2.MATLAB逻辑规则、结构基础及函数

2.1.逻辑基础

2.3.自定义函数

2.2.结构基础

2.4.特殊函数

2.5.常用函数

2.1.1 逻辑运算符

2.1.2 all、any、find函数

2.1.3 本章小节

2.1.4 课后小练

2.1.1 逻辑运算符

运算方法	函数名	运算符	运算规则（针对逻辑值）	示例
逻辑与	**and**	**&**	都为 1 时返回 1，只要有一个是 0 返回 0	and(1, 1) % 返回 1 and(1, 0) % 返回 0 and(0, 0) % 返回 0 1 & 1 % 返回 1　1 & 0 % 返回 0 0 & 0 % 返回 0
逻辑或	**or**	|	只要有一个为 1 返回 1，都是 0 时返回 0	or(1, 1) % 返回 1 or(1, 0) % 返回 1 or(0, 0) % 返回 0 1 | 1　% 返回 1　1 | 0 % 返回 1 0 | 0　% 返回 0
逻辑非	**not**	~	原来为 1 时返回 0，为 0 时返回 1	not(1)　% 返回 0 not(0)　% 返回 1 ~1　% 返回 0 ~0　% 返回 1
逻辑异或	**xor**	**无**	不相同时取 1，相同时取 0	xor(1,0)　xor(0,1) % 返回 1 xor(1,1)　xor(0,0) % 返回 0

2.1.1 逻辑运算符

MATLAB 推荐大家直接使用运算符进行计算，因此&、|和~这三个符号的功能大家要牢记。以“&”为例：

（1）针对矩阵运算时：“逻辑与&”是对 A 和 B 进行计算的，计算时会比较 A 和 B 对应位置的元素。需要符合算术运算中介绍的五种兼容模式即可。

（2）“逻辑与&”不仅可以作用在逻辑值 0 和 1 上，还可以用于普通的数值上，这时候，**MATLAB 会将非零数值视为逻辑 1，将数值零视为逻辑 0 进行运算。**

（3）“逻辑与&”也可以进行连续运算，例如 1 & 2 & 3。

（4）为避免运算优先级的问题，直接使用小括号来指定计算顺序。

2.1.1 逻辑运算符

例题：随机生成20 名同学的单科成绩（假设成绩为满分 100 分的整数制），现需要从中找到成绩等级为良（成绩在区间[60,80)内）的同学。如果要找出[0 , 60) ∪ [80 , 100]分的同学呢？

2.1.1 逻辑运算符

MATLAB 中另外两个使用频率很高的逻辑运算符：&&和||

（1）&&和||只能对标量（只有一个元素）进行逻辑运算，不能对有多个元素的向量或者 矩阵进行运算，而&和|可以。
（2）&&和||进行逻辑运算时具有短路功能，可以提高运行效率：

- 计算 A && B 时，如果 A 为逻辑 0，则 B 不会被判断，因为最后的结果一定是逻辑 0；
- 计算 A || B 时，如果 A 为逻辑 1，则 B 不会被判断，因为最后的结果一定是逻辑 1。

2.1.2 all、any、find函数

函数名	作用
all	判断数组元素是否全为非零值（可指定沿什么维度判断），全为非零值时返回逻辑 1，否则返回逻辑 0
any	判断数组元素中是否存在至少一个非零值（可指定沿什么维度判断），是的话返回逻辑 1，否则返回逻辑 0
find	查找数组中的非零元素，并返回其索引

2.1.2 all、any、find函数

all 函数和 any 函数的用法类似，**以 all 函数**为例，它的用法如下：

如果 A 是一个矩阵，那么 all(A, dim) 沿着 dim 维来判断元素是否全为非零值。

- dim等于 1 时沿着行方向来判断每一列是否全为非零值，并将结果返回为一个全为逻辑值的行向量，可以直接简写成 all(A)。
- dim 等于 2 时表示沿着列方向判断每一行是否全为非零值，并将结果返回为一个全为逻辑值的列向量。

2.1.2 all、any、find函数

例题：请随机生成一个 100 行 3 列的矩阵，用来记录学生的考试成绩：矩阵每一行代表一名同学，每一列代表一门科目的成绩，矩阵中的每个元素都是区间[50,100]内的随机整数。

（1）请指出哪些同学挂科了，至少有一门科目没过 60 分就算挂科。要求返回一个包含100 个元素的逻辑向量，元素为逻辑值 1 的位置对应的同学挂科了。

（2）这三门科目中是否存在科目没有人挂科（所有同学的这一门科目的成绩都高于 60 分）。要求返回一个包含 3 个元素的逻辑向量，元素为逻辑 1 的位置对应的科目表示没有人挂科。

