



MATLAB语言初步

- 第一讲 MATLAB工作界面
- 第二讲 矩阵的赋值
- 第三讲 矩阵的四则运算
- 第四讲 元素群运算
- 第五讲 基本绘图方法
- 第六讲 程序文件



第一讲 MATLAB语言概述

- 1.1 MATLAB语言的发展
- 1.2 MATLAB语言的特点
- 1.3 MATLAB的工作环境
- 1.4 演示程序



科学计算工具的发展

- 算盘是最早的计算工具
- 1630年发明了计算尺，它以后成为大学理工科学生的必备工具；
- 1970年发明了计算器，在中国则从1980年起，计算尺迅速被计算器所取代；
- 计算机取代计算器的过程从1980年开始，以科学计算语言出现为标志。在发达国家大学中已经完成，中国尚需努力。
- 线性代数是计算机显示其优越性的最早发源地，可惜的是很多老师墨守经典，无视现代。我们希望本课程的听众能成为时代进步的实践者和宣传员。



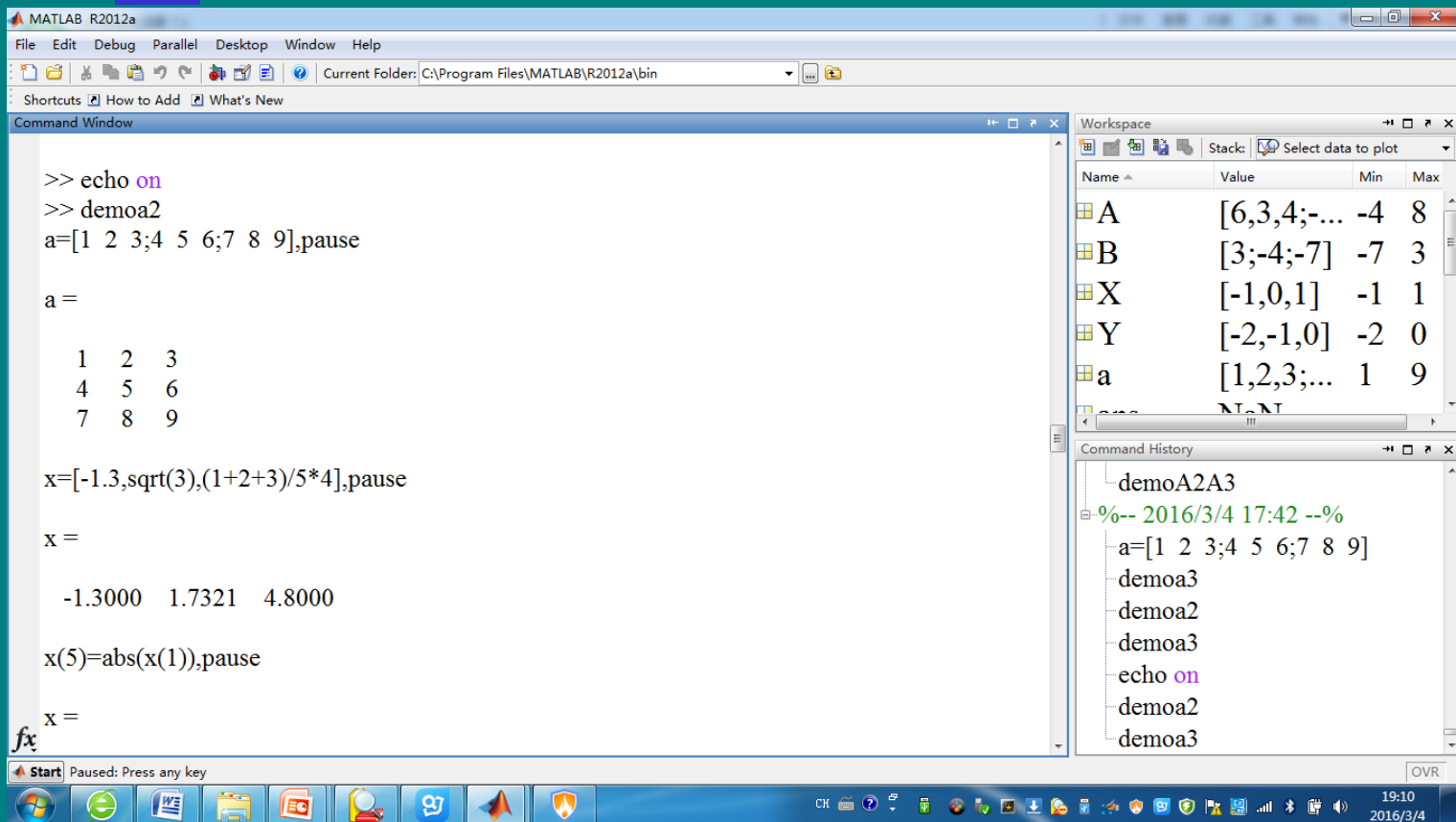
MATLAB语言的特点

MATLAB 是 ” 矩 阵 实 验 室 ” （ MATrix LABoratory）的缩写，是以矩阵运算为基础的交互式程序语言。

其主要特点是：

- 1. 起点高
- 2. 人机界面适合科技人员
- 3. 强大而简易的作图功能
- 4. 智能化程度高
- 5. 功能丰富,可扩展性强

MATLAB的启动界面





MATLAB的主要视窗

在默认条件下其主要视窗有：

- 命令窗(显示当前的命令和运行结果)
- 工作空间(记录和显示存储的变量)
- 历史命令窗(记录以前输入的命令)
- 文本编辑窗 (用来打开和编辑程序)；
- 图形窗（输入绘图命令时自动生成），

