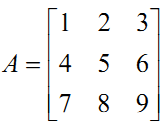
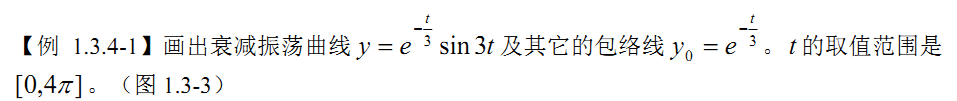
简单矩阵的输入方法：



A = [1,2,3;4,5,6;7,8,9]

图形表示



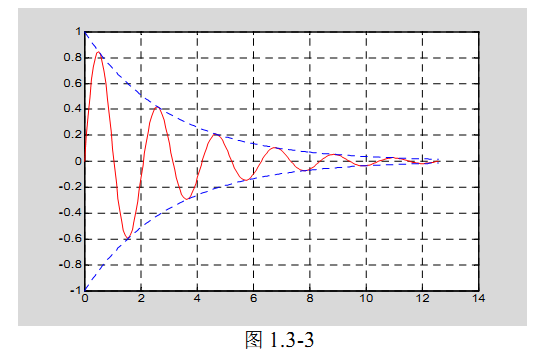
t=0:pi/50:4\*pi; 从0开始，4pi结束，步长为pi/50

y0=exp(-t/3); exp表示e的指数

y=exp(-t/3).\*sin(3\*t); .表示是向量相乘。

plot(t,y,'-r',t,y0,':b',t,-y0,':b') t表示横轴，y表示纵轴，''属性，可以有多个函数。

grid 增加网格线



二维绘图：

二维图形是将平面坐标上的数据点连接起来的平面图形。可以采用不同的坐标系，如直角坐标、对数坐标、极坐标等。二维图形的绘制是其他绘图操作的基础。

一．绘制二维曲线的基本函数

在Matlab中，最基本而且应用最为广泛的绘图函数为plot，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函数用于绘制二维平面上的线性坐标曲线图，要提供一组x坐标和对应的y坐标，可以绘制分别以x和y为横、纵坐标的二维曲线。plot函数的应用格式

plot(x,y)     其中x,y为长度相同的向量，存储x坐标和y坐标。

例51 在[0 , 2pi]区间，绘制曲线

程序如下：在命令窗口中输入以下命令

>> x=0:pi/100:2\*pi;

>> y=2\*exp(-0.5\*x).\*sin(2\*pi\*x);

>> plot(x,y)

程序执行后，打开一个图形窗口，在其中绘制出如下曲线

注意：指数函数和正弦函数之间要用点乘运算，因为二者是向量。

例52 绘制曲线

这是以参数形式给出的曲线方程，只要给定参数向量，再分别求出x,y向量即可输出曲线：

>> t=-pi:pi/100:pi;

>> x=t.\*cos(3\*t);

>> y=t.\*sin(t).\*sin(t);

>> plot(x,y)

程序执行后，打开一个图形窗口，在其中绘制出如下曲线

2． 含多个输入参数的plot函数

plot函数可以包含若干组向量对，每一组可以绘制出一条曲线。含多个输入参数的plot函数调用格式为：plot(x1，y1，x2，y2，…，xn，yn)

如下列命令可以在同一坐标中画出3条曲线。

>> x=linspace(0,2\*pi,100);

>> plot(x,sin(x),x,2\*sin(x),x,3\*sin(x))

当输入参数有矩阵形式时，配对的x,y按对应的列元素为横坐标和纵坐标绘制曲线，曲线条数等于矩阵的列数。

>> x=linspace(0,2\*pi,100);

>> y1=sin(x);

>> y2=2\*sin(x);

>> y3=3\*sin(x);

>> x=[x;x;x]';

>> y=[y1;y2;y3]';

>> plot(x,y,x,cos(x))

x,y都是含有三列的矩阵，它们组成输入参数对，绘制三条曲线；x和cos(x)又组成一对，绘制一条余弦曲线。

利用plot函数可以直接将矩阵的数据绘制在图形窗体中，此时plot函数将矩阵的每一列数据作为一条曲线绘制在窗体中。如

>> A=pascal(5)

