# •

# 开始

如果你是MATLAB的初学者,你应该从阅读矩阵操作开始。最重要的是要学习如何输入矩阵,如何使用:(冒号)算子,以及如何调用函数。 在你掌握了基础之后,你应该阅读其他章节并运行实例。

MATLAB的中心是一种在你能充分开发它的功能之前必须学习的工具。你能迅速学习MATLAB的基础并掌握它。你将会得到的回报是高产量以及高创造性用计算机计算的能力,这将会改变你工作的方式。

- 导言 描述MATLAB系统的组成。
- 开发环境- 介绍MATLAB的开发环境,包括有关工具和MATLAB桌面的信息。
- 矩阵操作- 介绍如何用MATLAB生成矩阵以及用矩阵进行数学运算。
- 作图- 介绍MATLAB的作图能力,包括有关作图数据,给图表加注释,以及图像的使用。
- <u>用MATLAB编程</u> 描述如何用MATLAB语言编写手稿和函数,以及利用数据结构,例如单元数组和多元数组。

### 有关信息

更多关于此收藏中的主题的指导信息另见Using MATLAB。 对于有关MATLAB函数的参考信息,见Reference。





### 冒号算子

冒号': '是MATLAB里面一个十分重要的算子,适用于各种不同的表达式。 例如表达式

1:10

表示一行从1到10的整数

1 2 3 4 5 6 7 8 9 10

为了改变递变的间隔,可以指定一个间隔长度,例如

100:-7:50

表示为

100 93 86 79 72 65 58 51

而

0:pi/4:pi

表示

0 0.7854 1.5708 2.3562 3.1416

含有冒号的下标表达式也可以用来表示矩阵的一部分

A(1:k, j)

i表示A的第j列前k个元素,因此

sum(A(1:4,4))

表示计算a的第4列元素的和。但是这里有一种更加方便的办法。关键字':'本身可以表示某行或某列的全体元素,而我们可以用关键字end来表示最后一行或者最后一列的全体元素,所以

sum(A(:,end))

是计算a的最后一列元素的和。故有

ans = 34

为什么一个4×4的魔方矩阵的魔术和是34呢?如果我们从1加到16再把答案除以4,即

sum(1:16)/4

答案当然是

ans = 34

▼ 下标

魔方函数▶

Getting Started

+ +

# 动画

MATLAB提供两种生成生动活泼动画的方法:

- 在屏幕上不断擦除再重画对象,每次重画作增量变化。
- 保存许多不同图, 然后作为电影回放。

● 图形用户界面

删除模式方法▶

Getting Started





# MATLAB系统

MATLAB系统包括五个主要部分:

开发环境. 这是一组帮助你使用MATLAB的函数和文件的工具和设备。这些工具大部分是图形用户界面。它包括MATLAB桌面和命令窗口,命令历史,和用于查看帮助的浏览器,工作空间,文件和查找路径。

MATLAB数学函数库. 这里汇集了大量计算的算法,范围从初等函数如:求和,正弦,余弦和复数的算术运算,到复杂的高等函数如:矩阵求逆,矩阵特征值,

贝塞尔(Bessel)函数和快速傅立叶变换等。

MATLAB语言. 这是一种高水平的矩阵/数组语言,含有控制流语句,函数,数据结构,输入/输出,和面向对象编程特征。它允许"小型编程"以迅速创立快速抛弃型程序,以及"大型编程"以创立完整的大型复杂应用程序。

**句柄制图®**. 这是MATLAB制图系统。它包括高级别的二维、三维数据可视化,图像处理,动画,以及表现图形的命令。它还包括低级别的命令,这使你不但能在MATLAB的应用中建立完整的图形用户界面,而且还能完全定制图形的外观。

MATLAB应用程序界面(API). 这是使你编写与MATLAB相合的C或Fortran程序的程序库。它包括从MATLAB中调用程序(动态链接),调用MATLAB为计算引擎,和读写MAT-文件的设备。

● 导言

MATLAB文件▶

Getting Started





### MATLAB在线帮助

要查看在线文件,在MATLAB菜单栏中选择Help选项。(更多关于使用在线文件的信息见帮助浏览器.)

在"用MATLAB"下,文件主要由以下主题构成:

- 开发环境- 提供关于MATLAB桌面的全部信息。
- 数学- 描述如何使用MATLAB的数学和统计功能。
- 编程和数据类型- 描述如何用MATLAB语言创建脚本和函数。
- 作图- 描述怎样用MATLAB的图形功能绘制数据图。
- <u>3D可视化</u>- 介绍怎样用查看,光照,透明等函数达到比仅用基本作图函数更复杂的图形效果。
- <u>外部界面/API</u> 描述MATLAB的下列界面: C和Fortran程序, Java类别和对象, 数据文件, 串行端口I/0, ActiveX(网络化多媒体对象技术),和DDE(动态数据交换)。
- 创建图形用户界面- 描述如何用MATLAB的图形用户界面设计工具。

在"参考"下,文件主要由以下主题组成:

• MATLAB函数参考- 覆盖MATLAB的所有核心函数,提供有关函数句法,详细资料,数学算法(恰当时),以及相关函数。

你可以用"函数类别"或"函数字母列表"简单地定位任何函数。

• <u>外部界面/API参考</u>- 覆盖被MATLAB用于外部界面的函数,提供关于调用语言的句法,详细资料,参数,返回值和例子的信息。

MATLAB在线帮助也包括"图形对象属性浏览器",它使你能容易地访问图形对象属性地详细资料。更多关于MATLAB图形地信息,见操作图形。

MATLAB文件

开发环境▶

Getting Started



导言

# MATLAB是什么?

MATLAB® 是一种对技术计算高性能的语言。它集成了计算,可视化和编程于一个易用的环境中,在此环境下,问题和解答都表达为我们熟悉的数学符号。典型的应用有:

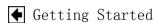
- 数学和计算
- 算法开发
- 建模,模拟和原形化
- 数据分析,探索和可视化
- 科学与工程制图
- 应用开发,包括图形用户界面的建立

MATLAB是一个交互式的系统,其基本数据元素是无须定义维数的数组。这让你能解决很多技术计算的问题,尤其是那些要用到矩阵和向量表达式的问题。而要花的时间则只是用一种标量非交互语言(例如C或Fortran)写一个程序的时间的一小部分。.

名称 "MATLAB"代表  $matrix\ laboratory$  (矩阵实验室)。MATLAB最初是编写来提供给对由LINPACK和EINPACK工程开发的矩阵软件简易访问的。今天,MATLAB使用由LAPACK和ARPACK工程开发的软件,这些工程共同表现了矩阵计算的软件中的技术发展。

MATLAB已经与许多用户输入一同发展了多年。在大学环境中,它是很多数学类、工程和科学类的初等和高等课程的标准指导工具。在工业上,MATLAB是高产研究、开发和分析所选择的工具。

MATLAB以一系列称为工具箱的应用指定解答为特征。对多数用户十分重要的是,工具箱使你能学习和应用专门的技术。工具箱是是MATLAB函数(M-文件)的全面的综合,这些文件把MATLAB的环境扩展到解决特殊类型问题上。具有可用工具箱的领域有:信号处理,控制系统神经网络,模糊逻辑,小波分析,模拟等等。



MATLAB系统 ▶





# 动画

MATLAB提供两种生成生动活泼动画的方法:

- 在屏幕上不断擦除再重画对象,每次重画作增量变化。
- 保存许多不同图, 然后作为电影回放。

● 图形用户界面

删除模式方法▶

Getting Started





# 桌面工具

这一节给出了MATLAB桌面工具的介绍。你还可以通过运用MATLAB里的函数去执行桌面工具的大部分特征。这些工具是:

- 命令窗口
- 命令历史
- 发射台
- 帮助浏览器
- 当前目录浏览器
- 工作空间浏览器
- 数组编辑器
- 编辑器/调试器

**▲** MATLAB桌面

命令窗口▶

Getting Started

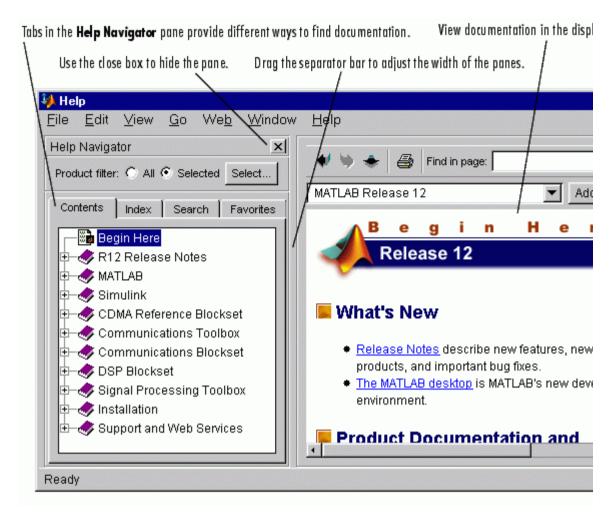




### 帮助浏览器

用帮助浏览可以在你所有的MathWorks 产品中找到并查看文档。帮助浏览器是一个集成在MATLAB桌面中显示HTML文件的网页浏览器。

要打开帮助浏览器,可以在工具栏中点击?或者在控制窗口中输入helpbrowser。



帮助浏览包含两个控制台,帮助导航--用来查询信息,以及显示控制台--可以查看信息。

### 帮助导航

利用帮助导航可以查询信息,它包括:

- 产品过滤器一设置过滤器可以显示那些你符合分类的产品。
- 内容标签-查看要求产品的名字并且把文档内容制表。
- **索引标签**-在Mathworks文档中查找对你需要的产品特定的索引入口(选定关键词)。
- 搜索标签-在文档中寻找特定词语。要得到特定功能帮助,设置"搜索类型"到"搜索名字"。
- 收藏标签-查看以前设定的收藏文档的列表。

### 显示控制台

在用帮助导航查找到文档之后,在显示控制台查看。在查看文件的同时,可以:

- 浏览其他页-用这些页的上下部的箭头,或者用工具栏的前后键。
- 书签页一点击工具栏"添加到收藏夹"键。
- 打印页 一点击工具栏"打印"键。
- 在该页查询词一在工具栏的"在页搜索"区键入搜索词,然后点击Go。

其他在显示控制台的可用功能有: 复制信息, 提炼选择内容, 以及浏览网页。

### 更多帮助

除了帮助浏览之外,还可以用帮助函数。要得到一个特定函数的帮助信息,用doc。例如,doc format 是显示在帮助浏览器中关于格式函数的帮助信息。其他得到帮助信息的方式包括联系技术支持(http://www.mathworks.com/support)以及参加MATLAB用户而设的新闻组: comp.soft-sys.matlab。

★ 发射台

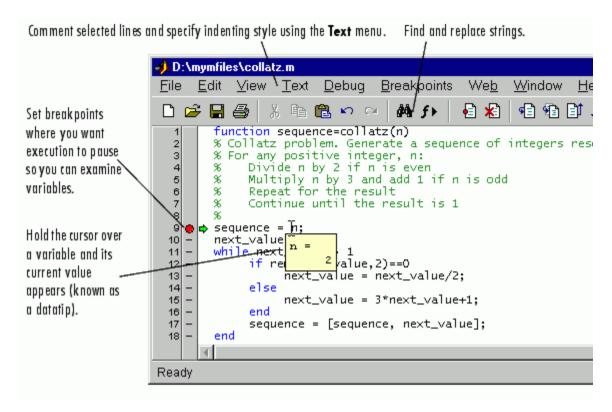
当前目录浏览器▶

Getting Started



### 编辑器/调试器

编辑器/调试器是用来生成和调试M-文件的,这些M-文件是你为运行MATLAB函数所写的程序。编辑器/调试器为基本文字编辑和M文件的调试提供图形用户界面。



你可以使用任何文档编辑器来生成M-文件,例如Emacs,也可以用参数选择(可从桌面文件菜单得到)来指定默认的编辑器。如果用的是别的编辑器,你仍然可以使用MATLAB的编辑器/调试器来调试,或者使用用来插入断点的调试程序,例如dbstop。