2.1.2 all、any、find函数

find函数

帮助文档

说明

k = find(X) 返回一个包含数组 X 中每个非零元素的线性索引的向量。

- 如果 X 为向量，则 find 返回方向与 X 相同的向量。
- 如果 X 为多维数组，则 find 返回由结果的线性索引组成的列向量。

k = find(X,n) 返回与 X 中的非零元素对应的前 n 个索引。

k = find(X,n,direction) (其中 direction 为 'last') 查找与 X 中的非零元素对应的最后 n 个索引。direction 的默认值为 'first'，即查找与非零元素对应的前 n 个索引。

[row,col] = find(___) 使用前面语法中的任何输入参数返回数组 X 中每个非零元素的行和列下标。

[row,col,v] = find(___) 还返回包含 X 的非零元素的向量 v。

2.1.3 本章小节

- 逻辑运算符，逻辑 与、或、非、异或以及具有短路功能的&&和||
- 和逻辑相关的常用函数，all、any和find函数。

2.1.4 课后小练

请随机生成一个 100 行 3 列的矩阵，用来记录学生的考试成绩：矩阵每一行代表一名同学，每一列代表一门科目的成绩，矩阵中的每个元素都是区间[50,100]内的随机整数。

（1）找出恰好挂了两门科目的同学的编号。

（2）找到总分超过 260 分的同学的编号。

2.2.1 条件结构

2.2.2 循环结构

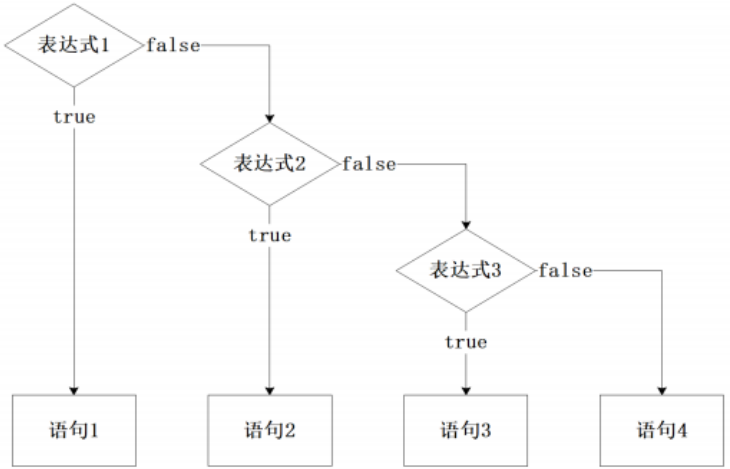
2.1.3 本章小节

2.1.4 课后小练

2.2.1 条件结构

- if-elseif-else-end 语句：
  - elseif 关键字中间不能加空格，这和 C、Java 等语言不同。
  - 在使用 if 语句时，if 和 end 这两个关键字是无论如何都不能省略的。

```
if 表达式 1
    语句 1
elseif 表达式 2
    语句 2
elseif 表达式 3
    语句 3
else
    语句 4
end
```



2.2.1 条件结构

- if-elseif-else-end 语句：

例题：（**分段函数**）给定一个同学的成绩（假设为整数），输出这个同学的等级。等级规则如下：90 至 100 分为 1 级、80 至 89 分为 2 级、60 至 79 分为 3 级、低于 60 分为 4 级；如果成绩小于 0 分或者大于 100 分，则代表成绩输入有误，此时等级为 0。

$$y=\begin{cases}1, & 90\le x\le 100\\ 2, & 80\le x<90\\ 3, & 60\le x<80\\ 4, & 0\le x<60\\ 0, & x\text{取其他值}\end{cases}$$