A =

     1     1     1     1     1

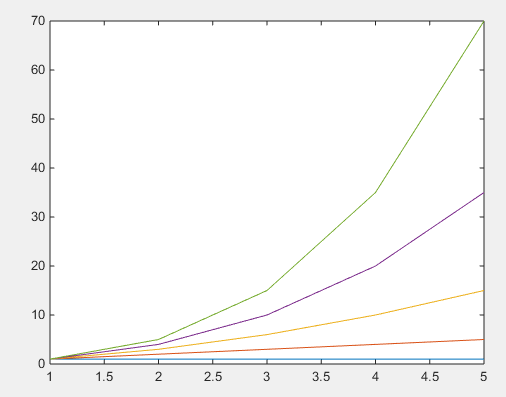
     1     2     3     4     5

     1      3     6    10    15

     1     4    10    20    35

     1     5    15    35    70

>> plot(A)



3． 含选项的plot函数

Matlab提供了一些绘图选项，用于确定所绘曲线的线型、颜色和数据点标记符号。这些选项如表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 线型 | 颜色 | 标记符号 | |
| - 实线 | b蓝色 | .   点 | s 方块 |
| : 虚线 | g绿色 | o 圆圈 | d 菱形 |
| -. 点划线 | r红色 | × 叉号 | ∨朝下三角符号 |
| -- 双划线 | c青色 | + 加号 | ∧朝上三角符号 |
|  | m品红 | \* 星号 | <朝左三角符号 |
|  | y黄色 |  | >朝右三角符号 |
|  | k黑色 |  | p 五角星 |
|  | w白色 |  | h 六角星 |

 组合使用

例 用不同的线型和颜色在同一坐标内绘制曲线 及其包络线。

>> x=(0:pi/100:2\*pi)';

>> y1=2\*exp(-0.5\*x)\*[1,-1];

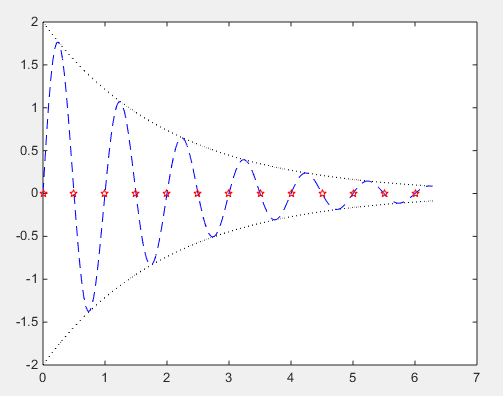
>> y2=2\*exp(-0.5\*x).\*sin(2\*pi\*x);

>> x1=(0:12)/2;

>> y3=2\*exp(-0.5\*x1).\*sin(2\*pi\*x1);

>> plot(x,y1,'k:',x,y2,'b--',x1,y3,'rp');

在该plot函数中包含了3组绘图参数，第一组用黑色虚线画出两条包络线，第二组用蓝色双划线画出曲线y，第三组用红色五角星离散标出数据点。



4． 双纵坐标函数plotyy

在Matlab中，如果需要绘制出具有不同纵坐标标度的两个图形，可以使用plotyy函数，它能把具有不同量纲，不同数量级的两个函数绘制在同一个坐标中，有利于图形数据的对比分析。使用格式为：plotyy(x1,y1,x2,y2)

x1,y1对应一条曲线，x2,y2对应另一条曲线。横坐标的标度相同，纵坐标有两个，左边的对应x1,y1数据对，右边的对应x2,y2。

二．绘制图形的辅助操作

绘制完图形以后，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。

1． 图形标注 可以在图中进行直接修改

在绘制图形时，可以对图形加上一些说明，如图形的名称、坐标轴说明以及图形某一部分的含义等，这些操作称为添加图形标注。有关图形标注函数的调用格式为：

title（’图形名称’） （都放在单引号内）

xlabel（’x轴说明’）

ylabel（’y轴说明’）

text（x，y，’图形说明’）

legend（’图例1’，’图例2’，…）

其中，title、xlabel和ylabel函数分别用于说明图形和坐标轴的名称。text函数是在坐标点（x，y）处添加图形说明。legend函数用于绘制曲线所用线型、颜色或数据点标记图例，图例放置在空白处，用户还可以通过鼠标移动图例，将其放到所希望的位置。除legend函数外，其他函数同样适用于三维图形，在三维中z坐标轴说明用zlabel函数。