如果仅仅是要看看M-文件的内容,你可以用type命令使它命令窗口中显示。

▲ 工作空间浏览器

其他开发环境特征▶

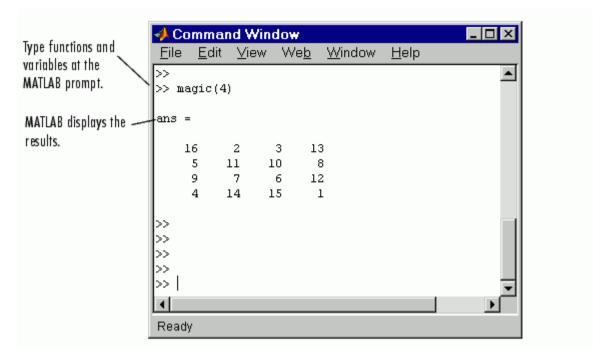
Getting Started





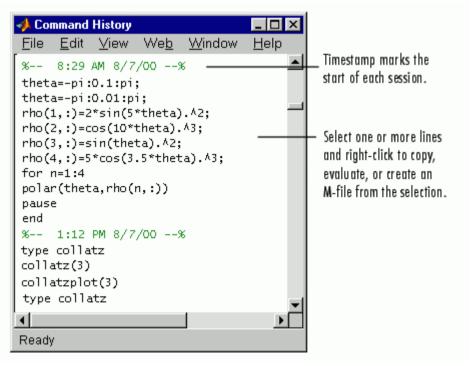
### 命令窗口

运用命令窗口去调入变量,运行函数和M-文件。如要得到更多的关于控制输出与输入的信息,请看控制命令窗口输入与输出。



### 命令历史

在控制窗的输入内容都被记录在 命令历史窗口中。在命令历史中,你可以看到以前用过的函数,并且可以复制和执行选定行。



要从MATLAB任务到一个文件保存输入与输出的内容,可以用diary函数。

#### 运行外部程序

你可以在MATLAB控制窗口中运行外部程序。感叹号!为起点就可以跳出当前程序,也意味着之后的输入内容就是对操作系统的的命令。这对于调用效用或运行其他程序而不关闭MATLAB有益。例如,在Linux中:

!emacs magik.m

调用一个叫emacs的编辑器来对magik.m文档操作。当你关闭外部程序时,操作系统会自动回到MATLAB。

● 桌面工具

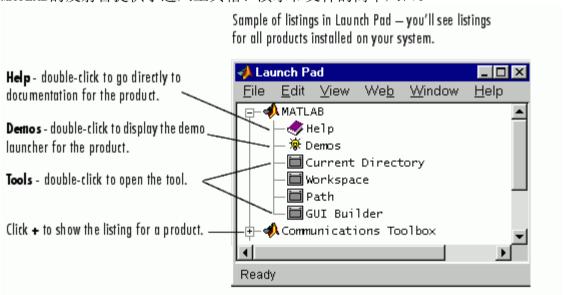
发射台▶

Getting Started



### 发射台

MATLAB的发射台提供了进入工具箱、演示和文件的简单入口。



● 命令窗口

帮助浏览器▶

Getting Started





# 当前目录浏览器

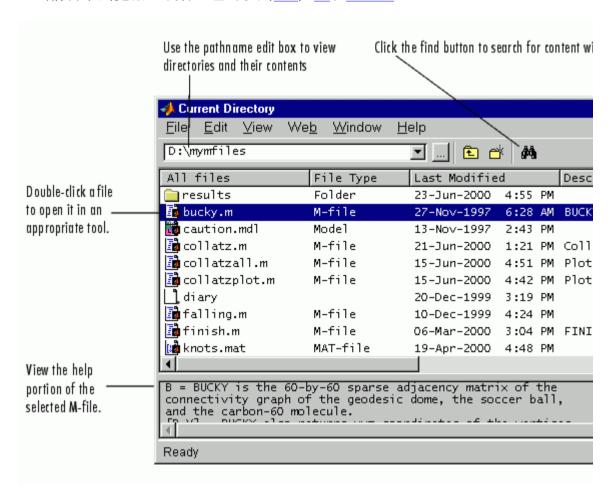
MATLAB文档的所有操作是以当前目录或者搜索路径作为参考点的。你需要运行的

任何文档都必须在当前目录下或者搜索目录下完成MATLAB。

一种快速查看或者改变当前路径的方法是用下面所示工具栏中的"当前目录"。



如果要搜索、查看、打开或者改变MATLAB相关目录以及文档,就要用到MATLAB "当前目录浏览器"或者,也可以用dir, cd和delete。



#### 搜索路径

要决定怎样去执行你所调用的函数,MATLAB利用"搜索路径"去查找在文件系统目录中已经安排好的M-文件和MATLAB相关文件。任何要用MATLAB运行的文档都必须在当前目录下或者在搜索路径里的目录中。在默认情况下,MATLAB和MathWorks工具箱所以支持的文档都在搜索路径中。

要查看哪个目录在搜索路径下或者要改变搜索路径,请在桌面File菜单中选择Set Path,然后利用Set Path对话框。当然,你可以用path函数来查看搜索路径,用

addpath把目录加到路径下,用rmpath可以从路径中删除目录。

▼ 帮助浏览器

工作空间浏览器▶

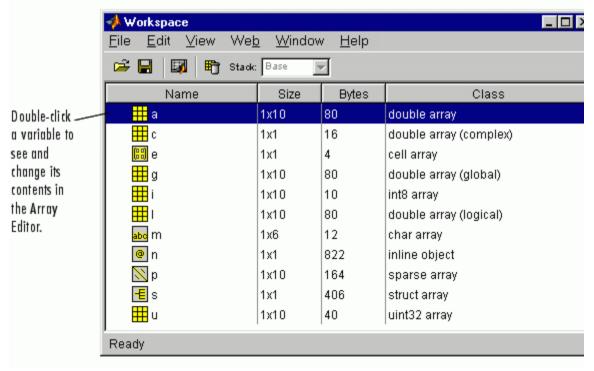
Getting Started



### 工作空间浏览器

MATLAB工作空间包含内建MATLAB处理任务时以及储存在记忆体中的变量排列。你可以通过使用函数、运行M-文件、装载已存文件等方式增加变量到工作区。

要查看关于每个变量的工作空间和信息,请用工作空间浏览器,或用<u>who</u>和<u>whos</u>函数。

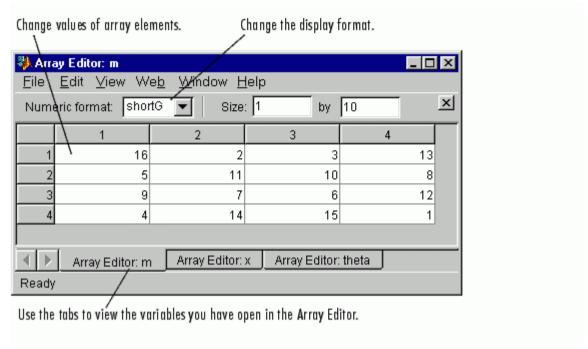


要在工作空间删除一个变量,先选择那个变量,然后从Edit菜单中选择Delete,或者用<u>clear</u>命令。

当你退出MATLAB后,工作空间的内容是不会继续存在的。如果要把工作空间保存到一个文件里,以便下次的MATLAB任务中能够读出来,选择File菜单中Save Workspace As的,或者用<u>save</u>命令。这把工作空间保存到一个叫做MAT-文件的二进制文件,它的后缀名是. mat。另外,还有选项可以把文件保存成不同的格式的。要读出一个MAT-文件,选择File菜单中的Import Data,或者用load命令。

#### 数组编辑器

在工作空间浏览器里双击一个变量后,会弹出一个数组编辑器。你可以用这个数组编辑器来查看或者编辑在工作空间里的一维或者二维的数值矩阵,字符串和字符串的单元数组的直观表示。



● 当前目录浏览器

编辑器/调试器▶

Getting Started





# 其他开发环境特征

其他开发环境特征有:

- <u>Importing and Exporting Data</u> 通过其他的操作引入或者导出已被创建的数据到工作界面,包括引入向导,以及其他命令引进的数据包。
- <u>Improving M-File Performance</u> Profiler是一种衡量文件运行时间的工具,使用它能帮助你提高运行速度。
- <u>Interfacing with Source Control Systems</u> 从MATLAB, Simulink<sup>®</sup>,和 Stateflow<sup>®</sup>中存取资源控制系统。
- <u>Using Notebook</u> 从文字处理器的环境(Microsoft Word)存取MATLAB的数值计算和可视化软件进行MATLAB的数字运算,使整个运算过程能很清晰的显示出来。

矩阵操作▶

Getting Started

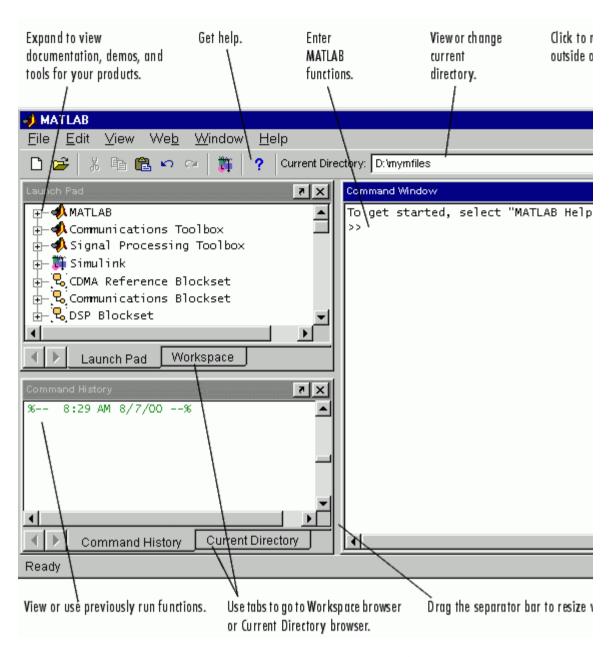




# MATLAB桌面

当你进入MATLAB,MATLAB桌面显现出来,包含管理文件的工具(图形用户界面),变量,以及与MATLAB关联的应用软件。

第一次启动MATLAB时,即使你的发射台窗口可能包含不同的条目,桌面也出现如下显示图象。



你可以通过打开,关闭,移动和调整界面里的工具来改变界面的外观。你还可以将工具移动出界面或返回界面里(称为"入船坞",docking)。所有的界面工具提供共同的特征,如上下文菜单和键盘的快捷操作。

你可以通过File菜单里的参数选择为界面工具指定某一个特征。例如,你可以为命令窗口文本指定字体特征。如要得到更多信息,请点击参数选择对话框中的 Help按钮。 Getting Started





# 开发环境

# 导言

这一章简短地介绍启动和退出MATLAB,并介绍以及帮助你运行MATLAB中的变量和 文件的工具和函数。如果想得到更多关于这里提到的主题的信息,请看MATLAB文 件中的Development Environment相应主题,也可以在线查阅。

# 启动与退出MATLAB

### 启动MATLAB

在微软Windows平台下,要启动MATLAB,请双击在Windows桌面上的MATLAB快捷方式图标♪。

在UNIX平台下,要进入MATLAB,在操作系统提示符处输入matlab。

启动MATLAB后, MATLAB窗口打开了-----请看MATLAB桌面。

你可以改变进入MATLAB启动的路径,定义包含一个在启动上的脚本的启动选项, 从而在某些状况下减少启动时间。

### 退出MATLAB

要结束MATLAB,从桌面File菜单选择Exit MATLAB,或者在命令窗口中输入quit。要在MATLAB每次退出时执行指定的函数,譬如保存工作空间,你可以建立并运行一个finish.mpa本。

**▲** MATLAB在线帮助

MATLAB桌面▶

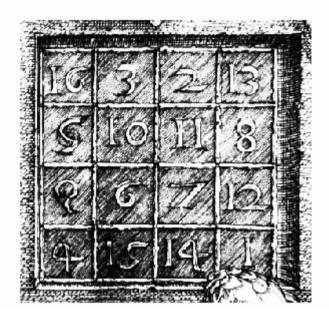


# 矩阵操作

# 矩阵和魔方

在MATLAB里,矩阵由数字行列构成。有时也指1\*1矩阵,它是标量,而一行或一列的矩阵指代向量。MATLAB有另外一些存储数字与非数字数据的方法,但在开始的阶段,最好将一切以矩阵看待。MATLAB内的操作被设计得尽可能接近自然语言。别的程序设计语言以一次处理一个数据,MATLAB则让你简单而快捷地处理一整个矩阵。有一个贯穿此书的很好的例子,它出现在德国的一个艺术家兼业余数学爱好者阿布列西特. 杜勒的作品一文艺复兴雕版画一忧郁者I中。