本书着重介绍命令窗

文本编辑窗和图形窗

Figure 2

```
1 - a=[1 2 3;4 5 6;7 8 9],pause
2 - x=[-1.3,sqrt(3),(1+2+3)/5*4
3 - x(5)=abs(x(1)),pause
4 - a(4,3)=6.5,pause
5 - a(5,:)=[5,4,3],pause
6 - b=a([2,4],[1,3]),pause
7 - a([2,4,5],:)=[],pause
8 - a',pause
9 - f1=ones(3,2),pause
10 - f2=zeros(2,3),pause
11 - f3=magic(3),pause
12 - f4=eye(2),pause
13 - fb1=[f1,f3;f4,f2],pause
14 - who,pause
15 - whos,pause
16 - 0/0,pause
17 - 1/0 pause
```

Value

[6,3,4;-...
[3;-4;-7]
<51x51...
[0.130...
[-1,0,1]

figure
echo off
logo

Start

幻灯片 6/71 "Blend"

19:38
2016/3/4



命令窗的主要布局

- ，主菜单各项的功能：(与WORD相同处)
- 【file】 中的New,Open,...,等，用来打开文本编辑窗。path用来把外加程序集放入MATLAB.
- 【edit】 中的Cut,Copy,Paste,...,等；
- 【desktop】 桌面布局，常用default状态
- 【window】 显示当前打开的各视窗；
- 【help】 进入帮助视窗。



命令窗常用命令和显示功能

- 对大小写敏感，即把a和A看成不同字母；
- format 命令：改变显示格式
- help 命令：给出MATLAB当前管理的所有程序库名；向下依次给出MATLAB中每一个函数库的函数用法；
- demo(演示命令)
- exit或quit命令：正常退出MATLAB。



format控制数的显示格式

- 数的显示格式控制：例如

format short (默认) 如 $\pi=3.1416$

format long 如 $\pi=3.14159265358979$

format rat 如 $\pi=355/113$

- 页面显示的疏密控制

format compact 密集显示格式

format loose (默认) 稀疏显示格式



help 命令

- (1) help:
 - 得到全部子程序和工具箱库名。
- (2) help 库名 (如help dskpla)
 - 得到库中全部子程序名。
- (3) help 子程序名 (如help format)
 - 得到该子程序的功能与用法。



本书程序集



本课程提供的程序集

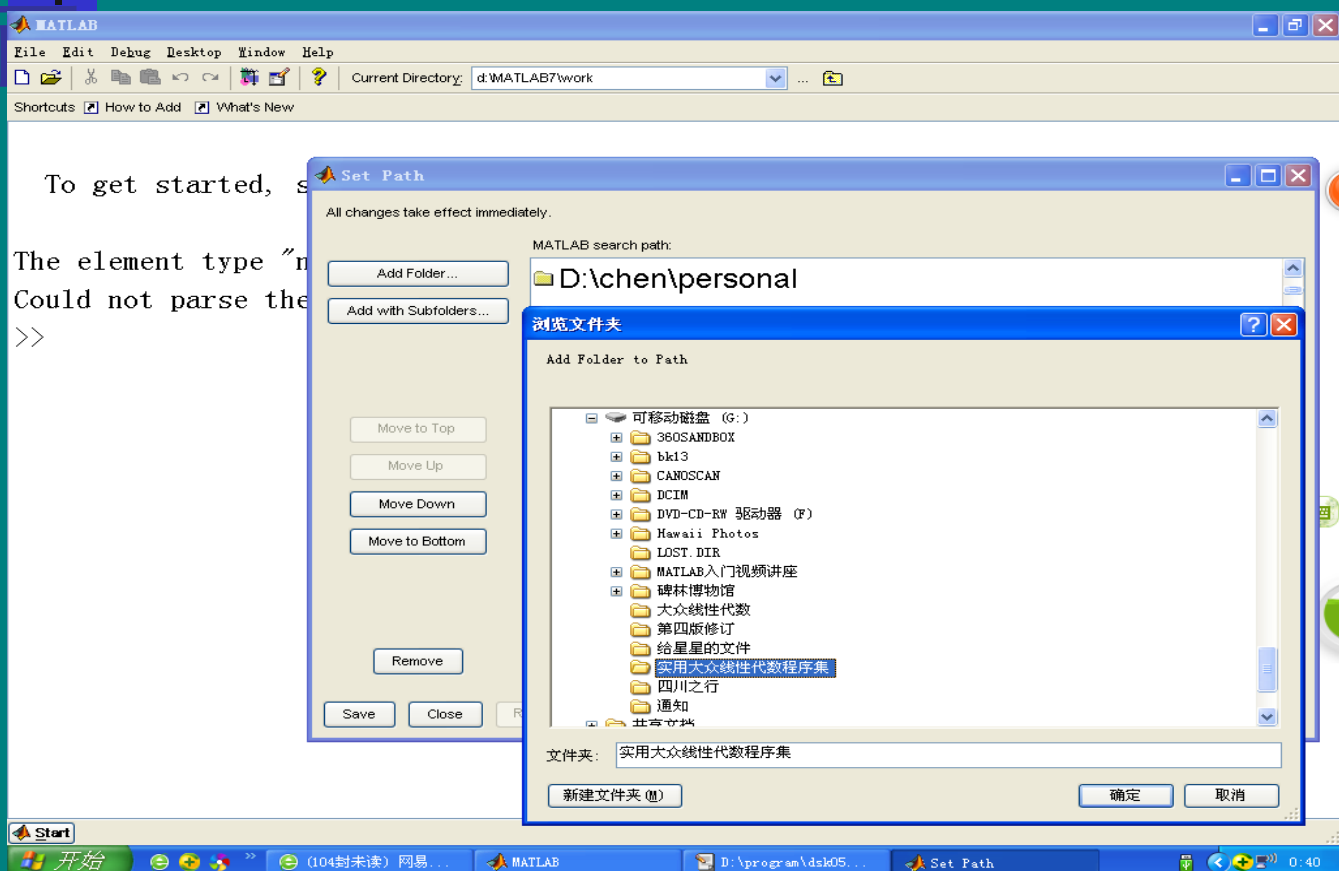
- 我们这门课程将提供读者一个下载压缩文件 dskpla.rar，其中包括100多个MATLAB程序。读者将它下载解压后，存在你的文件系统中，按下述方法将它置于MATLAB搜索路径的某个目录下。以后你就可以运行本课程中介绍的任何题目。(此文件的中文名是“实用大众线性代数程序集.rar”，但我们发现有的MATLAB版本不接受汉字的路径名，为了可靠，建议大家还是用英文的文件名，且其上级路径也最好是英文名)