上述函数中的说明文字，除了使用标准的ASCII字符外，还可以使用LaTex（一种流行的数学排版软件）格式的控制字符，这样就可以在图形上添加希腊字符，数学符号和公式等内容。在Matlab支持的LaTex字符串中，用/bf , /it , /rm控制字符分别定义黑体、斜体和正体字符，受LaTex字符串控制部分要加大括号{}括起来。例如，text(0.3，0.5，’the usful {/bf MATLAB}’)，将使MATLAB一词黑体显示。一些常用的LaTex字符见表，各个字符可以单独使用也可以和其他字符及命令配合使用。如text(0.3 ,0.5 ,’sin({/omega}t+{/beta})’)

将得到标注效果 。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 标识符 | 符号 | 标识符 | 符号 | 标识符 | 符号 |
| /alpha |  | /epsilon |  | /infty |  |
| /beta |  | /eta |  | /int |  |
| /gamma |  | /Gamma |  | /partial |  |
| /delta |  | /Delta |  | /leftarrow |  |
| /theta |  | /Theta |  | /rightarrow |  |
| /lambda |  | /Lambda |  | /downarrow |  |
| /xi |  | /Xi |  | /uparrow |  |
| /pi |  | /Pi |  | /div |  |
| /omega |  | /Omega |  | /times |  |
| /sigma |  | /Sigma |  | /pm |  |
| /phi |  | /Phi |  | /leq |  |
| /psi |  | /Psi |  | /geq |  |
| /rho |  | /tau |  | /neq |  |
| /mu |  | /zeta |  | /forall |  |
| /nu |  | /chi |  | /exists |  |

**2． 坐标控制**

在绘制图形时，Matlab可以自动根据要绘制曲线数据的范围选择合适的坐标刻度，使得曲线能够尽可能清晰的显示出来。所以，一般情况下用户不必选择坐标轴的刻度范围。但是，如果用户对坐标不满意，可以利用axis函数对其重新设定。其调用格式为

axis（[xmin xmax ymin ymax zmin zmax]）

如果只给出前四个参数，则按照给出的x、y轴的最小值和最大值选择坐标系范围，绘制出合适的二维曲线。如果给出了全部参数，则绘制出三维图形。

axis函数的功能丰富，其常用的用法有：

axis equal ：纵横坐标轴采用等长刻度

axis square：产生正方形坐标系（默认为矩形）

axis auto：使用默认设置

axis off：取消坐标轴

axis on ：显示坐标轴

还有：给坐标加网格线可以用grid命令来控制，grid on/off命令控制画还是不画网格线，不带参数的grid命令在两种之间进行切换。

给坐标加边框用box命令控制。和grid一样用法

3． 图形保持

一般情况下，每执行一次绘图命令，就刷新一次当前图形窗口，图形窗口原有图形将不复存在，如果希望在已经存在的图形上再继续添加新的图形，可以使用图形保持命令hold。hold on/off 命令是保持原有图形还是刷新原有图形，不带参数的hold命令在两者之间进行切换。

4． 图形窗口分割

在实际应用中，经常需要在一个图形窗口中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。同一图形窗口下的不同图形称为子图。Matlab提供了subplot函数用来将当前窗口分割成若干个绘图区，每个区域代表一个独立的子图，也是一个独立的坐标系，可以通过subplot函数激活某一区，该区为活动区，所发出的绘图命令都是作用于该活动区域。调用格式：

subplot（m，n，p）

该函数把当前窗口分成m×n个绘图区，m行，每行n个绘图区，区号按行优先编号。其中第p个区为当前活动区。每一个绘图区允许以不同的坐标系单独绘制图形。

三．绘制二维图形的其他函数

1． 其他形式的线性直角坐标图

在线性直角坐标中，其他形式的图形有条形图、阶梯图、杆图和填充图等，所采用的函数分别为：

bar（x，y，选项）      选项在单引号中

stairs（x，y，选项）

stem（x，y，选项）

fill（x1，y1，选项1，x2，y2，选项2，…）

前三个函数和plot的用法相似，只是没有多输入变量形式。fill函数按向量元素下标渐增次序依次用直线段连接x，y对应元素定义的数据点。

例5-8：分别以条形图、填充图、阶梯图和杆图形式绘制曲线

x=0:0.35:7;

y=2\*exp(-0.5\*x);