这幅图布满了各种各样的数学符号,而且如果你仔细观察的话就会发现在右上角 有一个矩阵。那个就是奇妙的魔方矩阵,早在杜勒的时代人们就相信它有不可思 议的性质。而它的确具有值得人们去研究的神奇特性。

▼ 其他开发环境特征

输入矩阵▶

Getting Started





# 矩阵的利用

这部分将介绍矩阵的其他产生方法。

▼ 表达式举例

生成矩阵▶

Getting Started

•

### 生成矩阵

MATLAB提供产生基本矩阵的四个函数:

zeros	产生一个全零矩阵	
<u>ones</u>	产生一个全1矩阵	
rand	产生一个元素是均匀分布随机数的矩阵	

### 例如:

### ₩ 矩阵的利用

加载命令▶

### Getting Started





### 加载命令

加载命令<u>load</u>读出含有由早期MATLAB版本形成的矩阵的二进制文件,或者读出含有数值数据的文本文件。文本文件应该由一个数字的矩阵表格构成,由空格隔开,每一线排一行,而且每一行有相同数量的元素。例如,在MATLAB外部,创建一个含有这四行的文本文件。

16.0	3.0	2.0	13.0
5.0	10.0	11.0	8.0
9.0	6.0	7.0	12.0
4.0	15.0	14.0	1.0

以magik.dat为名储存这个文件。而命令

load magik.dat

读这个文件并创建一个含有我们的矩阵例子的变量magik。

在许多文本或者二进制格式中,一个把数据读进MATLAB的简易方法师用输入向导。

▼ 生成矩阵

M−文件▶

Getting Started



#### M-文件

你可以用M-文件创建你自己的矩阵,这些M-文件是包含MATLAB代码的文本文件。 用MATLAB编辑器或另一种文本编辑器去创建一个文件,它包含你将输入到MATLAB 命令行的同样的语句。用一个后缀名为. m的名字保存该文件。

例如, 创建一个文件含有这五行。

用名字magik.m保存该文件,然后语句

magik

读出该文件并产生一个变量A,含有我们的矩阵示例。

▼ 加载命令

连接▶

Getting Started





# 连接

连接是一个合并小矩阵成大矩阵的过程。事实上,你创建第一个矩阵时就是通过将它的各自独立的元素连接在一起的。中括号[]是连接的算子。例如,以4\*4魔方A开始,构造

 $B = [A \quad A+32; A+48 \quad A+16]$ 

结果是含有加入的四个子矩阵的8×8矩阵。

B =

16	3	2	13	48	35	34	45
	•	_					
5	10	11	8	37	42	43	40
9	6	7	12	41	38	39	44
4	15	14	1	36	47	46	33
64	51	50	61	32	19	18	29
53	58	59	56	21	26	27	24
57	54	55	60	25	22	23	28
52	63	62	49	20	31	30	17

这个矩阵仅是成为另一个魔方的过渡。它的元素是一群重新整理的1:64的整数。它的列加起来是一个8×8魔方的修正值。

sum(B)

ans =
260 260 260 260 260 260 260 260

但是它的行加起来sum(B')',并不是相同的。因此要做一个有效的8×8魔方需要更多的操作。

**★** M-文件

行与列的删除▶

Getting Started







你只要使用一对方括弧就能够删除一个矩阵的行和列。以

$$X = A;$$

开始。然后,删除X的第二列,使用

$$X(:,2) = []$$

这使X变为

如果你删除矩阵的单个元素, 其结果就不再是矩阵了。所以, 如下的表达式

$$X(1,2) = []$$

导致错误。然而,用单独的下标则是删除一个单独元素或连续的元素,并把剩下的元素改成行向量的形状。因此

$$X(2:2:10) = []$$

的结果是

$$X = 16 9 2 7 13 12 1$$

● 连接

关于矩阵和数组的其他内容▶

Getting Started



•



这一节将给你展现更多关于矩阵和数组的内容。集中在以下内容:

- 线性代数
- 数组
- 多元数据

◆ 行与列的删除

线性代数▶

Getting Started





# 线性代数

简单的说,矩阵和数列这两个术语是经常可以替换使用的。更准确地说,矩阵是一个表示线性变换的二维数字数组。矩阵定义下的数学运算是线性代数的主题。

### 杜勒的魔方

提供了几个例子,给出了MATLAB矩阵运算的一点味道。你已经看到了矩阵转置 A',加一个矩阵到它的转置得到一个对称矩阵。

乘号\*表示包括行与列的内积矩阵的乘法。用矩阵去乘它的转置也能得到对称矩阵。

这矩阵的行列式为0,表示该矩阵是奇异阵。

$$d = \det(A)$$

$$d = 0$$

A的简化行阶梯型不是单位矩阵。

由于矩阵是奇异的,因而它没有逆矩阵。如果你试图计算它的逆:

$$X = inv(A)$$

你会得到警告信息:

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 1.175530e-017.

四舍五入的误差从精确奇异性的探测中阻止了矩阵转置的算法。但是rcond的值(表示倒数条件估计)是和eps即浮点相对精度同阶数的,因此计算得的逆不太可能有用。

魔方的特征值很有趣。

-8.0000

特征值之中有一个为零。这是奇异性的另一个结论。最大的特征值为34,为幻和。那是因为元素全为1的向量是特征向量。

当魔方按照它的幻和为比例分配时,

$$P = A/34$$

结果是双重随机矩阵,其行与列之和全是1。

```
P =
    0.4706
              0.0882
                        0.0588
                                  0.3824
                        0.3235
    0.1471
             0.2941
                                  0.2353
    0.2647
              0.1765
                        0.2059
                                  0.3529
    0.1176
              0.4412
                        0.4118
                                  0.0294
```

这样的矩阵表示转换的概率是一个马尔科夫(Markov)过程。矩阵重复的幂次表示过程重复的步骤。对于我们的例子,第5次方

 $P^{5}$ 

是

0.2507	0.2495	0.2494	0.2504
0.2497	0.2501	0.2502	0.2500

0. 2500 0. 2498 0. 2499 0. 2503 0. 2496 0. 2506 0. 2505 0. 2493

这表示当k趋于无穷大时,所有元素的k次方产趋于1/4。

最后,特征多项式中的系数

poly(A)

是

这表明特征多项式

$$\det(A - \lambda I)$$

是

$$\lambda^4 - 34\lambda^3 - 64\lambda^2 + 2176\lambda$$

常数项是零,因为矩阵是奇异的,立方项系数是-34,因为矩阵是魔方!

### ● 关于矩阵和数组的其他内容

数组▶

Getting Started





# 数组

当把矩阵从线性代数的世界中"取"出来时,它们变成二维数组。数组上的算术运算是元素对元素进行的。也就是说加减对于数组和矩阵是一样的,但乘法运算则不同。MATLAB用一个点或小数点作为数组乘法运算符号。

### 一系列的算子包括:

+	加法
_	减法
.*	元素对元素乘法
./	元素对元素除法
.\	元素对元素左除

. ^	元素对元素求幂
	不取共轭的数组转置

如果杜勒的魔方用数组乘法自乘

A. \*A

结果是包括1到16的平方的不寻常次序的数组。

### 建立表格

数组运算对建立表格很有用。设n是列向量

$$n = (0:9)$$
';

那么

pows = 
$$[n \quad n.^2 \quad 2.^n]$$

建立一个平方以及2的乘方的数表。

初等数学函数在数组上是元素对元素运算的。故

建立一个对数表。

logs	=	
	1.0	0
	1. 1	0.04139
	1.2	0.07918
	1.3	0.11394
	1.4	0. 14613
	1.5	0.17609
	1.6	0.20412
	1.7	0. 23045
	1.8	0. 25527
	1.9	0.27875
	2.0	0.30103

◀ 线性代数

多元数据▶

#### Getting Started



### 多元数据

MATLAB对多元统计数据用列导向分析。数据集中每一列表示一个变量,每一行则表示观察值。第(i,j)个元素是第i次观察值的第j个变量。

作为一个例子,考虑一个三变量数据集:

- 心率
- 体重
- 每周锻炼小时数

对5次观察值,结果数组可能如下:

第一行包含患者1的心率,体重和锻炼时间,第二行含有患者2的数据,等等。现在你能对这个数据集用MATLAB的多种数据分析函数。例如,计算每列的均值和标准差:

5.6303

25, 499

2, 2107

#### 要查看MATLAB中可用的一系列数据分析函数,键入

help datafun

如果你想使用统计工具箱, 键入

help stats



标量扩张▶

#### Getting Started



### 标量扩张

矩阵和标量可用几种不同方法合并。例如,通过从矩阵中每元素减去同一个数,矩阵就减去了一个标量。我们的魔方中元素的平均值是8.5,因此

$$B = A - 8.5$$

组成一个列向量和为零的矩阵。

sum(B)

ans = 
$$0 0 0 0$$

通过标量扩张,MATLAB把一个特定的标量在一定范围内赋给所有指标,例如:

$$B(1:2,2:3) = 0$$

使B的一部分变成0。

### Getting Started





### 逻辑下标

从逻辑和有关运算创建的逻辑向量可以用于参考子数组。设X是普通的矩阵,L是由一些逻辑运算得到的同尺寸的矩阵。那么X(L)指定X的元素中L的元素非零的部分。

通过指定逻辑运算为下标表达式,这种下标可以一步完成。设有下列数据集。

$$x = 2.1 1.7 1.6 1.5 NaN 1.9 1.8 1.5 5.1 1.8 1.4 2.2 1.6 1.8$$

NaN 表示一次丢失的观察值,譬如对一份问卷调查中的某项未答。要用逻辑标定指数移除丢失数据,用finite(x),它对于所有有限数值和NaN及Inf(无穷大)都正确。

这里还有一个观察值5.1,它似乎和其他有很大差别。它是一个局外者。以下语句 把局外者删除,在此例中是指那些到均值距离超过标准差3倍的元素。

另一示例是,通过用逻辑标定指数和标量扩张把杜勒的魔方中的非素数变成0,来突出素数的位置。

▼ 标量扩张

find函数 🕨





### find函数

<u>find</u>函数确定已给逻辑条件的数组元素的指标。 以它最简单的形式,返回一个指标的列向量。求这个向量的转置以获得一个指标的单行矩阵。例如:

用一维标定指数挑选出素数在魔方中的位置。

以按照k决定的次序的行向量展示这些素数,有

$$A(k)$$
 ans =  $5 3 2 11 7 13$ 

当用k作为一个指定陈述的左侧指标时,这个矩阵结构被保存起来。

▼ 逻辑下标

控制命令窗口输入与输出▶

Getting Started



# 控制命令窗口输入与输出

现在, 你已经在运用 MATLAB 的命令行了, 输入命令语句, 就会看到结果出现在命令窗口中。 这一节介绍如何:

- 控制输出值的格式
- 用MATLAB命令隐藏输出
- 在命令行中输入长的命令
- 编辑命令行

#### Getting Started





### 格式命令

format 命令控制了由 MATLAB 显示的数据的数值格式。这个命令只能影响到数据的显示结果,而与 MATLAB 如何计算及储存它们无关。这里是一些不同的数值格式,以及对一由不同数量级的元素组成的向量x使用这些数值格式所得到的输出结果。

注 为了保证合适的间隔,下面用的是固定宽度的字体,如 Fixedsys 和 Courier。

 $x = [4/3 \ 1.2345e-6]$ 

format short

1. 3333 0. 0000

format short e

1. 3333e+000 1. 2345e-006

format short g

1. 3333 1. 2345e-006

format long

format long e

1. 333333333333333e+000 1. 23450000000000e-006

format long g

1. 33333333333333

1.2345e-006

format bank

1. 33 0. 00

format rat

4/3 1/810045

format hex

如果矩阵中最大的元素比10<sup>3</sup>大,或者最小的元素比10<sup>-3</sup>小,那么 MATLAB 会为这个最大或最小的元素乘上一个公共的系数。

除了以上的格式命令之外

format compact (格式集成)

消除在输出结果中的空白行。这样你可以在一页或一个窗口中看到更多的内容。如果你想掌握更多的输出格式,请使用sprintf和fprintf功能。

★ 控制命令窗口输入与输出

隐藏输出▶

Getting Started



### 隐藏输出

输入一个语句后按回车键, MATLAB自动在屏幕上显示结果, 但如果该行命令是以"; "结尾, 则MATLAB仍会进行运算赋值, 但不显示任何输出结果. 这一点在生成复杂矩阵时特别有用. 例如:

A = magic(100);

▲ 格式命令

长命令行的输入▶

Getting Started



# 长命令行的输入

如果表达式无法在一行输入完,用三点"···"加回车键表示下一行继续该语句的输入.例如:

$$s = 1 -1/2 + 1/3 -1/4 + 1/5 - 1/6 + 1/7 \dots - 1/8 + 1/9 - 1/10 + 1/11 - 1/12;$$

"=","+"和"-"符号前后的空格可以省略,但使用可以增加易读性,

● 隐藏输出

命令行的编辑▶

Getting Started





### 命令行的编辑

键盘上的多个箭头键和控制键允许撤消,编辑和重新执行之前的命令。例如,如果你错误输入

$$rho = (1 + sqt(5))/2$$

你错误拼写sqrt, MATLAB 会显示

Undefined function or variable 'sqt'. (未定义的函数或变量 'sqt'.)