2.2.1 条件结构

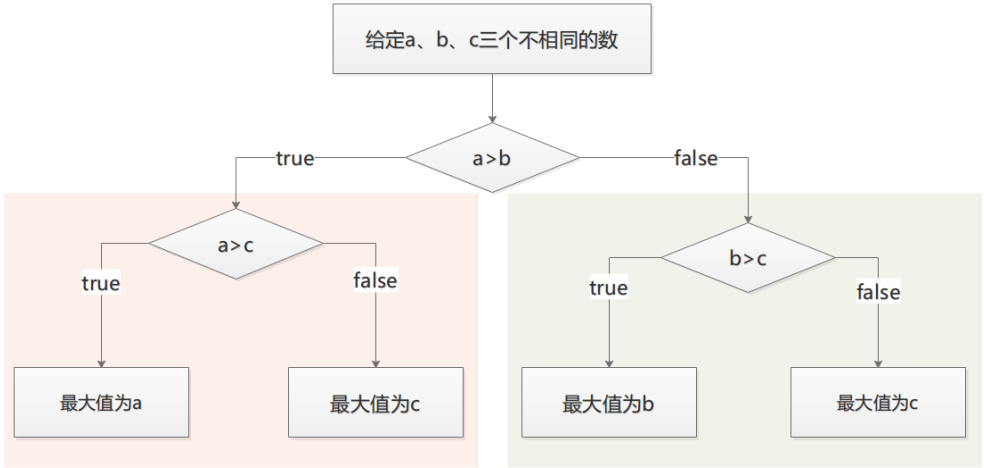
- if-elseif-else-end 语句：

if 和 elseif 后面的表达式也支持其他运算，例如算术运算，其计算结果可以是一个数值常数，不一定非要是逻辑值 1 或者 0。**如果 if 和 elseif 后面表达式的计算结果为非零数值，就会被当成逻辑值 1；如果计算结果为数值零，则会被当成逻辑值0.**

2.2.1 条件结构

➢ if-elseif-else-end 语句：

例题：（**if 的嵌套**）已知 a、b 和 c 是三个互不相等的常数，请使用 if 语句找出 a、b 和 c 三个数的最大值。



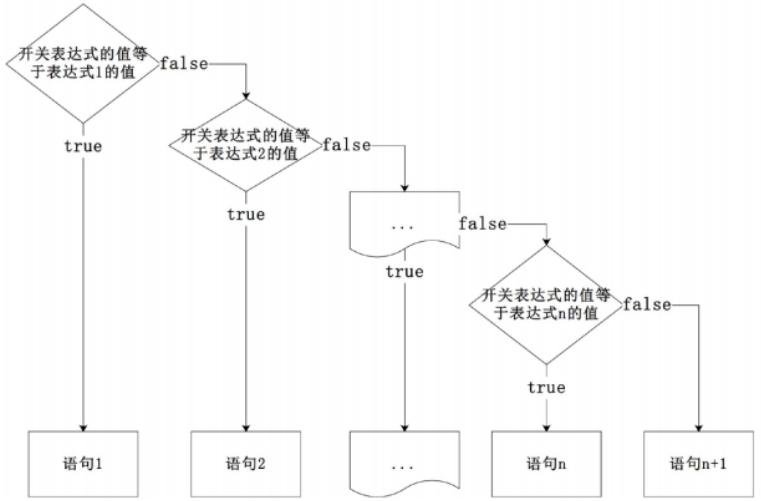
2.2.1 条件结构

➢ switch-case-otherwise-end 语句

开关表达式的计算结果必须是一个数值标量或者是一个字符向量/字符串，不能是向量或者矩阵。

可被 if 代替

```
switch 开关表达式
    case 表达式 1
        语句 1
    case 表达式 2
        语句 2
    …
    case 表达式 n
        语句 n
    otherwise
        语句 n+1
end
```



2.2.1 条件结构

- switch-case-otherwise-end 语句

例题：season 是在区间[1,4]上随机生成的一个整数，用来表示季节（春夏秋冬）。分别使用switch-case-otherwise-end 语句和if-elseif-else-end 语句来输出对应季度。

2.2.2 循环结构

➢ for-end 语句（for循环）

用于事先已知循环次数的情形，其语法如下：

```
for 循环变量 = 向量或者矩阵
    循环体
end    % 循环体以 end 结束，千万不能漏写了！
```

循环变量：用于迭代的变量名，它会在每次循环迭代中从向量或矩阵中取出一列的值。数值向量或者矩阵则表示了循环变量可以取值的范围，一旦循环变量遍历完数值向量或者矩阵中的所有值，循环就会结束。

2.2.2 循环结构

- for-end 语句（for循环）

例题：不使用 sum 函数，计算行向量 x 中所有元素的和。

例题：计算从公元 1 年到公元 9999 年间，有多少个闰年。闰年的判读条件是年份能够被4 整除，但不能被 100 整除，或者年份能够被 400 整除。

2.2.2 循环结构

➢ while-end 语句（while 循环）

允许我们在不知道具体循环次数的情况下执行循环体，其语法如下：

```
while 表达式
      循环体
end   % 循环体以 end 结束，千万不能漏写了！
```