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

title('bar(x,y,''g'')');axis([0, 7, 0 ,2]);

subplot(2,2,2);fill(x,y,'r');

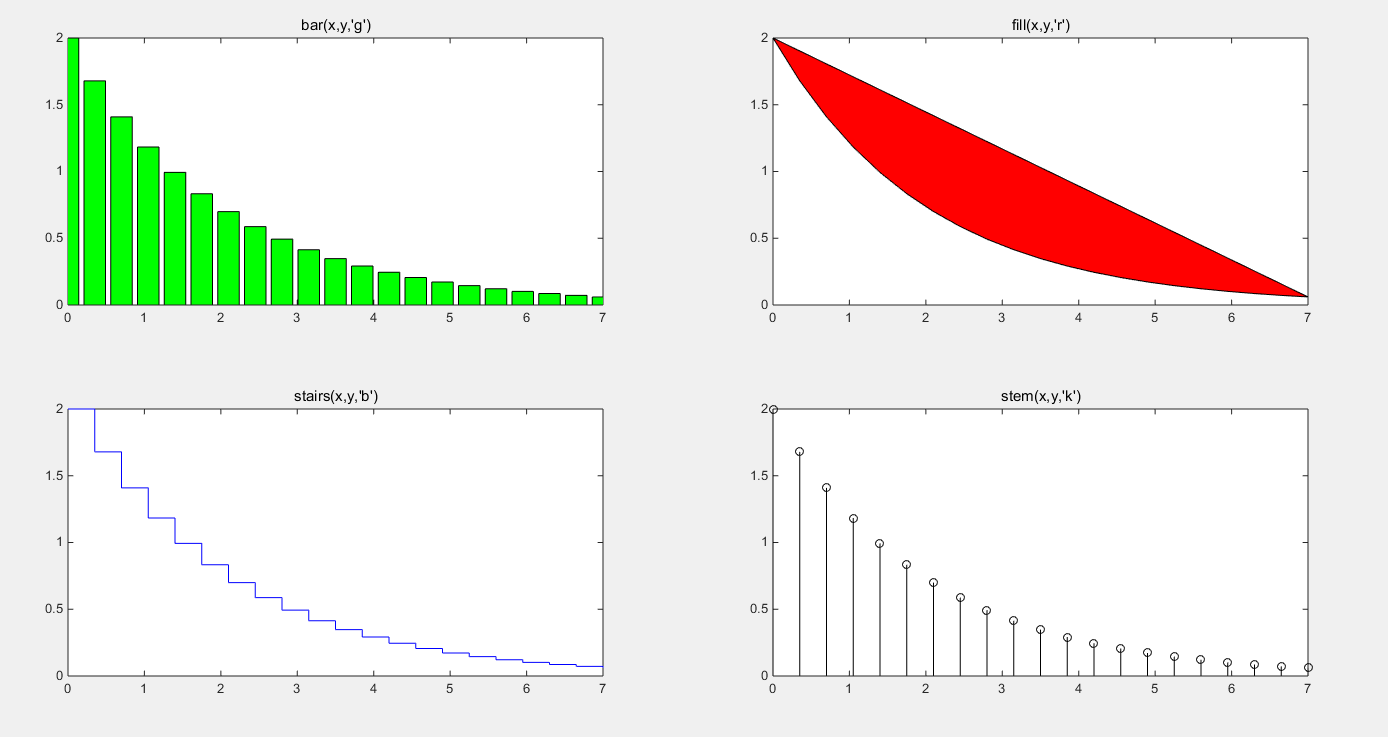
title('fill(x,y,''r'')');axis([0, 7, 0 ,2]);

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

title('stairs(x,y,''b'')');axis([0, 7, 0 ,2]);