这时不用重新输入整行的命令,只需按↑键,错误拼写命令会再次显示。 用←键移动光标,并加上漏掉的r。再次使用 ↑ 键召回之前的调用。 输入一串字符再使用 ↑ 键便可找到前面输入的行中以上输入的字符串开头的行. 也可以从命令库中调用之前执行过的命令,详细内容见命令历史

在不同的电脑上可执行命令用不同的编辑键表示. 上机实验看以下哪些键在你的电脑上可行(许多键都类似编辑程序宏指令的编辑)

<b>↑</b>	Ctrl+p	召回上一行
<b>↓</b>	Ctrl+n	召回下一行
←	Ctrl+b	后退一个字符
$\rightarrow$	Ctrl+f	前进一个字符
Ctrl+→	Ctrl+r	右移一个单词
Ctrl+←	Ctrl+1	左移一个单词
Home	Ctrl+a	移至行始
End	Ctrl+e	移至行末
Esc	Ctrl+u	清除该行
Del	Ctrl+d	删除光标所在字符
Backspace	Ctrl+h	删除光标之前字符
	Ctrl+k	删除至行末

● 长命令行的输入

作图▶

**+** 



## 表达式

和其他计算机语言一样,MATLAB也有数学表达式。但与多数计算机语言不同的是,MATLAB的表达式是针对矩阵的。表达式有以下构成:

- 变量
- 数
- 算子
- 函数

● 魔方函数

变量▶

Getting Started



### 算子

表达式使用常用的算术算子和优先规则。

+	加
_	减
*	乘
/	除
\	左除(定义见使用手册的"矩阵和线性代数"一章)
^	乘方
,	复共轭转置
( )	指定计算顺序

◆ 数

函数▶

**Getting Started** 





### 数

MATLAB中的数使用常用的十进制,带有一个可选的小数点和正负号。科学计数法使用字母e指定10的指数。虚数使用i或者j作为虚数单位。以下是一些合法的数的例子:

所有的数将按照IEEE(电气和电子工程师协会)制定的浮点数标准存为长型。浮点数有16位有效小数位的精度,其范围大约是10<sup>-308</sup>到 10<sup>+308</sup>。

● 变量

算子▶

Getting Started



### 函数

MATLAB提供丰富的标准初等数学函数,包括abs(绝对值)、sqrt(平方根)、exp(指数函数)和sin(正弦函数)等。对负数取平方根或对数不会出错;MATLAB将自动产生复数的结果。MATLAB也提供很多高等数学函数,比如Bessel(贝塞耳)函数和Gamma(伽马)函数。这些函数绝大部分支持复变量。要得到一个初等数学函数的列表,可以键入:

help elfun

要得到一个高等数学函数或者矩阵函数的列表,可以键入:

help specfun help elmat

这里面的一些函数像<u>sqrt</u>和<u>sin</u>是内置的,是MATLAB内核的一部分,用起来比较高效,但是不容易看到这些函数的源程序。其他的一些(像<u>gamma</u>、<u>sinh</u>)函数是执行M-文件的。你可以看到这些函数的源程序,如果需要,甚至可以修改这些代码。

### 一些特殊的函数提供一些常用的常数:

<u>pi</u>	3. 14159265
<u>i</u>	虚数单位,√-1
j	和i一样
<u>eps</u>	浮点相对精度,2-52
realmin	最小浮点数, 2-1022
realmax	最大浮点数,(2- <b>s</b> )2 <sup>1023</sup>
<u>Inf</u>	无穷大
NaN	

无穷大是通过非零数除以零或者良定义运算溢出(如:超过realmax)产生的;NaN不定数通常由0/0运算、Inf/Inf等良定义的运算得出。

函数名将不保留。函数名赋予任意新的变量,将会覆盖掉这些函数,例如:

$$eps = 1.e-6$$

执行这个赋值语句后,在下面的运算中就可以使用这个变量(eps的值为1.e-6)了。而原函数可以用下面的语句恢复:

clear eps

### ● 算子

表达式举例▶

Getting Started



### 表达式举例

你已经看过一些MATLAB表达式的例子,下面再举一些例子和运算结果。

#### Getting Started





## 变量

MATLAB中的变量不需要对变量类型和维数的声明。当MATLAB遇到新的变量名时, 会自动生成这个变量并赋予适当的存储单元。如果该变量已经存在,MATLAB将改 变这个变量的值,必要时会赋予新的存储单元。例如:

num students = 25

生成一个1×1的矩阵变量,名为num students,并将25存入到这个矩阵中唯一的 元素。

变量必须以字母开头,后可以跟任意字母、数字或下划线。MATLAB只区别变量名 的前31个字符:MATLAB对字母的大小写敏感,A和a是不同的变量。要查看赋值给 变量的矩阵, 只要键入该变量名即可。

▼ 表达式



Getting Started





### 输入矩阵

从学习利用MATLAB处理矩阵来入手,是入门MATLAB的最好方法!下面就让我们来 看看下面这些例子。

输入矩阵有以下几种不同方法:

- 直接输入组成矩阵的元素。
- 从外部文件引入矩阵。
- 利用自带的函数构成矩阵。
- 利用在\*. M文件内自定义的函数来构成矩阵。

下面让我们来输入杜勒的魔方的一组元素,只要遵循下列几个基本步骤就可以 了:

- 用空格或者逗号来区分一行里不同的元素。
- 用分号': '来区分不同的行。
- 用方括号来括住全体元素。

要输入杜勒的矩阵只要在命令窗口输入:

 $A = \begin{bmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{bmatrix}$ 

按回车后,MATLAB就会显示:

完全符合雕画中的数字! 只要你这样输入过矩阵,MATLAB会自动存储该矩阵在工作台中。简单地用A就可以代表这个矩阵。 现在我们已经有了一个A矩阵在工作台内,让我们来看看它有些什么有趣的地方,为什么它有魔力。

₩ 矩阵操作

求和,转置与对角线▶

Getting Started



### 求和,转置和对角线

可能聪明的你已经发现了魔方矩阵各行或列数字相加后的奇妙属性,无论是各行相加,还是各列相加,还是两条对角线的数字相加,都得到同一个结果。下面让我们用MATLAB来证实这一结论。首先试着输入

sum(A)

MATLAB显示

如果你没有指定变量,MATLAB就会用变量ans(answer的缩写),来储存计算的结果。这里显示的是一个行向量,它的每一个元素是矩阵A相应列的元素的和。很明显,每一列的数字加起来的结果都是同一个数字,34。

如果每一行加起来又会怎样呢?由于MATLAB自带了计算矩阵列元素和的函数,所以最简单的计算行元素和的方法就是求原矩阵的转置的列元素和。一个矩阵的转置可以用在矩阵名后面加一个'''来表示。例如输入

A'

结果显示

```
2 11 7 14
13 8 12 1
```

并且

```
sum(A')'
```

产生一个包含由行元素的和组成的列向量

```
ans = 34 34 34 34
```

对角线元素由diag函数很容易就被列出来

```
diag(A)
```

结果显示:

```
ans = 16 10 7 1
```

如果输入

```
sum(diag(A))
```

结果显示

另外的对角线,即所谓的反对角线,在数学上就不是那么重要的了。所以MATLAB就没有相应的函数。但在图形函数中,函数fliplr产生一个左右调转的矩阵

```
sum(diag(fliplr(A)))
ans =
    34
```

你已经证明了在杜勒的木板图中的矩阵的确是一个魔术方阵,在证明过程中举了一些用MATLAB作矩阵运算的例子。以下的章节将用此矩阵继续介绍MATLAB的其他功能。

### Getting Started





### 下标

记i行j列元素为A(i, j). 例如A(4, 2)就是第4行2列的元素。在魔方阵中就是15。 所以可以算出每行的和

$$A(1,4) + A(2,4) + A(3,4) + A(4,4)$$

结果是

但这不是计算行元素和的最好方法。

实际上用A(k)就可以代表阵中的第k行了,这只用了一个下标。这是经常用来表示行和列向量的方法。也可以应用到两维的矩阵,其中的列是由原来的列一直排下去构成的。对魔方来说,A(8)就是A(4,2)的另一中表达方法,它的值是15。

如果你想用矩阵以外的值,系统就会出错。

$$t = A(4, 5)$$

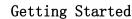
Index exceeds matrix dimensions.

另一方面,如果你输入一个数值到矩阵以外的位置,矩阵的大小就会变化来容纳新的数据。

$$X = A;$$
  
 $X(4, 5) = 17$ 

承 求和,转置与对角线

冒号算子▶







冒号':'是MATLAB里面一个十分重要的算子,适用于各种不同的表达式。例如表达式

1:10

表示一行从1到10的整数

1 2 3 4 5 6 7 8 9 10

为了改变递变的间隔,可以指定一个间隔长度,例如

100:-7:50

表示为

100 93 86 79 72 65 58 51

而

0:pi/4:pi

表示

0 0. 7854 1. 5708 2. 3562 3. 1416

含有冒号的下标表达式也可以用来表示矩阵的一部分

A(1:k, j)

i表示A的第j列前k个元素,因此

sum(A(1:4,4))

表示计算a的第4列元素的和。但是这里有一种更加方便的办法。关键字':'本身可以表示某行或某列的全体元素,而我们可以用关键字end来表示最后一行或者最后一列的全体元素,所以

sum(A(:,end))

是计算a的最后一列元素的和。故有

ans =

为什么一个4×4的魔方矩阵的魔术和是34呢?如果我们从1加到16再把答案除以4,即

答案当然是

▼ 下标

魔方函数▶

Getting Started



### 魔方函数

实际上MATLAB提供了一个函数可以计算出任意规格的魔方矩阵,毫不奇怪,这个函数就是magic.

