适应算法，如对超定方程（无解）用最小二乘法，对欠定方程（多解）它将给出范数最小的一个解。

另外求解欠定方程组（多解）可以使用求矩阵  $A$  的行最简形命令 `rref(A)`，求出所有的基础解系。

## 例 A.22 求超定方程组

$$\begin{cases} 2x_1 + 4x_2 = 11, \\ 3x_1 - 5x_2 = 3, \\ x_1 + 2x_2 = 6, \\ 2x_1 + x_2 = 7. \end{cases}$$

**例 A.23** 用最小二乘解法解方程组

$$\begin{cases} x_1 + x_2 = 1, \\ x_1 + x_3 = 2, \\ x_1 + x_2 + x_3 = 0, \\ x_1 + 2x_2 - x_3 = -1. \end{cases}$$

### 例 A.24 求解方程组

$$\begin{cases} x_1 - x_2 - x_3 + x_4 = 0, \\ x_1 - x_2 + x_3 - 3x_4 = 1, \\ x_1 - x_2 - 2x_3 + x_4 = -1/2. \end{cases}$$

## A.5.4 相似矩阵及二次型

例 A.25 求一个正交变换  $x = Py$ ，把二次型

$$f = 2x_1x_2 + 2x_1x_3 - 2x_1x_4 - 2x_2x_3 + 2x_2x_4 + 2x_3x_4$$

化为标准形。

**例 A.26** 判别二次型  $f = 2x_1^2 + 4x_2^2 + 5x_3^2 - 4x_1x_2$  的正定性，并求正交变换把二次型化成标准型。



## **A.6 数据处理**

### **A.6.1 Matlab中的默认数据文件mat文件**

**例 A.27** 把 Matlab 工作空间中的数据矩阵 a, b, c 保存到数据文件 data1.mat 中。

```
save data1 a b c
```

**注：**Matlab 中的默认数据文件 mat 文件可以省略后缀名。

**例 A.28 把例 A.27 生成的 data1.mat 中的所有数据加载到 Matlab 工作空间中。**

```
load data1
```

## A.6.2 纯文本文件

可以把 word 文档中整行整列的数据粘贴到纯文本文件，然后调入到 Matlab 工作空间中。

例 A.29 把纯文本文件 data2.txt 加载到工作空间。

```
a=load('data2.txt');
```

或者是

```
a=textread('data2.txt');
```

**例 A.30** 使用 `dlmwrite` 命令把矩阵 `b` 保存到纯文本文件 `data3.txt`。

```
dmlwrite('data3.txt',b)
```

**例 A.31** 生成服从标准正态分布随机数的 $100 \times 200$ 矩阵，然后用 `fprintf` 命令保存到纯文本文件 `data4.txt`。

解 `clc, clear`

```
fid=fopen('data4.txt','w');
```

```
a=normrnd(0,1,100,200);
```

```
fprintf(fid,'%f\n',a);
```

```
fclose(fid);
```

注：对于高维矩阵，用 `dlmwrite` 构造的纯文本文件，Lingo 软件不识别；为了 Lingo 软件识别，纯文本文件必须用 `fprintf` 构造，而且数据之间的分割符为“`\n`”。

### A.6.3 Excel文件

**例 A.32** 把一个 $5 \times 10$ 矩阵写到 Excel 文件 data5.xls 表单 Sheet2 中 B2 开始的域中。

```
a=rand(5,10);
```

```
xlswrite('data5.xls',a,'Sheet2','B2')
```

**例 A.33** 把例 A.32 生成的 Excel 文件 data5.xls 中表单 Sheet2 的域“C3： F6”中的数据赋给 b。

```
b=xlsread('data5.xls','Sheet2','C3:F6')
```

## A.6.4 字符串数据

**例 A.34** 统计下列五行字符串中字符 a、c、g、t 出现的频数。

1.aggcacggaaaaacgggaataacggaggaggacttggcacggcatta  
cacggagg

2.cggaggacaaacgggatggcggtattggagggtggcggactgttcggggg  
a

3.gggacggatacggattctggccacggacgggaaaggaggacacggcg  
acataca

4.atggataacggaaacaaaccagacaaacttcggtagaaatacagaagc  
tta

5.cggctggcggaacaacggactggcggttccaaaaacggaggaggcg  
acggaggc



## A.6.5 图像文件

**例 A.35** 把一个比较大的 bmp 图像文件 data6.bmp，转化成比较小的 jpg 文件，命名成 data7.jpg，并显示。

**解**

```
a=imread('data6.bmp');  
imshow(a)  
imwrite(a,'data7.jpg');  
figure, imshow('data7.jpg')
```

**例 A.36** 生成 10 幅彩色 jpg 文件，依次命名成  
jpg1.jpg, ..., jpg10.jpg。

```
clc, clear
for i=1:10
    str=['jpg',int2str(i),'.jpg'];
    a(:,:,1)=rand(500);    a(:,:,2)=rand(500)+100;
a(:,:,3)=rand(500)+200;
    imwrite(a,str);
end
```