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

title('stem(x,y,''k'')');axis([0, 7, 0 ,2]);



2． 极坐标图

polar函数用来绘制极坐标图，调用格式为：

polar（theta，rho，选项）

其中，theta为极坐标极角，rho为极径，选项的内容和plot函数相似。

例5-9：绘制 的极坐标图

theta=0:0.01:2\*pi;

rho=sin(3\*theta).\*cos(5\*theta);

polar(theta,rho,'r');

**3． 对数坐标图**

在实际应用中，经常用到对数坐标，Matlab提供了绘制对数和半对数坐标曲线的函数，其调用格式为：

semilogx（x1，y1，选项1，x2，y2，选项2，…）

semilogy（x1，y1，选项1，x2，y2，选项2，…）

loglog（x1，y1，选项1，x2，y2，选项2，…）

这些函数中选项的定义和plot函数完全一样，所不同的是坐标轴的选取。semilogx函数使用半对数坐标，x轴为常用对数刻度，而y轴仍保持线性刻度。semilogy恰好和semilogx相反。loglog函数使用全对数坐标，x、y轴均采用对数刻度。

例：略

4． 对函数自适应采样的绘图函数

5． 其他形式的二维图形

三维的使用plot3来做

clear清除所有的变量

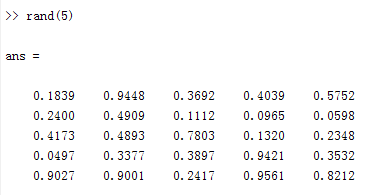
clc只清屏

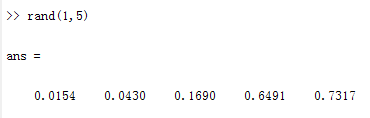
数据文件的存取

一维数组

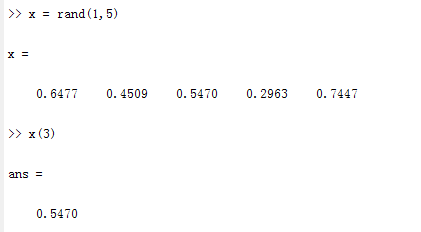
rand随机数，在0，1之间随机选择数字

rand（m，n），其中m表示行数，n表示列数，构成m×n，如果只有一个m，则表示m×m

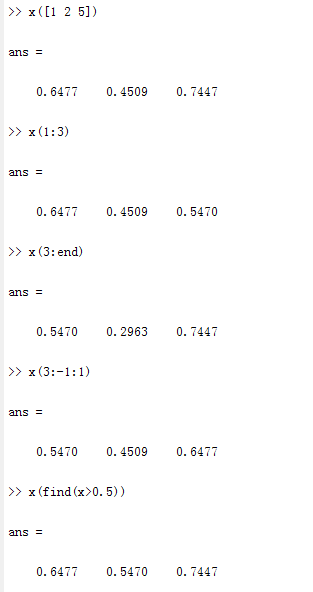


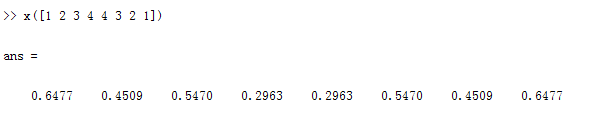


其内部的数字标识为1开始，如



3表示第三个数字





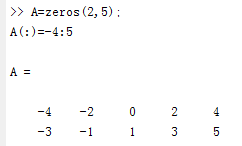
子数组的赋值

x(3) = 0

x([1 4])=[1 1]