与刻在杜勒的木板上的那个矩阵差不多一样,具有相同的性质。它们的区别在于中间的两列调换了位置。为了得到木板画中的矩阵A,我们调换B中间的两列

$$A = B(:, [1 \ 3 \ 2 \ 4])$$

上述表达式表示对于矩阵B的每一行,将其列的顺序按照'1,3,2,4'重新编排,即得到

为什么杜勒既然可以用MATLAB提供的排序,他还会在重新调整列向量时遇到麻烦呢?无疑是因为他想把雕刻画的创作年份1514放到魔方的底部的缘故。



# 作图

## 基本作图

MATLAB提供广大的设备用图表来显示向量和矩阵,同时也包括注释和打印这些图表。这部分描述了一些很重要的制图功能且提供一些典型应用的例子。

● 命令行的编辑

创建图▶

Getting Started





## 创建图

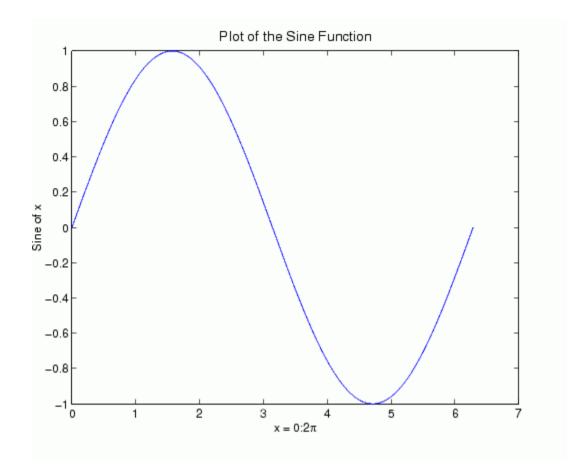
根据输入的不同,plot函数有不同的窗体。如果y是向量的形式,plot(y)则在y对应的轴上作出一个分段线状图。如果指定要求含两个向量时,则plot(x,y)作出一个y相对于x的图表。

例如:下面这些语句了用colon(冒号)算子来创建一个定义值取从0到2 $\pi$ 的向量x,计算出这些值的正弦函数值,然后画出结果。

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x, y)
```

现在给轴加上标签和标题,用\pi作符号π。

```
xlabel('x = 0:2\pi')
ylabel('Sine of x')
title('Plot of the Sine Function', 'FontSize', 12)
```



◆ 作图

一个图表中的多重数据集▶

## Getting Started





## 一个图表中的多重数据集

一个函数作图命令plot使不同的(x-y)变元函数生成不同的函数图象。MATLAB自动 地通过预设地颜色库来区别不同的函数(也可用户自设)。例如,以下是三个x的 相关函数的图象,每条曲线都由各自不同的颜色加以区分。

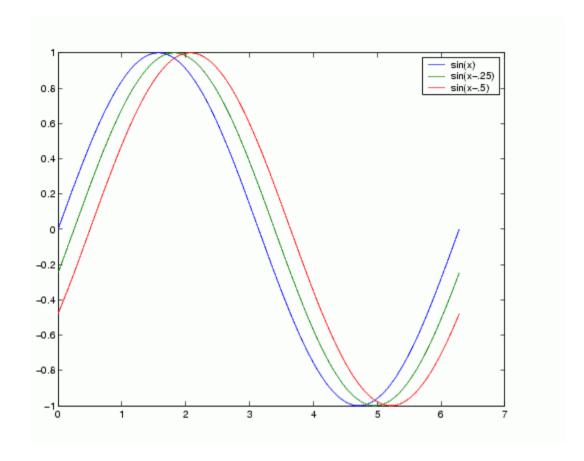
```
y2 = \sin(x-.25);

y3 = \sin(x-.5);

plot(x, y, x, y2, x, y3)
```

legend命令提供一种简易方式来辨别不同的函数作图。

```
legend('\sin(x)', '\sin(x-.25)', '\sin(x-.5)')
```



● 创建图

指定线型与颜色▶

Getting Started



## 指定线型和颜色

在使用plot命令来为数据作图时,你可以自定义颜色,线型和标记(比如加号和圆圈)。

plot(x, y, 'color\_style\_marker')

color\_style\_marker是一个由一到四个字符构成的字符串(用单引号括起来),用以定义颜色,线型和标记形式:

- 颜色字符串有'c', 'm', 'y', 'r', 'g', 'b', 'w', 和'k'。分别表示青, 红紫, 黄, 红, 绿, 白和黑。
- 线型字符串有: '-' 为实线, '--' 为虚线, ':' 为点线, '-.' 为点虚线, 及'none' 表示不用线型。
- 标记形式有'+', 'o', '\*', 和'x', 填入's'代表正方形, 'd'代表菱形, '^'

为上三角形, 'v' 为下三角形, '>' 为右三角形, '<' 为左三角形, 'p' 为五角星形, 'h' 为六角星形, none 为不用标记。

你还可以用交互的方式编辑颜色,线型和标记。详细内容见图形的编辑。

● 一个图表中的多重数据集

作出线与标记▶

Getting Started



•

### 作出线与标记

若指定作图为标记型而非线型,则MATLAB只画出标记,例如:

plot(x, y, 'ks')

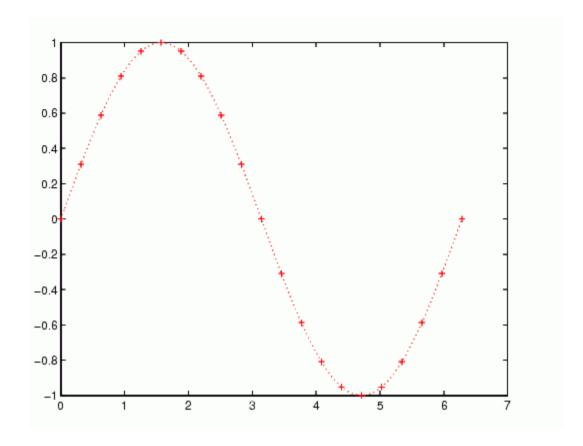
在每个数值点画黑色小正方形,但没有用线连接起来。

而:

plot(x, y, 'r:+')

画出一条红色的点线,在每个数值点上有标记点(+)。你或许想要用更少的数值点来画标记点,下面我们用两组不同数目的点来分别画点线和标记图。例如:

```
x1 = 0:pi/100:2*pi;
x2 = 0:pi/10:2*pi;
plot(x1, sin(x1), 'r:', x2, sin(x2), 'r+')
```



▲ 指定线型与颜色

虚数与复数数据▶

Getting Started



## 虚数与复数数据

当语句是作(plot)复变量时,虚部是忽略的,除非作图给出一个单复变元。对于这种特殊情况,有一捷径的命令来画出实部以及与之对应的虚部。因此,

plot(Z)

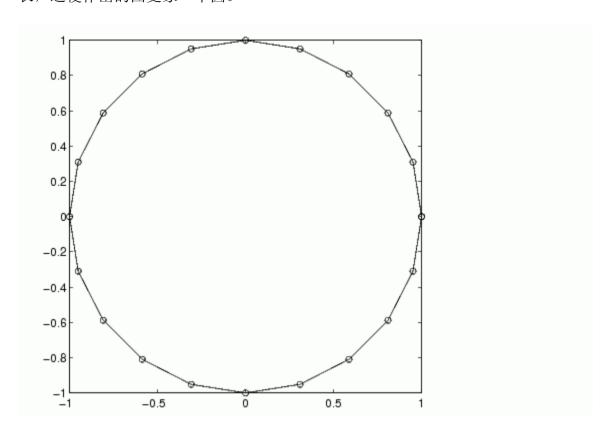
其中z是复向量或矩阵, 其等价于

plot(real(Z), imag(Z))

例如,

```
t = 0:pi/10:2*pi;
plot(exp(i*t),'-o')
axis equal
```

作出一个二十边形并用小圆标记顶点,命令<u>axis</u> equal 使x轴与y轴的单位长度等长,这使作出的图更象一个圆。



◆ 作出线与标记

在现有图中添加图▶

Getting Started





## 在现有图中添加图

hold 命令让你能在现有图中添加图,输入

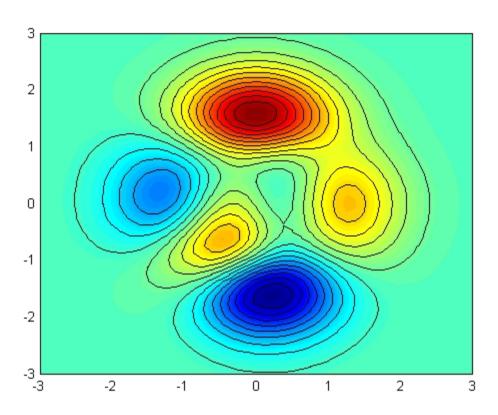
hold on

当你输入另一个作图命令时,MATLAB不会替换现有图;而是把新数据图添加到现有图中,必要时会改变坐标轴的标尺。

例如,以下语句先作一个peaks 函数的轮廓图,并在同一个函数图象中添加伪彩色图。

[x, y, z] = peaks; contour(x, y, z, 20, 'k') hold on
pcolor(x, y, z)
shading interp
hold off

hold on 使命令伪彩色图与轮廓图在同一图象中相结合。



▲ 虚数与复数数据

Figure窗口▶

Getting Started



## Figure窗口

如果屏幕上还没有figure窗口,作图函数会自动打开一个新的figure窗口。如果figure窗口已经存在,MATLAB会用它来输出图象。如果已有多个figure窗口,MATLAB会在指定的"当前"窗口作图。

要使已有的窗口成为当前窗口,可以用鼠标点击该窗口或者输入

figure(n)

其中n是标题栏中的窗口号。结果会显示在该窗口。

要打开一个新窗口并使它成为当前窗口,则输入

figure

▲ 在现有图中添加图

同一Figure中作多幅图▶

Getting Started



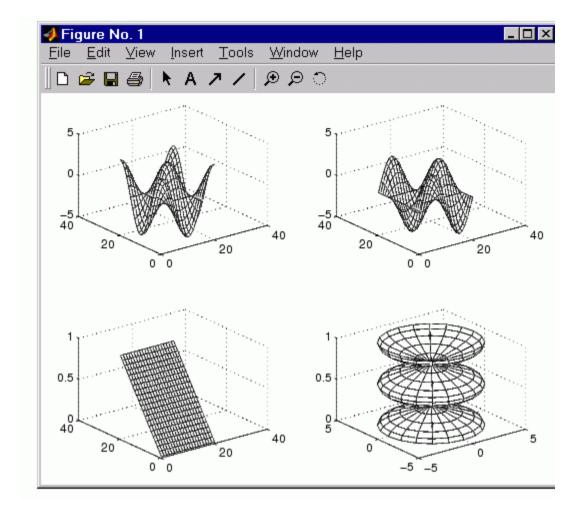
### 同一Figure中作多幅图

用 <u>subplot</u> 命令可以在同一窗口中作多幅图或把它们打印到同一纸上。输入

subplot (m, n, p)

把figure窗口分成m\*n个子区域及选择第p个区域为当前图。所作图是从figure窗口的顶行开始标号,然后第2行,依次类推。例如,以下语句在figure窗口的4个不同子区域分别作图。

t = 0:pi/10:2\*pi; [X, Y, Z] = cylinder(4\*cos(t)); subplot(2, 2, 1); mesh(X) subplot(2, 2, 2); mesh(Y) subplot(2, 2, 3); mesh(Z) subplot(2, 2, 4); mesh(X, Y, Z)



◀ Figure窗口

轴的控制▶

Getting Started



## 轴的控制

axis命令可以规定图象的缩放比例,方位,和纵横比,你可以交互的使用指令进行操作,详见图形的编辑。

### 设置轴的范围

默认时,MATLAB可以根据数值的最大值和最小值决定合适的范围,用<u>axis</u>命令可以自己定义数值的标尺范围:

axis([xmin xmax ymin ymax])

```
三维图则用,
```

axis([xmin xmax ymin ymax zmin zmax])

### 用命令

axis auto

使MATLAB重新自动选择范围。

## 设定纵横比

用axis也可以指定预先确定的数。例如,

axis square

使x轴和y轴等长。

axis equal

使x轴与y轴的单位长度相等。也就是说

plot(exp(i\*[0:pi/10:2\*pi]))

无论后面跟着axis square还是axis equal都把椭圆变成正圆。

axis auto normal

返回默认模式中定义的缩放比例。

### 设定轴的可见性

用axis命令还可以使轴隐藏或显示。

axis on

使轴显示出来。这是默认情况。

axis off

使轴隐藏。

### 设置网格线

grid命令设置网格线显示或隐藏。语句

```
grid on
```

使网格线显示,

grid off

隐藏网格线。

**●** 同一Figure中作多幅图

轴的标签与标题▶

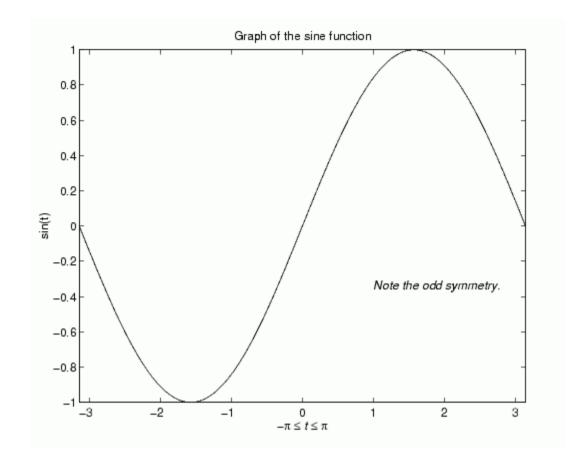
Getting Started



### 轴的标签与标题

用<u>xlabel</u>, <u>ylabel</u>, 及<u>zlabel</u> 命令添加x-, y-, z-等标签。用<u>title</u> 命令在图象顶部加标题,用<u>text</u> 函数在图象中任何部位添加文本。<u>TeX</u>标记的子集则产生希腊字母。可以交互地设置这些选项。详见图形的编辑。