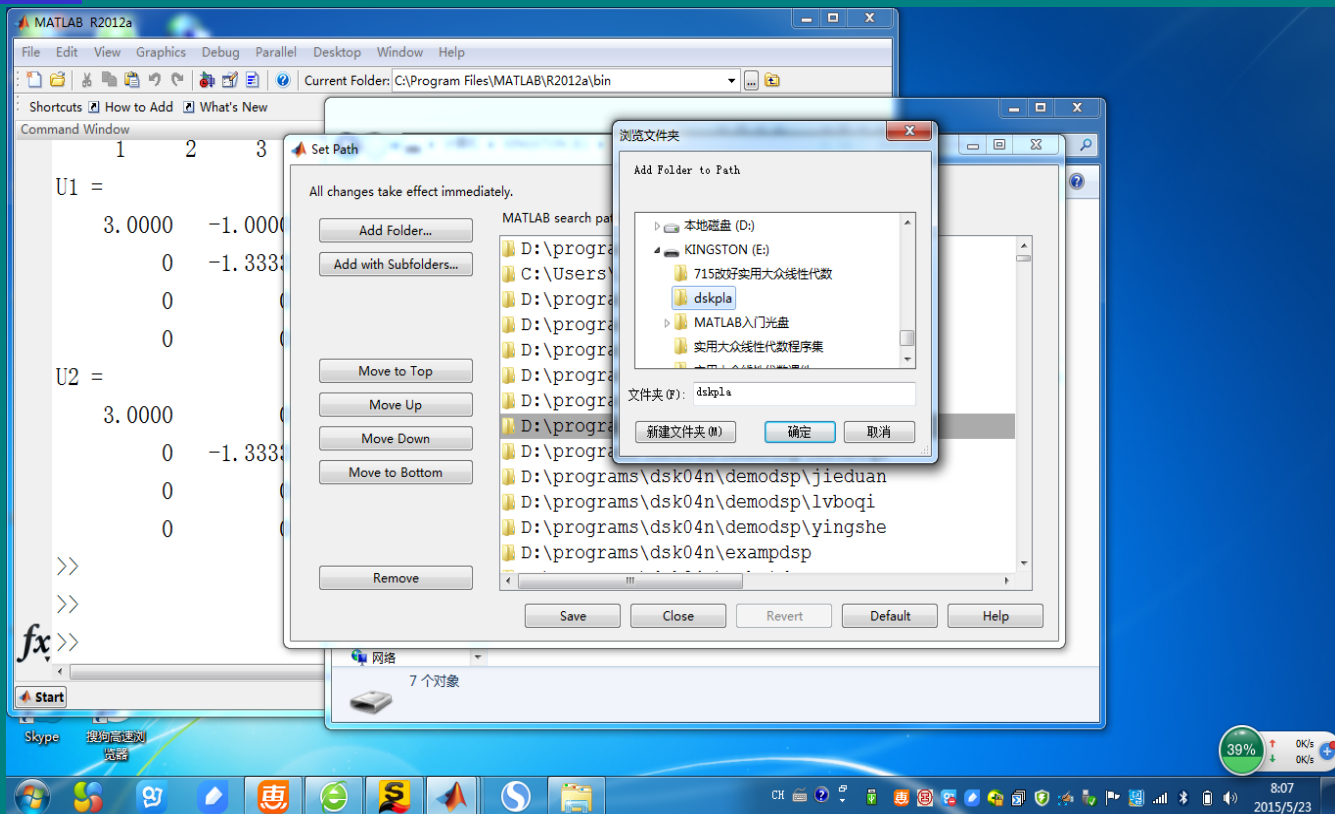
将自编函数库放入MATLAB

- (1) 首先要把这些函数库拷贝进计算机硬盘的一个目录下(拷在U盘中插入USB接口也行)。
- (2) 在命令窗中单击【File】→【Set Path】，就会出现如图A-7所示的【Set Path】对话框。该对话框左侧是一排按钮。选择【Add with Subfolders】等。此时将弹出一个系统文件浏览框，即图A-7右下角的小框。在其中找到该文件夹，选中它，再按【确定】，小框即关闭。然后，再在【Set Path】对话框中，按【Save】和【Close】
- 有的MATLAB版本不认汉字，所以最好程序库名用英文，可把本书的程序库名不用“实用大众线性代数程序集”而改为dskpla。在我们的下载文件中，这也是一个选项。