2.2.2 循环结构

- while-end 语句（while 循环）

例题：根据斐波那契数列的递推公式，求数列中第一个大于 99999 的元素。

$$\begin{cases} F(1)=1, F(2)=1 \\ F(n)=F(n-1)+F(n-2),\ n \geq 3 \end{cases}$$

2.2.2 循环结构

➢ while-end 语句（while 循环）

允许我们在不知道具体循环次数的情况下执行循环体，其语法如下：

```
while 表达式
    循环体
end  % 循环体以 end 结束，千万不能漏写了！
```

- 一个无限循环，在命令行窗口中按下快捷键 Ctrl+C 来中断程序的运行。
- while 后面表达式的计算结果不一定非得是逻辑值 1 或 0。如果表达式的计算结果是一个数值常数，则只有当这个常数为非零值时循环才会进行。

2.2.2 循环结构

➢ break 和 continue

break 和 continue 只能与 for 循环或 while 循环一同使用

- break 关键字用于终止执行 for 或 while 循环。实际使用中，当满足某个条件时，我们会使用 break 立即退出循环。这在找到所需结果后立即退出循环的场景非常有用。

- continue 关键字用于跳过循环的当前迭代，然后继续下一次迭代。实际使用中，当满足某个条件时，continue 将跳过当前循环迭代的剩余部分，然后继续进行下一次迭代。这对于在某些情况下跳过特定的迭代非常有用，而不必完全退出循环。

2.2.2 循环结构

- break 和 continue

例题：输出1至10中所有的奇数。

例题：质数（Prime number），又称素数，指在大于 1 的自然数中，除了 1 和该数自身外，无法被其他自然数整除的数（也可定义为只有 1 与该数本身两个正因数的数）。给定任意一个大于 100 的自然数 n（例如 n=135389），请判断 n 是否为质数。

2.2.2 循环结构

- break 和 continue

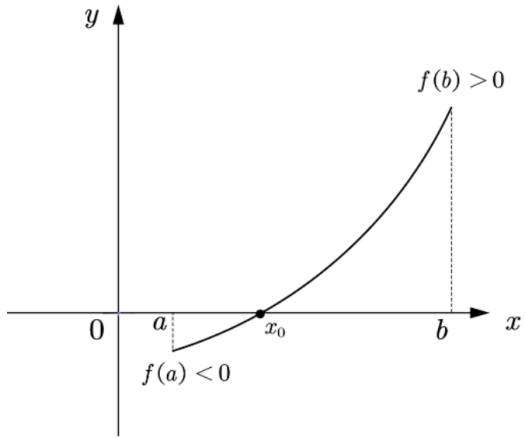
若存在循环的嵌套，break和continue仅在调用它的循环主体中起作用。即break仅从它所发生的循环中退出，continue仅跳过它所发生的循环体内的剩余语句。

2.2.3 本章小节

- 条件结构
  - if-elseif-else-end 语句：用于多个条件判断，根据条件的不同执行不同的代码块。
  - switch-case-otherwise-end 语句：将开关表达式的值依次和各个 case 后面的表达式的 值判断是否相等，如果相等则执行对应的代码块，都不相等则执行 otherwise 的代码。
- 循环结构
  - for-end 语句：用于循环次数已知的情形。
  - while-end 语句：适用于循环次数未知的情形，根据条件判断是否继续执行循环。
  - break 和 continue 的用法：灵活地控制循环过程的执行

2.2.4 课后小练

若函数$f(x)$在区间$[a,b]$上连续严格单调，且满足$f(a)\times f(b)<0$，那么$f(x)$在区间$[a,b]$上有且仅有一个零点。二分搜索法的基本思想是不断将区间$[a,b]$一分为二，然后判断零点位于哪一半区间内，接着继续将包含零点的那一半区间一分为二，如此循环，直到得到足够精确的零点的估计值。以下是二分搜索法的一般步骤：



2.2.4 课后小练

步骤 1：选择函数零点所在的初始区间$[a,b]$，确保$f(a)\times f(b)<0$。

步骤 2：计算区间的中点$c=(a+b)/2$，并计算函数在c处的值$f(c)$。

步骤 3：如果$f(c)$的值恰好等于零，或者$f(c)$的绝对值小于某个给定的误差阈值，那么c就可以当成零点，迭代结束。

步骤 4：如果$f(c)$与零的差异较大，那么需要根据$f(c)$的正负号，将原来包含零点的区间$[a,b]$更换为$[a,c]$或$[c,b]$，确保零点仍然在新的区间内（例如: $f(a)\times f(c)<0$则更换为$[a,c]$）。