## **A.6 数据处理**

### **A.6.1 Matlab 中的默认数据文件 mat 文件**

**例 A.27** 把 Matlab 工作空间中的数据矩阵 a, b, c 保存到数据文件 data1.mat 中。

```
save data1 a b c
```

**注：**Matlab 中的默认数据文件 mat 文件可以省略后缀名。

**例 A.28** 把例 A.27 生成的 data1.mat 中的所有数据加载到 Matlab 工作空间中。

```
load data1
```

## A.6.2 纯文本文件

可以把 word 文档中整行整列的数据粘贴到纯文本文件，然后调入到 Matlab 工作空间中。

**例 A.29** 把纯文本文件 data2.txt 加载到工作空间。

```
a=load('data2.txt');
```

或者是

```
a=textread('data2.txt');
```

**例 A.30** 使用 `dlmwrite` 命令把矩阵 `b` 保存到纯文本文件 `data3.txt`。

```
dmlwrite('data3.txt',b)
```

**例 A.31** 生成服从标准正态分布随机数的 $100 \times 200$ 矩阵，然后用 `fprintf` 命令保存到纯文本文件 `data4.txt`。

**解** `clc, clear`

`fid=fopen('data4.txt','w');`

`a=normrnd(0,1,100,200);`

`fprintf(fid,'%f\n',a);`

`fclose(fid);`

**注：**对于高维矩阵，用 `dlmwrite` 构造的纯文本文件，Lingo 软件不识别；为了 Lingo 软件识别，纯文本文件必须用 `fprintf` 构造，而且数据之间的分割符为“`\n`”。

### A.6.3 Excel 文件

**例 A.32** 把一个 $5 \times 10$ 矩阵写到 Excel 文件 data5.xls 表单 Sheet2 中 B2 开始的域中。

```
a=rand(5,10);  
xlswrite('data5.xls',a,'Sheet2','B2')
```



**例 A.33** 把例 A.32 生成的 Excel 文件 data5.xls 中  
表单 Sheet2 的域“C3: F6”中的数据赋给 b。

**b=xlsread('data5.xls','Sheet2','C3:F6')**

## A.6.4 字符串数据

**例 A.34** 统计下列五行字符串中字符 a、c、g、t 出现的频数。

**1.aggcacggaaaaacgggaataacggaggaggacttggcacggcatta  
cacggagg**

**2.cggaggacaaacgggatggcggtattggagggtggcggactgttcggggg  
a**

**3.gggacggatacggattctggccacggacggaaaggaggacacgggcgg  
acataca**

**4.atggataacggaaacaaaccagacaaacttcggtagaaatacagaag  
ctta**

**5.cggctggcggaacaacggactggcggtattccaaaaacggaggaggcgg  
acggaggc**

## A.6.5 图像文件

**例 A.35** 把一个比较大的 bmp 图像文件 data6.bmp, 转化成比较小的 jpg 文件, 命名成 data7.jpg, 并显示。

**例 A.36** 生成 10 幅彩色 jpg 文件，依次命名成 jpg1.jpg, ..., jpg10.jpg。

## **A.6.6 数据的批处理**

**例 A.37** 现有数据文件 `book1.xls`, ..., `book5.xls`, 用命令 `importdata` 读入数据。

**例 A.38** 现有数据文件 book01.xls, ..., book05.xls, 读入各 Excel 文件的第 1 个表单(Sheet1)的域“A2:G10”的数据。

**例 A.39** 读入当前目录下所有后缀名为 xls 的 Excel 文件的数据。

## A.6.7 时间序列数据

例 A.40 时间序列数据的处理。



**例 A.41** 对于 Matlab 当前工作路径下所有时间序列型的纯文本文件，进行相关的数据操作。

## **A.6.8 日期和时间**

**Matlab 日期和时间的函数有 datenum, datevec, datestr, now, clock, date, calendar, eomday, weekday, addtodate, etime 等，这里就不一一说明各个函数的用法了，下面举一个小例子说明有关函数的使用。**