```
t = -pi:pi/100:pi;
y = sin(t);
plot(t, y)
axis([-pi pi -1 1])
xlabel('-\pi \leq {\itt} \leq \pi')
ylabel('sin(t)')
title('Graph of the sine function')
text(1,-1/3,' {\itNote the odd symmetry.}')
```



◀ 轴的控制

Figure的保存▶

Getting Started



## Figure的保存

要保存图形,从File菜单选择Save。要用图形格式如TIFF保存,以便在其他应用中使用,则从File菜单选择Export。还可以从命令行中保存-用<u>saveas</u>命令,包括任何以其他格式保存图象的选项。

▲ 轴的标签与标题

图形的编辑▶

Getting Started





## 图形的编辑

MATLAB在一个图形中提供了可读性,设定轴的比例(包括单位长度),使用颜色

及线型区分图形的格式。然而,如果要创建用于介绍的图形,可能需要改变这些默认格式或添加描述性标签,符号,图例或其他注释以助解释数据。

MATLAB支持两种编辑所建图形的途径。

- 用鼠标交互地选择并编辑对象
- 在命令行或M-文件中用MATLAB函数

## 交互式的作图编辑

若在MATLAB的Figure窗口中打开作图编辑模式,就可以在图形中向对象作鼠标指向和点击的编辑。在这模式里,双击选择要编辑的对象。这就打开了属性编辑器,它提供了可编辑的对象属性,这些属性控制了对象的外观和性能。

关于交互式编辑的更多内容见<u>作图编辑模式的使用</u>。有关在作图编辑模式中编辑对象属性的信息见属性编辑器的使用。

注意 作图编辑模式对访问MATLAB的图形对象属性提供了可选的途径。 然而,在此机制中只能访问对象属性的一个子集。要达到所要求的效果, 可能需要组合使用交互式编辑及命令行编辑。

## 用函数编辑图表

若你更喜欢用MATLAB的命令行工作或在创建M-文件,你可以用MATLAB命令编辑所建图形。使用MATLAB的句柄图形系统,则可以用set或get命令改变图形中的对象属性。关于使用命令行的更多内容见句柄图形.

**◆** Figure的保存

作图编辑模式的使用▶

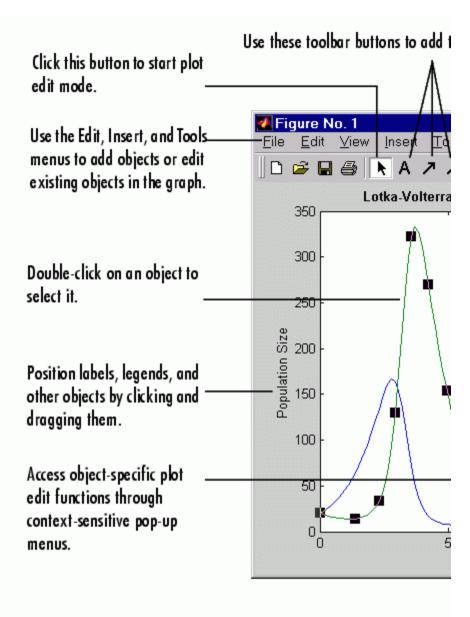
Getting Started





## 作图编辑模式的使用

MATLAB的figure窗口支持一种指向和点击方式的编辑模式,用它可以自定义图形的外观。以下插图说明打开了作图编辑模式的figure窗口及标明了此模式的主要特点。



● 图形的编辑

属性编辑器的使用▶

Getting Started



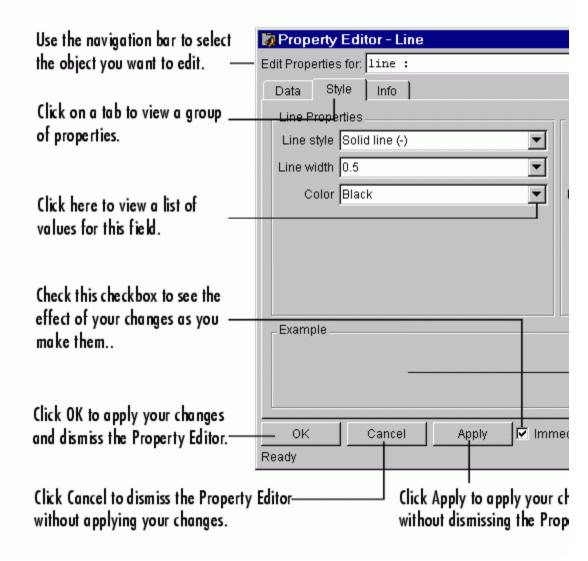
## 属性编辑器的使用

在作图编辑模式里,可以用一个称为模式编辑器的图形用户界面来编辑图形中的对象属性。属性编辑器提供途径以访问下列对象的属性:根,轮廓,轴,线,光,斑点,图像,表面矩形和文本对象。例如,用属性编辑器可以改变线的粗

度,加标题和轴标签,加光照,及执行许多其他作图编辑任务。

这个插图说明属性编辑器界面的成分。

Use these buttons to move back and forth among the graphics objects you have edited



### 启动属性编辑器

要启动属性编辑器,只需双击图形中的某个对象,例如线,或者右键单击对象并 从上下文菜单选择Properties选项。

还可以通过从figure窗口的Edit菜单选择Figure Properties, Axes Properties,或 Current Object Properties来打开属性编辑器。如果作图模式未启动,则用

这些选项可以自动启动它。

一旦启动了属性编辑器,在编辑过程中要保持打开。它提供了对图形中所有对象的访问途径。若单击图形中的其他对象,属性编辑器会展示该对象类型的面板。 也可以用编辑器的导航栏来选择图形中的对象以编辑。

● 作图编辑模式的使用

网格和表面作图 ▶

Getting Started



## 网格和表面作图

MATLAB通过在x-y平面的网格上方的点的z坐标来定义一个表面。mesh和surf作图函数在三维空间显示表面。mesh提供线框表面图,并只对连接定义点的线涂色。surf同时以色彩显示连线和表面外观。

▲ 属性编辑器的使用

双变量函数的可视化▶

Getting Started



## 双变量函数的可视化

要显示以下双变量函数z = f(x, y)的图像:

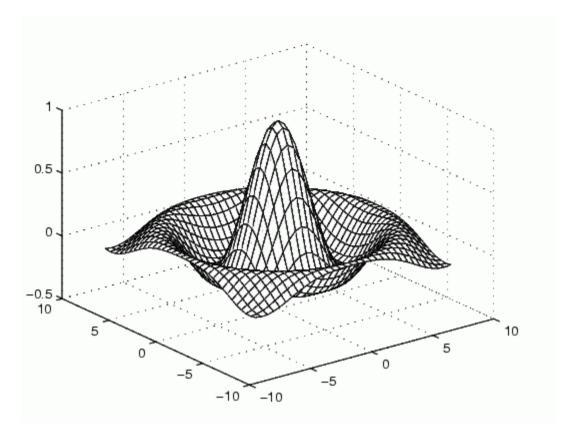
- 分别对函数的定义域生成含有重复行与列的X和Y矩阵。
- 用X和Y来计算函数的图像。

函数 $\underline{\text{meshgrid}}$ 把由单向量或双向量x, y指定的定义域变换成矩阵X和Y,以用于计算双变量函数。X的行和Y的列分别是x和y向量的复制。

### 举例-sinc函数图像

此例计算并作出sinc函数sin(r)/r的在x和y方向间的图象。R是到原点的距离,出现在矩阵中心。添加eps(在系统中返回最小值的MATLAB命令)以避免在原点处出现0/0的不定型。

[X, Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(X, Y, Z, 'EdgeColor', 'black')



在默认的情况下,MATLAB用通用流行的颜色对网孔进行着色 ,不过,这例子只使用单一的颜色着色网孔并且选定了<u>EdgeColor</u> 的表面属性。 查看<u>surface</u>参考页可以看到全部的表面属性。

可以通过把原来的隐藏线显露出来建立一个透明的网孔。

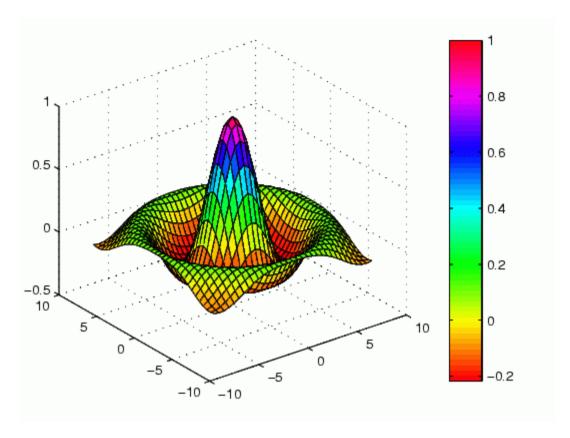
hidden off

查看hidden参考页可以得到更多的关于这个操作的信息。

### 例子—为表面小块着色Example - Colored Surface Plots

表面小块和网孔小块的情形是相似的,除了表面小块的矩形的外观是有色的.它的颜色是由Z和colormap(colormap是一个颜色的顺序列表)的值所确定的.这些陈述用表面图来表示sinc函数的图像,选择一种颜色,然后添加颜色条使表示映射数据的图象得到颜色。

surf(X, Y, Z)
colormap hsv
colorbar



看colormap参考页以得到颜色图的信息。

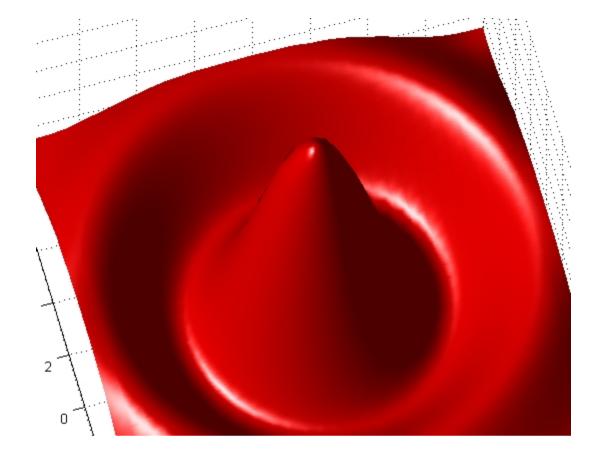
#### 带光照的表面图

光照是一种利用方向光源照亮物体的技术。在某些情况下,这技术能使表面微妙的差异更容易的看到。光照也能用来对三维的图象增加现实感。

这例子使用同前一个例子一样的表面,但是涂上了红色并且移除了网孔线。一个发光的物体被加到了"镜头"的左边(那是你从空间观看那个表面的位置)。

增加光源和设置好照明方式到phong后,使用<u>view</u>命令去改变视角,因而你从空间的另一个不同的点在观看表面(方位角-15和仰角65度). 最后,用工具栏缩放方式放大外观。