步骤 5：重复步骤 2 到 4，直到找到零点或者达到所需的精度停止迭代。

下面看一个具体的题目：函数$f(x)=x^3-8x^2+x-5$，$f(x)$在区间$[6,10]$严格递增且$f(6)<0, f(10)>0$，请用二分搜索法求零点x_0（$f(x_0)$和 0 的误差控制在 1e-8 内即可）。

2.3.1 m文件定义函数

2.3.2 匿名函数

2.3.3 本章小节

2.3.4 课后小练

2.3.1 m文件定义函数

- Matlab函数的基本结构：

function[输出形参表: output1, ...,outptn] = 函数名(输入形参表: input1, ... , inputn)

注释说明部分
函数体代码部分

end

2.3.1 m文件定义函数

- 函数调用

定义完的函数保存为同函数名的m文件，放置在Matlab的当前路径之下，然后就和使用Matlab自带的函数一样，直接在命令窗口或程序代码中使用该函数就是调用它了。

函数调用的一般格式为：

[输出实参表]=函数名(输入实参表)

2.3.1 m文件定义函数

- 函数调用
  - 函数中遇return语句时，将退出函数体，此函数调用结束；
  - 函数体里面也可以定义一个或几个函数，称为子函数；注意：子函数只能存在于主函数体内，不独立存在；子函数在主函数体内的位置可以任意，不影响使用；子函数只能被主函数以及其他位于同一主函数体下的子函数调用；
  - 在调用函数时，Matlab用两个永久变量nargin和nargout分别记录调用该函数时的输入实参和输出实参的个数。

2.3.1 m文件定义函数

- 函数调用

例题：编写m-函数文件：max_min_values.m，用来求向量的最大、最小值。

2.3.2 匿名函数

匿名函数是不存储在程序文件中，数据类型是 function_handle 的变量相关的函数。代替“将函数编写为单独的m-文件”，并且效率比更高。

基本格式：

f=@(参数1, 参数2, …) 函数表达式

2.3.3 本章小节

常用的自定义函数方式有：

- m文件定义函数：无需关注具体实现过程，需要编写额外的.m文件，定义完的函数保存为同函数名的m文件，放置在Matlab的当前路径之下。

- 匿名函数：可以代替“将函数编写为单独的m-文件”，方便地对含参变量函数进行操作。

2.3.4 课后小练

对带参数a的参数方程：$e^x + x^a + x^{\sqrt{x}} = 100$，要求针对a在[0,2]上的不同取值求解方程。

2.4.特殊函数

特殊函数是一组在实际应用中经常出现的著名数学函数。可以用它们来计算贝塞尔函数、beta 函数、gamma 函数、误差函数、椭圆积分等。

函数的详细信息：NIST Digital Library of Mathematical Functions

贝塞尔函数

airy	Airy 函数
besselh	第三类贝塞尔函数（汉克尔函数）
besseli	第一类修正贝塞尔函数
besselj	第一类贝塞尔函数
besselk	第二类修正贝塞尔函数
bessely	第二类贝塞尔函数

2.5.常用函数

函数名	功能	计算结果
abs	求绝对值，也可以用来计算复数的模长	abs(1.5) % 1.5 abs(-1.5) % 1.5 abs(3+4i) % 5
mod	mod(a,m)可以计算 a 除以 m 后的余数，其中 a 是被除数，m 是除数。	mod(11, 3) % 2 mod(9, 3) % 0
sqrt	sqrt(a)可以计算 a 的平方根，即对 a 开根号。如果 a 为负数则返回复数结果。其结果和 a^(1/2)等价。	sqrt(9) % 3 format long g % 计算结果显示为长格式 sqrt(2) % 1.414213562373095 sqrt(-4) % 2i
exp	exp 函数可以计算以自然常数 e 为底的指数。	exp(1) % 2.7183 exp(2) % 7.3891 exp(10) % 2.2026e+04