可以直接按号码进行赋值

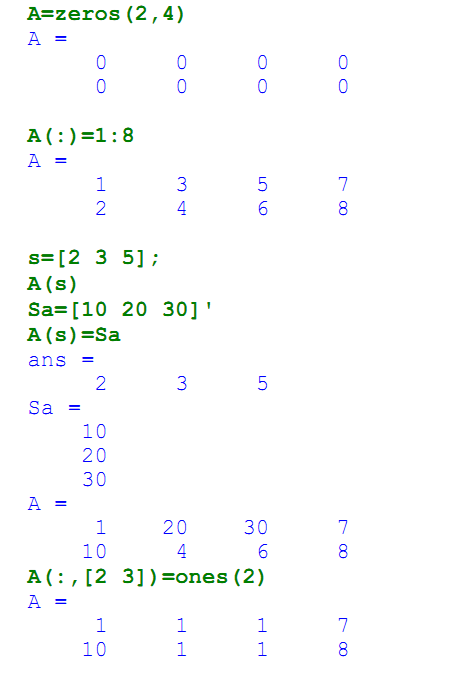
二维数组



首先使用zeros定义一个全0的矩阵，且2行5列，跟前面的rand一样

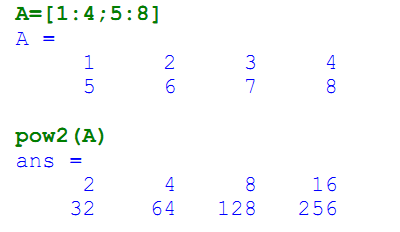
然后给A赋值，-4开始，5结束，步长为1，

-4，-3，-2，-1，0，1，2，3，4，5，但是要变成2行5列。就上面的，注意顺序



ones表示全1的矩阵，其用法和rand一致

pow2运算：



2^x次方运算。

A’转置

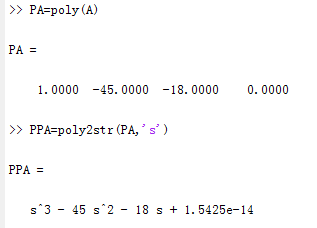
多项式

【例 2.8.1.2-1】求3阶方阵A的特征多项式。

A=[11 12 13;14 15 16;17 18 19];

PA=poly(A) 系数

PPA=poly2str(PA,'s') 表达式



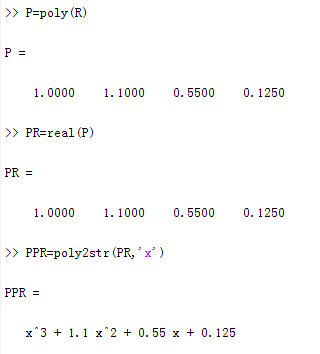
【例 2.8.1.2-2】由给定根向量求多项式系数向量。

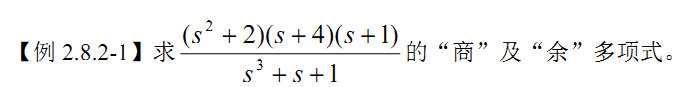
R=[-0.5,-0.3+0.4\*i,-0.3-0.4\*i];

P=poly(R)

PR=real(P)

PPR=poly2str(PR,'x')





p1=conv([1,0,2],conv([1,4],[1,1]));

p2=[1 0 1 1];

[q,r]=deconv(p1,p2); q是商，r是余，p1是分子，p2是分母

cq='商多项式为 '; cr='余多项式为 ';

disp([cq,poly2str(q,'s')]),disp([cr,poly2str(r,'s')])

商多项式为 s + 5

余多项式为 5 s^2 + 4 s + 3

polyval

polyvalm

【例 2.8.2-3】部分分式展开。

a=[1,3,4,2,7,2];

b=[3,2,5,4,6];

[r,s,k]=residue(b,a)

标准数组

ones 全1

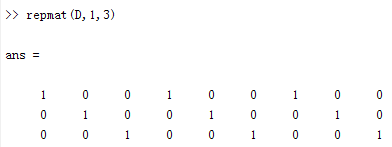
zeros 全0

rand 随机

eye 单位矩阵

diag 对角线矩阵，或者获取对角线的元素

repmat ，复制矩阵



LU分解：

在线性代数中， LU分解(LU Decomposition)是矩阵分解的一种，可以将一个矩阵分解为一个单位下三角矩阵和一个上三角矩阵的乘积（有时是它们和一个置换矩阵的乘积）。LU分解主要应用在数值分析中，用来解线性方程、求反矩阵或计算行列式。