将自编函数库放入MATLAB的界



搜索路径(path)设定





演示命令

- 键入demo，可以得到MATLAB及其所有工具箱中的全部演示程序约数百个。有些演示的内容是很深的。我们只演示几个简单程序，使读者了解MATLAB的图形和多媒体功能。



退出MATLAB

- Exit或quit命令：正常退出MATLAB。退出后整个进程中输入的命令都保存在历史命令中，下次可以利用。
- 点击右上方×号：强制非正常退出MATLAB。进程中输入的命令都不保存（建议别用）
- Save命令：在退出前保存数据，下次用load命令再调出数据。





MATLAB中的矩阵运算

- 1 矩阵变量及其赋值
- 2 矩阵的初等运算
- 3 用MATLAB做行阶梯变换
- 4 用MATLAB解线性方程组



变量及其赋值

- (1) 标识符与数
- (2) 矩阵及其元素的赋值
- (3) 复数
- (4) 变量检查
- (5) 基本赋值矩阵



一、标识符与数

- 标识符：标识符是标志变量名、常量名、函数名和文件名的字符串的总称。
- **MATLAB**中的标识符最长允许19个字符，合法字符是52个英文字母（大小写看成不同字符）、10个数字和下划线。
- 第一个字符必须是英文字母。



MATLAB中的数

- MATLAB中的实数只有一种数据格式，那就是双精度（即64位二进制或8个字节），它的有效值是十进制16位，动态范围是10的 ± 308 次幂。

$$2.2251 \times 10^{-308} \sim 1.7977 \times 10^{+308}$$

数的存储和运算全按同一格式进行，使编程简单，减少差错，改善人机交互，付出代价是内存空间和运行速度，形成MATLAB的一大特色。



二、矩阵及其元素的赋值

- 格式1：变量=表达式（或常量）
- 格式2：表达式

表达式中的运算符有：加(+)、减(-)、乘(*)、左除(\)、右除(/)、指数(^)、共轭转置(')、矩阵([])等。

表达式的结尾标点：若为分号‘;’，不显示；
若为逗号‘,’或直接回车，显示运算结果。



三、复数

- 虚数符号MATLAB启动时定为i,j，可以不用乘号，连写在数字后面。
- 如果用户在程序中另外给i,j赋值，则它们的虚数意义就失效。
- `conj(x)`表示共轭，即把x的虚部反号。
- ‘`'`是共轭转置运算符，对实矩阵把行号与列号交换，对复矩阵除行列交换外，还要把矩阵元素取共轭。



四、变量检查

- 变量检查命令who, whos
- 打开工作空间视窗
- 在检查中不显示内部变量

eps, realmax, realmin, pi

i, j, inf, NaN



五、特殊矩阵赋值函数

基 本 矩 阵	zeros(m,n)	全0矩阵($m \times n$ 阶)	linspace	均分向量($1 \times n$ 维数组)
	ones(m,n)	全1矩阵($m \times n$ 阶)	logspace	对数均分向量($1 \times n$ 维数组)
	rand(m,n)	随机数矩阵($m \times n$ 阶)	meshgrid	画三维曲面时的X, Y网格
	randn(m,n)	正态随机数矩阵($m \times n$ 阶)	magic(n)	魔方矩阵
	eye(n)	单位矩阵(方阵)	size(A)	多维矩阵的各维长度
	length	一维矩阵的长度	diag	提取或建立对角阵
结构变换	fliplr	矩阵左右翻转	tril	取矩阵的左下三角部分
	flipud	矩阵上下翻转	triu	取矩阵的右上三角部分



矩阵组合与显示

- 大矩阵可由若干个小矩阵组成，但必须其行列数正确，恰好填满全部元素。
- 一个矩阵中所有元素用同一显示格式。有一个是小数则都用小数；
- 当矩阵中的最大元素小于0.001，或其最小元素大于1000时，MATLAB会将其公因子提出来。
- 元素大小差别很大时，不宜用矩阵显示。



矩阵的初等运算



2.2 矩阵的初等运算

- (1) 矩阵的大小； $[n, m] = \text{size}(A)$,
一维矩阵(向量)： $l = \text{length}(A)$
- (2) 矩阵的加减乘法；
- (3) 矩阵的除法和线性方程解；
- (4) 矩阵的乘方和超越函数；



矩阵的加减法

- 两矩阵进行加减的条件是其行数和列数相同，简称阶次相同，矩阵的加减即其对应元素的加减。

$$\text{size}(A)=\text{size}(B)$$

如果其中一个变量是标量，则会自动扩展成各元素均取此标量值的同阶次矩阵。



矩阵的乘法

$$\begin{bmatrix} c(1,1) & \cdots & \cdots & c(1,m) \\ \cdots & c(i,j) & \cdots & \\ c(n,1) & \cdots & \cdots & c(n,m) \end{bmatrix} = \begin{bmatrix} a(1,1) & \cdots & \cdots & \cdots a(1,p) \\ \cdots & \cdots & \cdots & \\ a(i,1) & a(i,2) & \cdots & a(i,p) \\ \cdots & \cdots & \cdots & \\ a(n,1) & \cdots & \cdots & \cdots a(n,p) \end{bmatrix} * \begin{bmatrix} b(1,1) & \cdots & b(1,j) & \cdots & b(1,m) \\ \cdots & \cdots & b(2,j) & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \\ b(p,1) & \cdots & b(p,j) & \cdots & b(p,m) \end{bmatrix}$$

$$\mathbf{C}(n,m) = \mathbf{A}(n,p) * \mathbf{B}(p,m)$$

$$c(i,j) = a(i,1)*b(1,j) + a(i,2)*b(2,j) + \cdots + a(i,p)*b(p,j) = \sum_p a(i,k)*b(k,j)$$