2.5.常用函数

log	log(x)用来计算以自然常数 e 为底数的对数。	log(2)　　% 0.6931 log(3)　　% 1.0986 log(exp(10))　% 10
log2 / log10	分别用来计算以 2 和 10 为底的对数。	log2(4)　　% 2 log2(1024)　% 10 log10(100)　% 2
round	真正意义上的四舍五入函数（将结果四舍五入为最近的整数，如果为 0.5，则会朝着偏离零的方向调整） round 函数还有第二种用法，它可以输入第二个参数： round(X,N)可以将 X 在第 N 位数四舍五入，分下面三种情况： 1) N＞0：四舍五入到小数点右侧的第 N 位数。 2) N＝0：四舍五入到最接近的整数。 3) N＜0：四舍五入到小数点左侧的第 N 位数。	% 只有一个输入参数的用法： round(1.1)　% 1 round(1.9)　% 2 round(-1.1)　% -1 round(-1.9)　% -2 round(1.5)　% 2 round(-1.5)　% -2 % 有两个输入参数的用法： round(3.14159, 1)　% 3.1 round(3.14159, 3)　% 3.142 round(3.14159, 0)　% 3 round(12345.6, -1)　% 12350 round(12345.6, -2)　% 12300 round(12345.6, -3)　% 12000

isempty 函数

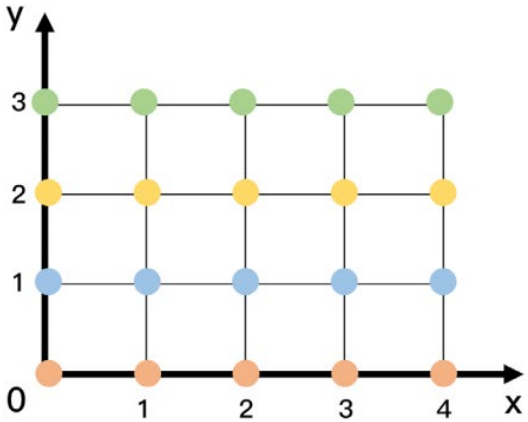
如果 A 为空数组 [ ]， isempty(A) 返回逻辑值 1 (true)，否则返回逻辑值 0 (false)。length(A) == 0 的返回结果和 isempty(A)的返回结果一样，MATLAB 推荐大家使用后者判断 A 是否为空数组[ ]，后者的运行效率更高。

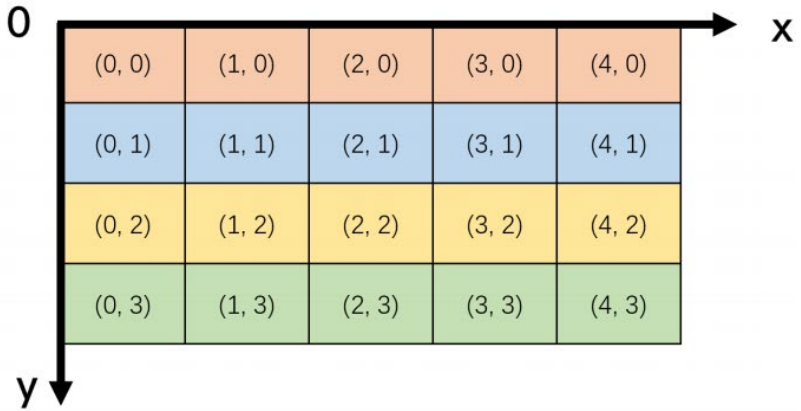
例题：判断常数 x 是否是数组 A 中的某个元素，如果是则返回逻辑值 1，不是则返回逻辑值 0。

meshgrid 函数

meshgrid 函数可以基于向量 x 和 y 中包含的坐标来返回二维网格坐标。

例子：假设 x 轴坐标上的取值是[0 1 2 3 4]，y 轴坐标上的取值是[0 1 2 3]，现在请使用 x 轴坐标和 y 轴坐标共同创建下图所示的二维网格坐标：





rng 函数

rng 函数可用来设置随机数种子，这样能生成可重复的随机数。

使用随机数生成函数(例如 rand, randi 等)之前，使用 rng(seed)命令设置随机数种子，这样能保证生成的随机数被固定下来。设置不同的随机数种子生成的随机数通常都不相同。

拓展：每次重新启动 MATLAB 时，随机数生成器均复位到相同的状态，这样使用生成随机数的命令会返回相同的结果。我们可以使用 rng('shuffle') 命令，它可以根据当前的时间使用不同的种子重新设定生成器的种子，这样能避免重复生成相同的随机数。

本章小节

- 特殊函数：MATLAB帮助中心的使用，贝塞尔函数、beta 函数、gamma 函数。
- 常用函数：数值函数，三角函数，isempty ，meshgrid和rng函数。

课后小练

使用ismember函数判断常数 x 是否是数组 A 中的某个元素，如果是则返回逻辑值 1，不是则返回逻辑值 0。