```
surf(X, Y, Z, 'FaceColor', 'red', 'EdgeColor', 'none');
camlight left; lighting phong
view(-15,65)
```



▼ 网格和表面作图

图像▶

Getting Started





## 图像

二维的数组能用图象显示,数组元素决定亮度或图像的颜色。例如,语句:

load durer whos		
Name	Size	Bytes Class
X	648x509	2638656 double array
caption	2x28	112 char array
map	128x3	3072 double array

载入文件 durer.mat,加三个变量到工作空间。矩阵X是648\*509矩阵,地图是128\*3矩阵,那是这图像的颜色条。

注意MAT-文件,例如 durer.mat,是二进制文件,它能在一个平台上创建,然后被

### MATLAB 在另一个平台所读取。

X的元素是1到128间的整数,在颜色条中充当指标,地图,那么

image(X)
colormap(map)
axis image

复制杜勒的出现在本书开头的铜版画。右上角的魔方的高解析度扫描能被另一个 文件所利用。输入

load detail

然后使用向上箭头键以重新执行图象,颜色条,和轴命令。语句

colormap(hot)

增加一些20世纪的色化处理到16世纪铜版画。功能 hot 生成一个包含红,橙,黄阴影的颜色条。特别地,一个给定地图像矩阵有一个特定的与之关联的颜色条。另一类预先确定的颜色图请看colormap参考页。

▼ 双变量函数的可视化

图形的打印▶

Getting Started





## 图形的打印

你能直接在一台已连接到你的计算机上的打印机上打印一个MATLAB图形,或输出图形到一种MATLAB所支持的图象文件格式。这里提供两种打印或输出图形的方法。

- 在File菜单下用Print选项
- 用<u>print</u>命令

#### 从菜单打印

在File菜单下方有四个菜单选择是关于打印的。

- Page Setup选项显示一个对话框,它使你能调整打印页图形的特征。
- Print Setup选项显示一个对话框,它设置打印默认值,但是实际上不打印图形。
- Print Preview选项使你能查看到打印出来页面的样子。
- Print选项显示一个对话框,它让你选择标准打印选项然后打印图形。

通常,使用Print Preview确定打印的输出是否你想要的效果。如果不是,使用 Page Setup对话框来改变输出设置。选择Page Setup对话框的Help按钮以显示如 何设置页面的帮助信息。

### 输出图形到图形文件

在文件菜单之下的Export选项使你能输出图形到多种标准图形文件格式。

### 用打印命令

<u>print</u>命令在输出的类型上提供更多的适应性送到打印机,并允许你到从M-文件控制打印。结果能直接送到你的默认打印机或保存到特定文件。多样化的输出格式,包括TIFF, JPEG和PostScript都可用到。

例如,以下语句将当前的图形窗口的内容保存到压缩2级PostScript颜色的文件,名字是magicsquare.eps。这也包括TIFF预览,能使大多数文字处理器显示这图画。

print -depsc2 -tiff magicsquare.eps

要用200dpi的分辨率把图像保存为TIFF文件,用命令

print -dtiff -r200 magicsquare. tiff

若在命令行中输入print,

print

MATLAB就在默认打印机上打印当前图形。

◆ 图像

句柄图形▶

Getting Started





## 句柄图形

当你用作图命令时,MATLAB用多种图形对象创建图形,例如线,文本,表面(完整列表见<u>Graphics Objects</u> )。所有图形对象都有控制其性能的属性。你可以在MATLAB中询问每种属性的取值并设置大部分的属性值。

每当MATLAB创建一个图形对象时,它就给该对象赋以一个标识符(称为句柄)。可以用此句柄访问对象属性。若你想作下列操作,句柄图形很有用。

• 修改图形外观。

• 通过编写能直接创建和操作对象的M-文件以创建自定义作图命令。

▼ 图形的打印

图形对象▶

Getting Started





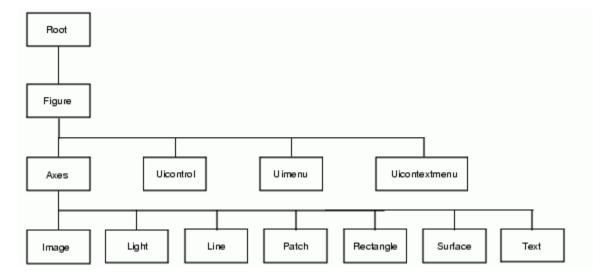
## 图形对象

图形对象是用以显示图形和用户界面元素的基本元素。下表列出了各种图形对象。

对象	描述
Root	对计算机屏幕最高级的对象
Figure	用来显示图形和用户界面的窗口
Axes	在窗口中显示图形的轴
Uicontrol	用户界面控制。执行一个对用户交互作用的函数。
Uimenu	用户定义窗口菜单
Uicontextmenu	右键单击对象时弹出的菜单
Image	二维像素基础图
Light	影响斑点和表面对象着色的光源
Line	用于诸如plot, plot3, semilogx等函数的线
Patch	带边缘的填充多边形
Rectangle	二维形状,从矩形到椭圆
Surface	由x-y平面以上的数据值所作图创建的矩阵数据的三维表示
Text	字符串

### 对象级别

不同对象分布在一个树形结构级别里,反映它们的独立性。例如,线对象是在轴对象的框架下的。依次,轴对象只存在于figure对象中。以下图表阐明了这种树形结构。



#### 创建对象

每个对象都有一个创建用的关联函数。这些函数与所创对象有相同名称。例如,text函数创立文本对象,figure函数创立figure窗口对象,等等。MATLAB的高级别的图形函数(例如plot和surf)会调用适当的低级别函数来作出它们各自的图形。更多关于对象及其属性描述的内容,见对象创建函数的参考页。对象创建函数与对象有相同名称。例如,创立轴的函数是axes.

### 用对象操作的命令

下表列出了操作对象常用的命令。

函数	目的
<u>copyobj</u>	复制图形对象
<u>delete</u>	删除对象
<u>findobj</u>	查找具有指定属性值的对象句柄
gca	返回当前轴的句柄
gcf	返回当前窗口的句柄
gco	返回当前对象的句柄
<u>get</u>	询问对象属性值
<u>set</u>	设置对象属性值

● 句柄图形

设定对象属性▶





## 设定对象属性

所有对象属性都有默认值。然而,改变一些属性设置以定制图形可能有用。有两个途径可以设定对象属性:

- 创建对象时指定属性值。
- 在现有的对象中设置属性值

### 从作图命令设定属性

类似作图函数,在诸如plot, mesh, 和surf的创建对象用的函数也可以指定对象属性值为变元。

例如,可以用创建线或表面的作图命令来指定属性名称和属性值为二元变元。命令

```
plot(x, y, 'LineWidth', 1.5)
```

以x和y为变元作出数据图,用到具有<u>LineWidth</u> 属性且设为1.5点(1点=1/72英寸)的线。可以用此法设定任何线对象。

#### 设定现有对象的属性

要修改现有对象属性值,可以用<u>set</u>命令,如若作图编辑模式已打开,则可用属性编辑器。属性编辑器对许多对象属性提供图形用户界面。本节描述如何用设置命令。详见<u>属性编辑器的使用</u>。

很多作图命令可以返回所建对象的句柄,这样你可以用<u>set</u>命令修改对象。例如,下列语句创立一个5\*5矩阵(创立5行,每行一列),然后设置<u>Marker</u>为正方形并设<u>MarkerFaceColor</u>为绿色。

```
h = plot(magic(5));
set(h, 'Marker', 's', MarkerFaceColor', 'g')
```

在此情形下,h是一个包含5个句柄的向量,每个句柄代表图形中五行之一。语句set设置每一行的Marker和MarkerFaceColor为同一个值。

### 设置多个属性值

若要对每条线的属性设置另一个值,可以用单元数组来储存所有数据并把它传给 set命令。例如,创建一幅图并保存线句柄。

```
h = plot(magic(5));
```

假如要对每条线添加不同的标记并对标记外观填上与线相同的颜色,则须定义两

个单元数组,一个包含属性名,另一个则包含想设的属性值。

单元数组prop\_name包含两个元素。

```
prop_name(1) = {'Marker'};
prop_name(2) = {'MarkerFaceColor'};
```

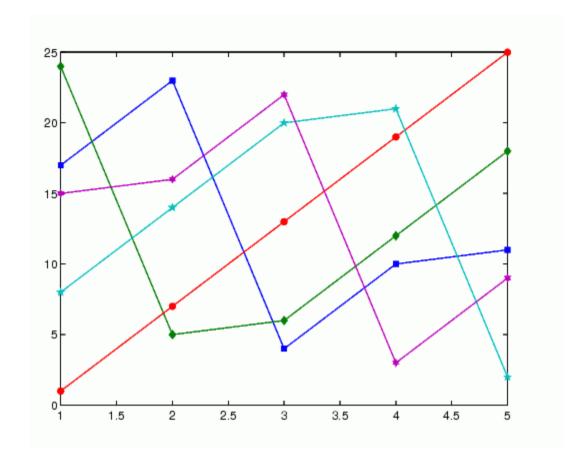
prop\_values 单元数组包含10个值-5个表示表示Marker标记属性,另5个表示MarkerFaceColor属性。注意prop\_values是一个二维单元数组。第一维表示不同值赋给h中的哪个句柄,第二维表示值赋给哪个属性。

```
prop_values(1, 1) = {'s'};
prop_values(1, 2) = {get(h(1), 'Color')};
prop_values(2, 1) = {'d'};
prop_values(2, 2) = {get(h(2), 'Color')};
prop_values(3, 1) = {'o'};
prop_values(3, 2) = {get(h(3), 'Color')};
prop_values(4, 1) = {'p'};
prop_values(4, 2) = {get(h(4), 'Color')};
prop_values(5, 1) = {'h'};
prop_values(5, 2) = {get(h(5), 'Color')};
```

MarkerFaceColor 总是把值赋给对应的线的颜色的(该颜色通过用get命令设置线的Color(颜色)属性得到)。

定义了单元数组后,调用set命令来指定新的属性值。

```
set(h, prop_name, prop_values)
```



● 图形对象

查找现有的对象句柄▶

Getting Started





## 查找现有的对象句柄

用命令<u>findobj</u>可以通过搜索带特定属性值的对象得到图形对象的句柄。用findobj,可以指定任何属性组合的值,这样要从多个对象选出一个就变的容易。例如,可能需要找出带有正方标记蓝色外观的蓝线。

也可以指定搜索哪个图形或轴。以下各节提供示例阐述如何用findobj命令。

### 查找所有某类型的对象

由于所有对象都具有一个用来辨别类型的类型属性,你可以查找所有某特定类型对象事件的句柄。例如,

h = findobj('Type','line');

查找所有线对象的句柄。

## 查找某特定属性的对象

可以指定多个属性以缩窄搜索范围,例如,

```
h = findobj('Type','line','Color','r','LineStyle',':');
```

查找所有红色点线的句柄。

### 限制搜索范围

可以在通过把开始的图形或轴的句柄作为第一变元给对象等级指定开始点,例如,

```
h = findobj(gca,'Type','text','String','\pi/2');
```

只在当前轴下查找字符串π/2。

## 把findobj用作变元

由于findob;返回所找到的句柄, 你可以用它来替换句柄变元, 例如,

```
set(findobj('Type','line','Color','red'),'LineStyle',':')
```

找出所有红线及设定线型为点线。

# ● 设定对象属性

图形用户界面▶

Getting Started





# 图形用户界面

这是一个简单的例子,它说明了如何用句柄图形建立用户界面。语句

```
b = uicontrol('Style', 'pushbutton', ...
    'Units', 'normalized', ...
    'Position', [.5 .5 .2 .1], ...
    'String', 'click here');
```

在figure窗口中心建立一个按钮,并返回一个句柄到新对象。但是,到这为止,单击此按钮无作用。语句

```
s = 'set(b,''Position'', [.8*rand .9*rand .2 .1])';
```

创建一个含有改变按钮位置命令的字符串。

eval(s)

把按钮移动到随机位置。最后,

set(b,'Callback',s)

安置s为按钮的回收动作,故每次点击按钮时,它就移动到一个新位置。

## 图形用户界面的设计工具

MATLAB提供图形用户界面设计环境(GUIDE)工具,它能简化图形用户界面的创作。要显示GUIDE设计编辑器,则发出guide命令。

● 查找现有的对象句柄

动画▶

Getting Started





# 动画

MATLAB提供两种生成生动活泼动画的方法:

- 在屏幕上不断擦除再重画对象,每次重画作增量变化。
- 保存许多不同图, 然后作为电影回放。

● 图形用户界面

删除模式方法▶

Getting Started





# 删除模式方法

用EraseMode属性适合于画面变化最小的简单图的长序列。这里有一个例子放映模拟的布朗运动。指定点的数目,例如

n = 20

以及温度或速度, 例如

s = .02

这两个参数的最佳值决定于你的计算机的速度。生成n个随机点,其(x,y)坐标介乎-1/2和+1/2。

```
x = rand(n, 1) - 0.5;

y = rand(n, 1) - 0.5;
```

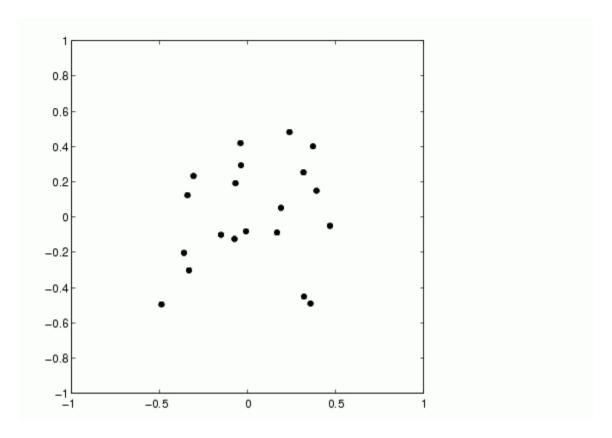
在边界为-1和1的正方形作出各点。保存点的向量的句柄并把它的EraseMode设置为"异或"。这让MATLAB的图形系统在某点的坐标发生变化时不重画全图,而是用"异或"运算把该点的邻域的背景色保存起来。

```
h = plot(x, y, '.');
axis([-1 1 -1 1])
axis square
grid off
set(h, 'EraseMode', 'xor', 'MarkerSize', 18)
```

现在让动画开始。这里有一个无穷的while循环,你可以最后输入Ctrl+c退出循环。每次循环,添加一个小的正态分布的随机噪声到点的坐标。然后,不是重创全新图,只是简单的变化原图的XData和YDataow属性。

```
while 1
    drawnow
    x = x + s*randn(n, 1);
    y = y + s*randn(n, 1);
    set(h, 'XData', x, 'YData', y)
end
```

一个点跑出正方形要多长时间? 所有点逃出方框又要多久?



★ 动画

创作电影▶

Getting Started





# 创作电影

在布朗运动的例子中,若增加点的数目,例如n = 300及s = .02, 运动就不是很流畅了;完成每一步要花太多时间。这时,如果保存预定数目的画面为位图并以电影方式回放则会比较有效。

首先,确定画面数,譬如

nframes = 50;

然后,跟往前一样设定第一幅图,不是用默认的EraseMode (正态).