求逆矩阵：

|  |
| --- |
| 例：给定一4阶矩阵，通过LU分解求逆矩阵。  解：算法过程为：，  第一步：求LU矩阵  设，通过（4）~（7）式可逐步进行矩阵L和U中元素的计算，如下所示：      经迭代计算，最后得到L和U矩阵为：  1ZANPKZPZ17P[6)YDA58{{L  第二步：求L和U矩阵的逆u，l    （1）求U矩阵的逆    由式（9）可得矩阵U的逆的各元素计算如下：    （2）求L矩阵的逆  由（8）式可得L矩阵的逆的各元素计算如下    所以得到的逆矩阵为：  ]HMQY$MQ$GAZ9C(B1[[@[N1  （3）求A的逆矩阵  由式（10）可计算得到矩阵A的逆，如下：  由程序计算出的结果如下：  2)DU@WX%`P@ND6`X5U]]`$M  MATLAB代码：  function [x] = Dool(A,b)  n=length(A);  A(2:n,1)=A(2:n,1)/A(1,1);  for t=2:n-1  A(t,t:n)=A(t,t:n)-A(t,1:t-1)\*A(1:t-1,t:n);  A(t+1:n,t)=(A(t+1:n,t)-A(t+1:n,1:t-1)\*A(1:t-1,t))/A(t,t);  end  A(n,n)=A(n,n)-A(n,1:n-1)\*A(1:n-1,n);  L=tril(A,-1)+eye(n);U=triu(A); %求出L和U  l=inv(L); u=inv(U);  a=u\*l;%a为A的逆 |

求解方程

%LU分解法求解Ax=b，假定A矩阵可进行LU分解以及对角线元素均不为0

function [x] = Dool(A,b)

n=length(A);

A(2:n,1)=A(2:n,1)/A(1,1);%将第一列的元素，除第一个之外，全部除以第一个数，得到的结果覆盖原来对应位置的元素。

for t=2:n-1 %进行LU分解

A(t,t:n)=A(t,t:n)-A(t,1:t-1)\*A(1:t-1,t:n);

A(t+1:n,t)=(A(t+1:n,t)-A(t+1:n,1:t-1)\*A(1:t-1,t))/A(t,t);

end

A(n,n)=A(n,n)-A(n,1:n-1)\*A(1:n-1,n);

L=tril(A,-1)+eye(n);U=triu(A); %矩阵A分解出L和U

for t=2:n %解Lx=b

b(t)=b(t)-L(t,1:t-1)\*b(1:t-1);

end

b(n)=b(n)/U(n,n); %解Ux=b

for t=1:n-1;

k=n-t;b(k)=(b(k)-U(k,k+1:n)\*b(k+1:n))/U(k,k);

end

x=b; %方程Ax=b的解即为x

恰定方程组的解

线性方程组如果方程数等于未知数个数，叫做恰定方程组，如果方程多于未知数，叫做超定方程组，反之称为欠定。换个角度说，系数矩阵如果是方阵，就是恰定方程组。

恰定方程组由n个未知数的n个方程构成，方程有唯一的一组解，其一般形式可用矩阵，向量写成如下形式： Ax=b 其中A是方阵，b是一个列向量；

在线性代数教科书中，最常用的方程组解法有：

（1）利用cramer公式来求解法；

（2）利用矩阵求逆解法，即x=(A^-1)\*b；

（3）利用gaussian消去法；

（4）利用lu法求解。

MATLAB中，出于对算法稳定性的考虑，行列式及逆的计算大都在lu分解的基础上进行。 在MATLAB中，求解这类方程组的命令十分简单，直接采用表达式：x=A\b。 在MATLAB的指令解释器在确认变量A非奇异后，就对它进行lu分解，并最终给出解x；若矩阵A的条件数很大，MATLAB会提醒用户注意所得解的可靠性。 如果矩阵A是奇异的，则Ax=b的解不存在，或者存在但不唯一；如果矩阵A接近奇异时，MATLAB将给出警告信息；如果发现A是奇异的，则计算结果为inf，并且给出警告信息；如果矩阵A是病态矩阵，也会给出警告信息。 注意：在求解方程时，尽量不要用inv(A)\*b命令，而应采用A\b的解法。因为后者的计算速度比前者快、精度高，尤其当矩阵A的维数比较大时。另外，除法命令的适用行较强，对于非方阵A，也能给出最小二乘解。