矩阵的乘法

- 两矩阵相乘 $C=A*B$ 的条件是A的列数等于B的行数，简称内阶数相同。设矩阵A 为 $n \times p$ 阶，矩阵B为 $p \times m$ 阶，则C是 $n \times m$ 阶。内阶数不同的矩阵不能相乘。
矩阵乘法不符合交换律： $A*B \neq B*A$ ，左乘不等于右乘；
- 若一个变量是标量，则按标量乘法。



矩阵求逆和矩阵除法

线性方程组 $D*X=B$ ，如果 D 非奇异，即它的逆矩阵 $\text{inv}(D)$ 存在；

则其解用MATLAB表为：

$$X=\text{inv}(D)*B=D\backslash B$$

符号‘ \backslash ’称为左除，即分母放在左边。

左除的条件： B 的行数等于 D 的阶数（ D 的行数和列数相同，简称阶数）



矩阵右除

若方程组表为为 $X * D1 = B1$ ， $D1$ 非奇异，即它的逆阵 $\text{inv}(D1)$ 存在。

则其解为

$$X = B1 * \text{inv}(D1) = B1 / D1$$

符号 ‘/’ 称为右除。

右除的条件： $B1$ 的列数等于 $D1$ 的阶数（ $D1$ 的行数和列数相同，简称阶数）



矩阵整体的幂次运算

- MATLAB的乘幂函数“^”、指数函数expm、对数函数logm、和开方函数sqrtm是对矩阵进行的，另外，还有矩阵乘法和除法也是把矩阵作为一个整体来运算。
- 除此之外，其他MATLAB函数都是对矩阵中的元素分别进行，



用MATLAB做行阶梯变换



用MATLAB进行行阶梯变换

将线性方程组表示为增广矩阵，行代表方程，在MATLAB中矩阵a的第i行全部元素表示为 $a(i,:)$ ：

(1) 将第i,j两行进行交换的语句为： $a([i,j],:)=a([j,i],:)$

这里的等号不是数学的等号，而是赋值，把等号右边的计算结果赋值给左边的变量。

(2) 将第j行乘以常数k的语句为： $a(j,:)=k*a(j,:)$

同样，等号必须理解为赋值。等式右、左的 $a(j,:)$ 不同，它们分别是赋值前、后行向量的值。

(3) 将第i行乘以常数k加到第j行的运算语句为：

$a(j,:)=a(j,:)+k*a(i,:)$ ，其中 $k=-a(j,i)/a(i,i)$



MATLAB中的行阶梯变换程序

- 最简形变换程序(Reduced Row Echelon Form—rref),调用方法:
- 对系数矩阵: $U0=rref(A)$, $[U0,ip]=rref(A)$,
- 对增广矩阵: $U0c=rref([A,b])=rref(C)$,
- 如果只要变到行阶梯形式, 则可调用本书改编的程序:
- $U1=ref1(A)$ (阶梯形), 或 $U2=ref2(A)$ (三角形),



消元法中的矩阵运算

- 消元法的基本运算是将矩阵中待消行 $a(j,:)$ 与基准行 $a(i,:)$ 乘以 $-a(j,i)/a(i,i)$ 后相加，使新行中的 $a(j,i)$ 变为零。涉及的算法只是行的数乘和相加。
- 运算实现的条件是 $a(i,i) \neq 0$ ，即主元不等于零。程序要对此进行检验。如果出现此情况，它要用行交换和列交换的方法予以排除。
- 程序第一阶段把矩阵A通过消元变为上三角矩阵U1，第二阶段通过回代变为对角矩阵U2，然后通过除以主元变为对角项均为1的行最简形U0。
- 这三个过程分别用MATLAB函数ref1,ref2,rref来完成。

消元法中的矩阵运算

$$\mathbf{A} \rightarrow \begin{bmatrix} p_{11} & * & * & * & \vdots & * \\ 0 & p_{22} & * & * & \vdots & * \\ 0 & 0 & p_{33} & * & \vdots & * \\ 0 & 0 & 0 & p_{44} & \vdots & * \end{bmatrix} \rightarrow \begin{bmatrix} p_{11} & 0 & 0 & 0 & \vdots & d_1 \\ 0 & p_{22} & 0 & 0 & \vdots & d_2 \\ 0 & 0 & p_{33} & 0 & \vdots & d_3 \\ 0 & 0 & 0 & p_{44} & \vdots & d_4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & \vdots & d_1/p_{11} \\ 0 & 1 & 0 & 0 & \vdots & d_2/p_{22} \\ 0 & 0 & 1 & 0 & \vdots & d_3/p_{33} \\ 0 & 0 & 0 & 1 & \vdots & d_4/p_{44} \end{bmatrix}$$

$$\mathbf{A} \rightarrow \mathbf{U1} = \text{ref1}(\mathbf{A}) \quad \rightarrow \mathbf{U2} = \text{ref2}(\mathbf{A}) \quad \rightarrow \mathbf{U0} = \text{rref}(\mathbf{A})$$

- 在MATLAB软件中，只有rref函数，没有ref1,ref2。
本书为了说明行列式的主元连乘定义
 $D = p_{11}p_{22}p_{33}p_{44}$ ，需要把消元过程中的主元 p_{ii} 突显出来，才编写了这两个求行阶梯矩阵的中间函数。