**例 A.42** 统计 1601 年 1 月到 2000 年 12 月，每月的 13 日分别出现在星期日、星期一、星期二，...，星期六的频数，并画出对应的柱状图。

**注：**Matlab 中 weekday 的 1 对应“星期日”，2 对应“星期一”，...，7 对应“星期六”。

解 所画出的柱状图见图 1，

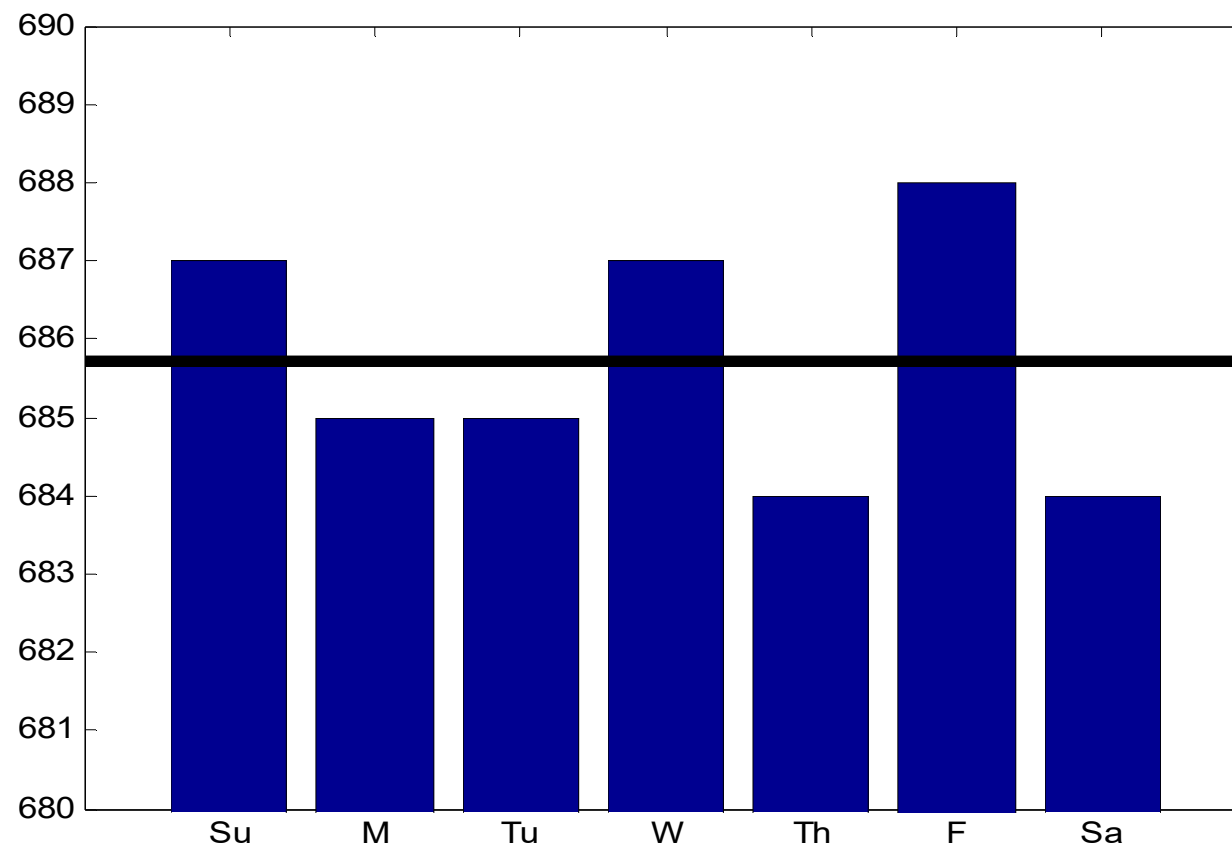


图 1 频数分布的柱状图

## A.6.9 视频文件

**Matlab** 除了支持各种图像文件的读写等操作,还支持视频文件的相应处理。实际上,视频文件本质上是由多帧具有一定大小、顺序、格式的图像组成的,只是一般的图像是静止的,而视频是可以将多帧静止的图像进行连续显示,从而达到动态效果。

**例 A.43** 读取一个视频文件 `test.avi`, 并把视频中的每一帧保存成 `jpg` 文件。