用MATLAB解线性方程组



线性方程组表为矩阵相乘

$$x_1 + 2x_2 + 3x_3 = 2$$

$$3x_1 - 5x_2 + 4x_3 = 0$$

$$7x_1 + 8x_2 + 9x_3 = 2$$

可以表为

$$A * X = \begin{bmatrix} 1 & 2 & 3 \\ 3 & -5 & 4 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} = b$$

故有 $X=[x_1;x_2;x_3]=A \setminus b$



用矩阵除法解线性方程组

解方程组

$$6x_1 + 3x_2 + 4x_3 = 3$$

$$-2x_1 + 5x_2 + 7x_3 = -4$$

$$8x_1 - 4x_2 - 3x_3 = -7$$

的程序为：

$$A = [6,3,4; -2,5,7; 8,-4,-3];$$

$$B = [3;-4;-7];$$

$$X = A \setminus B$$



元素群运算



元素群运算

- 数组及其赋值
- 元素群的四则运算
- 元素群的幂次运算
- 元素群的函数

元素群运算能大大简化编程，提高运算的效率，是MATLAB优于其他许多语言的一个特色。



1) 数组及其赋值

- 数组通常是指单行或单列的矩阵，也称为向量。
- 其赋值方法：
 - (1) $x=[\text{初值}:\text{增量}:\text{终值}]$
 - (2) 线性分割函数
 $x=\text{ linspace }(\text{起点},\text{终点},\text{点数})$
 - (3) 对数分割函数
 $\text{logspace}(\text{起点对数},\text{终点对数},\text{点数})$



2) 元素群的四则和幂次运算

- 元素群运算：就是把两矩阵按逐个元素进行运算。为了与矩阵作为整体的运算符号相区别，要在运算符“*、/、\、^”前加一点符号“.”。
- 例如， $X=[1, 2, 3]$ ， $Y=[4, 5, 6]$ ，则有以下表：

运 算 式	输 出 结 果
$Z=X.*Y$	$Z = 4 \quad 10 \quad 18$
$Z=X.\backslash Y$	$Z = 4.0000 \quad 2.5000 \quad 2.0000$
$Z=X.^Y$	$Z = 1 \quad 32 \quad 729$
$Z=X.^2$	$Z = 1 \quad 4 \quad 9$



3) 元素群的函数运算

- 所有的MATLAB函数都适用于作元素群运算，只有专门说明的几个除外。

就是*、/、\、^ 运算符和sqrtm、expm、logm三个函数。

- 元素群的优越性可从做一个三角函数表看出：

```
x=[0:0.1:pi/4]';
```

```
[x, sin(x), cos(x), tan(x)]
```



基本绘图方法



5 基本绘图方法

- (1) 直角坐标中的两维曲线
- (2) 多条曲线的绘制
- (3) 空间曲面
- (4) 快速绘图



直角坐标中的两维曲线

如果Y是一个数组，n是其下标，则：

- plot(Y)就以n为横坐标画出Y的折线，例如
 $Y = [-1.4, -2.2, 0.9, 0.9, 2, -0.6, 0.1, 1.7, -2.3, -2.2]$, plot(Y)

若Y是t的函数，则plot(t,Y)画出此函数曲线；

若想将三点(1,2),(3,4),(4,1),(1,2)封闭连接，

可用plot([1,3,4,1],[2,4,1,2])

最后一个变元前一个符号为线型（点型） 后一个字母表示颜色



线型、点型和颜色

表2—11 线型、点型和颜色

标志符	颜色	标志符	线型和点型
y	黄	.	点
m	品红	o	圆圈
c	青	x	x号
r	红	+	+ 号
g	绿	-	实线
b	兰	*	星号
w	白	:	虚线
k	黑	-.	点划线
		--	长划线



控制绘图的其他命令

- 1, `hold on(off)` 可把多条曲线放在一张图上;
- 2, `title` 给图加上标题,
- 3, `xlabel, ylabel` 标出纵、横坐标的名称,
- 4, `subplot(n, m, p)` 将图分割为 $n \times m$ 个子图,
- 5, `grid on(off)` 给图上加上 (去掉) 网格,
- 6, `axis` 规定图上的坐标范围和纵横比,
-



其他二维绘图

- 线性直角坐标系中其他形式绘图的命令有stem（绘脉冲图），stairs（绘阶梯图）
bar（绘直方图），fill（绘填充图）
- 对数直角坐标系中的绘图命令有loglog, semilogx, semilogy等；
- 非直角坐标系中的绘图命令有polar(theta, rho)



4. 坐标尺寸的设定—axis命令

- `V= axis`, 将返回图形边界的 4 元行向量, 即 $V=[xmin, xmax, ymin, ymax]$;
- `axis(v)`; 将把图形边界设置为用户规定的范围; $v=[x1, x2, y1, y2]$ 为四元行向量。
- `axis('equal')`使屏幕上x与y的比例尺相同, 在这种方式下, 斜率 1 的直线的倾角为 45 度, 圆不会变成椭圆。
- `axis('square')`使有效的图形窗成为正方形



三维绘图和快速绘图



三维曲面

- 三维曲面的自变量是两维矩阵 X 和 Y ，代表 xy 平面上点阵的 x, y 坐标；
- 第三维因变量 z 是 x, y 的函数， $z=f(X, Y)$ 得到的数组与 X, Y 同阶；
- 空间曲面的绘制 `mesh(X, Y, z)` 或 `mesh(z)`
 - X, Y 为两维自变量矩阵，表示平面上的点阵；
 - Z 是 X, Y 的函数，所以是两维的因变量矩阵；
- `mesh(X, Y, Z, C)`
 - 变元 C 是颜色，可以表示四维空间。



网格点的坐标矩阵的生成

- 将这25点的x和y坐标X和Y可用 $[X,Y]=\text{meshgrid}(x,y)$ 实现。假如要绘制平面 $z-2x-3y=2$ ，则程序可编写如下：

```
% 生成两个方向的自变量向量
```

```
x=[-1: 0.5: 1]; y=[-1: 0.5: 1];
```

```
% 生成二维自变量网格坐标矩阵
```

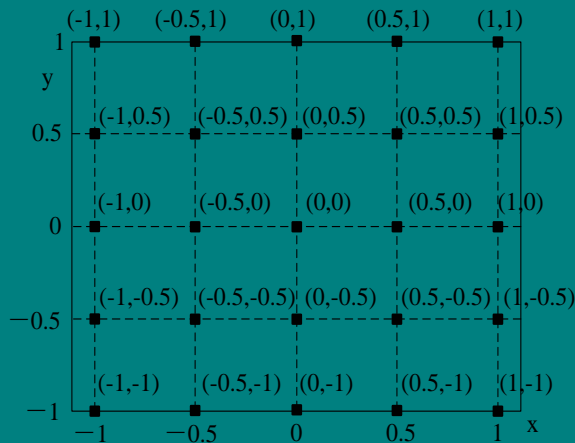
```
[X,Y]=meshgrid(x,y);
```

```
Z=2*X+3*Y+2; % 按方程求出因变量Z矩阵数据
```

3) 空间曲面的绘制

■ 前两行命令建立了共有 $5 \times 5 = 25$ 个网格点的坐标矩阵X和Y，形成了 5×5 网格的矩阵；第三行程序计算出25个网格点上的z坐标，构成与X和Y同阶的Z矩阵，最后用mesh函数绘出三维图形。

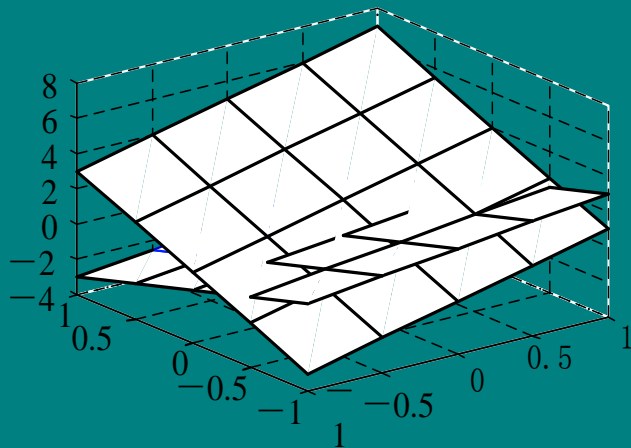
■ `mesh(X,Y,Z),pause`
% 画三维曲面



空间平面图形的生成

- 执行了第三、四行程序后将得出图A-5所示的图形。同样可以用hold on命令将它保持，并画入另一个平面方程的图形。例如键入：
■ hold on, Z1=X-2*Y; mesh(X,Y,Z1)

此时，图A-5出现两个平面，因为我们取的自变量网格太稀，所以它们的交线只能隐约地看出。





6 快速绘图

- 1) 直线的快速绘制
- MATLAB中的ezplot可以绘制很多函数的曲线，在它的第一输入变元中可以直接输入用MATLAB语句写出函数的形式，不过要用单引号括起来。第二输入变元为自变量的取值范围，在默认情况下其取值范围为 $[-2\pi, 2\pi]$ 。
- 键入ezplot('sin(x)*cos(x)'), 就绘制出 $y=\sin(x)*\cos(x)$ 在 $-2\pi \leq x \leq 2\pi$ 范围内的曲线。如果引号中的函数有两个自变量，那就代表隐函数，其典型格式为
- ezplot('3*x1+2*x2+3',[-3,3])
- 含义为，在 $-3 < x < 3$ 的范围内画出 $3x_1+2x_2+3=0$ 。

2) 平面的快速绘制

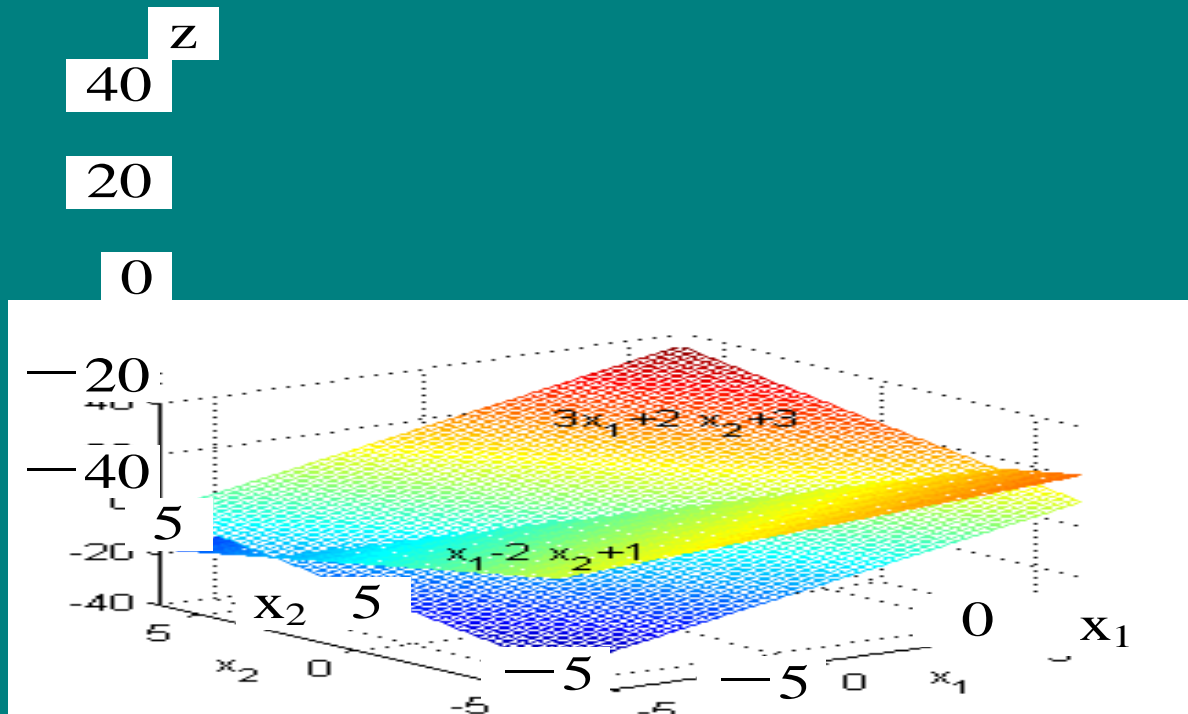
- MATLAB中的ezmesh(或ezsurf)函数可以绘制函数的曲面。与ezplot函数相仿，它的第一输入变元为用MATLAB语句写出函数的形式，注意要用单引号括起来。第二输入变元为自变量的取值范围，在x, y两个方向的默认取值范围都是 $[-2\pi, 2\pi]$ 。引号中的函数是显函数 $z=f(x,y)$ 中的 $f(x,y)$ 。它应该有两个自变量，例如键入：

```
ezmesh('3*x1+2*x2+3')
```

- 系统就绘制出 $z_1=3x_1+2x_2+3$ 在 $-2\pi \leq x_1, x_2 \leq 2\pi$ 范围内的平面图形。
- 要在同一张图上画出第二个平面，在画完第一个平面后，必须键入hold on，再键入画第二个平面的命令。例如

```
hold on, ezmesh('x1-2*x2+1')
```
- 这时所得到的图形如后页所示。

快速绘制两个平面



符号变量与公式推导

- MATLAB数值计算的许多功能可以扩展为公式的推演。安装了符号数学(Symbolic)工具箱后，可以把符号变量像数字那样直接运算。首先要使系统知道，用户给出的标识符(例如x)是数字变量还是符号变量。为此对自变量要用syms命令作出定义，比如求a,b,c,d四个元素组成的方阵的行列式D和逆阵V，其程序如下：

```
syms a b c d, A=[a,b;c,d],D=det(A),V=inv(A)
```

- 得到：

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{bmatrix} c & d \end{bmatrix}$$

$$D = a*d-b*c$$

$$V = \begin{bmatrix} d/(a*d-b*c), & -b/(a*d-b*c) \end{bmatrix}$$

$$\begin{bmatrix} -c/(a*d-b*c), & a/(a*d-b*c) \end{bmatrix}$$



下一讲 程序文件



8. 程序文件(M文件)

- 在入门阶段，通常在行命令模式下工作。键入一行命令后，让系统立即执行该命令。用这种方法时，操作虽然简单，但程序可读性很差且难以存储。对于复杂的问题，应编成可存储的程序文本，让MATLAB执行。这种工作模式称为程序文件模式。
- 由MATLAB语句构成的程序文件称为M文件M文件可分为两种：一种是主程序，也称为脚本文件(script file)，是由用户为解决特定的。问题而编制的；解决特定问题的程序都是主程序文件，如本书程序集中的plaxxx，都是求解例题xxx的程序；



1)

主程序文件

- 主程序文件就是把要求计算机执行的各条语句依次排列组成的文件。给它一个文件名并存储在MATLAB的搜索路径上。以后，只要键入这个文件名，系统就执行这些语句。本书上为各个例题所写的命名为“pla***”的程序都是主程序文件。这里提出三点注意事项：
 - (1) 最好用clear、close all等语句开始，清除掉工作空间中原有的变量和图形，以避免其他已执行的程序残留数据对本程序运行的影响。
 - (2) 程序中必须都用半角英文字母和符号，只有单引号括住的部分和%号后的内容可用汉字。用MATLAB的编辑器来编写比较好，因为它对出现的非法字符会显示出鲜明的红色，引起用户的注意。



函数文件

- 另一种是子程序，也称为函数文件(function file)，它必须由其他M文件来调用。函数文件往往具有一定的通用性，并且可以进行递归调用。MATLAB已有了几千个函数文件，并在不断扩充积累。通常初学者还没有这样的水平来自编，都是调用系统中高手们编好的程序。其特点是一般都有输入输出变元的调用关系，如`[U,ip]=rref(A)`，`plot(x,y)`，`[L,U]=lu(A)`，...等。



2) 函数文件

函数文件是用来定义子程序的。它与主程序文件的主要区别有3点：

- (1) 由function起头，后跟的函数名必须与文件名同；
- (2) 紧跟function的语句格式是[输出变元]=函数名(输入变元)，变元用来进行变量传递；
- (3) 程序中的变量均为局部变量，不保存在工作空间中，除非专门声明。
- 我们调用的[U,ip]=rref (A), [L,U]=lu(A), [Q,R]=qr(A), r=rank(A,tol), D=det(A),...等命令都是子程序，它们都有极大的通用性。



“MATLAB初步” 结束，谢谢！