```
x = rand(n, 1) -0.5;
y = rand(n, 1) -0.5;
h = plot(x, y, '.');
set(h, 'MarkerSize', 18);
axis([-1 1 -1 1])
```

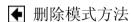
```
axis square grid off
```

生成电影并用getframe捕捉每一画面。

```
for k = 1:nframes
    x = x + s*randn(n, 1);
    y = y + s*randn(n, 1);
    set(h, 'XData', x, 'YData', y)
    M(k) = getframe;
end
```

最后,放映电影30次。

movie(M, 30)



用MATLAB编程 ▶





# 用MATLAB编程

# 流控制

MATLAB有几个流控制结构:

- if语句
- switch语句
- for循环
- while循环
- continue语句
- break语句

● 创作电影

if 🕨

Getting Started





# 脚本与函数

MATLAB是功能强大的程序设计语言和交互计算的环境。在MATLAB语言中,包含代码的文件称为M一文件。用文本编辑器创建M一文件,像使用其它的MATLAB函数或命令一样使用它们。

在MATLAB中有两种M-文件:

- 脚本: 不需要在其中输入参数或返回输出结果。可以在工作空间运行数据。
- 函数: 需要给定输入参数和返回输出结果。函数文件内定义的变量仅在函数文件内部起作用。

如果你是新的MATLAB程序员,只要将M一文件创建到欲试验的当前目录即可。随着你自己的M一文件增多,你可以通过增加MATLAB的搜索路径,把它们放到其他的目录或个人的工具箱。

若复制函数名称,MATLAB会执行在搜索路径中先出现的那个函数。

查看一个M一文件的内容,例如: myfunction.m, 使用 type myfunction

★ 结构体

脚本▶

#### Getting Started





## 脚本

当调用一个脚本时,MATLAB会简单地执行文件中找到的命令。脚本可以运行工作 空间中存在的数据,或者脚本创建新数据来运行。虽然脚本不能返回输出变量, 但是所有创建的变量将保留在工作空间中, 供给后面的计算使用。另外, 脚本能 提供图形输出,就像使用plot函数一样。

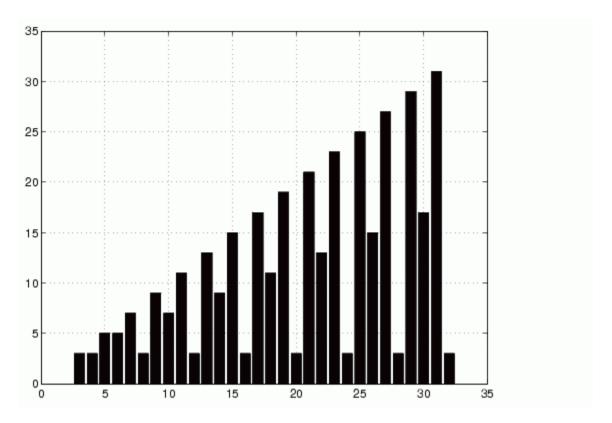
例如, 创建一个叫magicrank.m的文件, 它包含有MATLAB的命令。

```
% Investigate the rank of magic squares
r = zeros(1, 32):
for n = 3:32
  r(n) = rank(magic(n));
end
bar(r)
```

## 输入语句

magicrank

让MATLAB执行这个命令。首先,计算出前30个魔方的秩,然后用柱状图输出结 果。文件的命令执行完毕后,变量n和r保留在工作空间里。



▶ 脚本与函数

函数▶

Getting Started





# 函数

函数是可以定义输入参数或返回输出变量的M一文件。M一文件和函数的名称必须一致。函数只执行自己工作空间内的变量,对于超越工作范围的变量,你要使用 MATLAB命令提示符。

rank是一个很好的例子。rank.m文件可以在toolbox/matlab/matfun目录里获得。

toolbox/matlab/matfun

可用以下命令查看文件

type rank

以下是文件内容。

function r = rank(A, to1)

- % RANK Matrix rank.
- % RANK(A) provides an estimate of the number of linearly
- % independent rows or columns of a matrix A.
- % RANK(A, tol) is the number of singular values of A
- % that are larger than tol.
- % RANK(A) uses the default tol = max(size(A)) \* norm(A) \* eps.

```
s = svd(A);
if nargin==1
   tol = max(size(A)') * max(s) * eps;
end
r = sum(s > tol):
```

函数M-文件的第一行以关键字<u>function</u>开头。它给出函数名和变元的顺序。在此情况下,有最多两个输入变元和一个输出变元。

下面的几行,到第一个空格或可执行,都是提供帮助文本的注释行。输入以下语句则把这些行输出

help rank

帮助文件的第一行是H1行,当你用<u>lookfor</u>命令或在目录请求帮助时MATLAB会显示出。

文件其余部分是定义函数的可执行MATLAB代码。类似第一行的变量r, A和tol, 函数体中引入的变量s对于函数也是局部变量;它们在MATLAB工作空间里独立于任何变量。

此例说明了MATLAB不同于其他程序语言的一个方面——个可变的变元数目。有几种方法可以使用rank函数。

```
rank(A)
r = rank(A)
r = rank(A, 1, e-6)
```

许多M-文件都这样工作。若无输出变元提供,结果保存于ans。若第二个输入变元未提供,则函数计算一个默认值。在函数体内,有两个名为<u>nargin</u>和<u>nargout</u>的数量可用,它们告诉你在某特定应用中输入或输出的变元个数。rank函数用nargin,不用nargout。

₩ 脚本

全局变量▶

Getting Started





# 全局变量

如果你要多于一个函数共用一个简单的变量,简单的处理方法就是把这个变量在

所有函数中定义为global全局变量。在命令行做同样的事情,如果你要工作空间访问上述变量。这个全局变量的定义必须出现在变量被应用于一个函数之前。虽然不是要求,但全局变量也最好以大写字母开头,这样可以同其他变量区别出来。举个例子,做一个以falling,m命名的M一文件。

function h = falling(t)
global GRAVITY
h = 1/2\*GRAVITY\*t.^2;

### 然后交互地输入语句

global GRAVITY
GRAVITY = 32;
y = falling((0:.1:5)');

这两个变量在函数中表示同一个内容。之后你可以交互地修改GRVITY并获得新的解法,而不用再编辑文档。

● 函数

把字符串变元赋给函数▶

Getting Started





# 把字符串变元赋给函数

可以编写认可字符串变元的MATLAB函数,而无须用括号或引号。也就是说, MATLAB把

foo a b c

理解为

foo('a','b','c')

然而, 当用非引用形式时, MATLAB不能返回变元。例如,

legend apples oranges

在一幅图创立一个图例并用字符串apples和oranges作为标签。若想<u>legend</u>命令返回它的输出变元,则必须用引号。

[legh, objh] = legend('apples', 'oranges');

此外,若有任何变元不是字符串时,你都不能用非引号形。

## 用编码建立字符串变元

可以用引号形式在代码中建立字符串变元。以下例子处理多个数据文件 August1. dat, August2. dat, 等等。它用函数<u>int2str</u>,该函数把整数变成字符来建立 文件名。

```
for d = 1:31
    s = ['August' int2str(d) '.dat'];
    load(s)
    % Code to process the contents of the d-th file
end
```

## 一条值得注意的注释

在无引号的句法使用上去颇方便的同时,它也可以在不导致MATLAB产生错误的情况下而被不正确的使用。例如,给出一个矩阵A,

$$\begin{array}{ccccc} A & = & & & & \\ & 0 & -6 & -1 \\ & 6 & 2 & -16 \\ & -5 & 20 & -10 \end{array}$$

用<u>eig</u>返回A的特征值。

```
eig(A)
ans =
-3.0710
-2.4645+17.6008i
-2.4645-17.6008i
```

下面的语句是不允许的,因为A不是字符串,然而MATLAB不会产生错误。

```
eig A
ans = 65
```

MATLAB实际上取出与字母A相等的ASCII码数值的特征值(是65)。

# ◆ 全局变量

eval函数 🕨

Getting Started





## eval函数

eval函数与文本变量一起实现有力的文本宏工具。语句

eval(s)

用MATLAB的注释器求表达式的值或执行包含于文本字符串s的语句。

前一节的例子也可以在下面的代码中实现,虽然这有点低效率因为它用到完全注释器,而不只是函数调用。

```
for d = 1:31
    s = ['load August' int2str(d) '.dat'];
    eval(s)
    % Process the contents of the d-th file
end
```

# ▼ 把字符串变元赋给函数

向量化▶

### Getting Started



# 向量化

要想MATLAB最高速地工作,重要的是在M-文件中把算法向量化。其他程序语言可能用for或DO循环,MATLAB则可用向量或矩阵运算。下例是创立一个算法表。

```
x = .01;
for k = 1:1001
y(k) = log10(x);
x = x + .01;
```

同样代码地向量化翻译是

```
x = .01:.01:10;

y = log10(x);
```

对于更复杂的代码,矩阵化选项不总是那么明显的。当速度重要时,应该想法把算法向量化。

# ● eval函数

预分配▶

Getting Started





## 预分配

若一条代码不能向量化,你可以通过预分配任何输出结果已保存其中的向量或数组以加快for循环。例如,这个代码用zeros函数把for循环产生的向量预分配。

这使得for循环的执行显著加快。

```
r = zeros(32, 1);
for n = 1:32
    r(n) = rank(magic(n));
end
```

上例中若没有使用预分配,MATLAB的注释器利用每次循环扩大r向量。向量预分配排除了该步骤使执行加快。

● 向量化

函数的句柄▶

Getting Started





## 函数的句柄

你可以创建任何MATLAB函数的句柄,然后用这些句柄作为函数链接的途径。函数句柄主要用来传递自变量列表给其他函数,用句柄执行函数或求值。

在MATLAB中,通过在函数名前加上符号@构造函数句柄。下面例子为SIN函数创建一个函数句柄,然后赋值给变量fhandle:

fhandle = @sin;

用MATLAB的fevel函数对函数句柄求值。下面的plot\_fhandle函数接收一个函数句柄和数据,然后用feval函数对已经给定初使数据的函数句柄求值:

```
function x = plot_fhandle(fhandle, data)
plot(data, feval(fhandle, data))
```

当你用SIN函数句柄调用plot\_fhandle函数和后面的变量时,求值结果是正弦波形图象。

plot fhandle(@sin, -pi:0.01:pi)

● 预分配

函数的函数▶

Getting Started





# 函数的函数

一种以标量变量的非线性函数为变量的函数称为"函数的函数",即以函数名为自变量的函数。这类函数包括:

- 求零点
- 最优化
- 求积分
- 常微分方程

MATLAB通过M文件的函数表示该非线性函数。例如,下面是一个简化的humps函数,来源于matlab/demos路径。

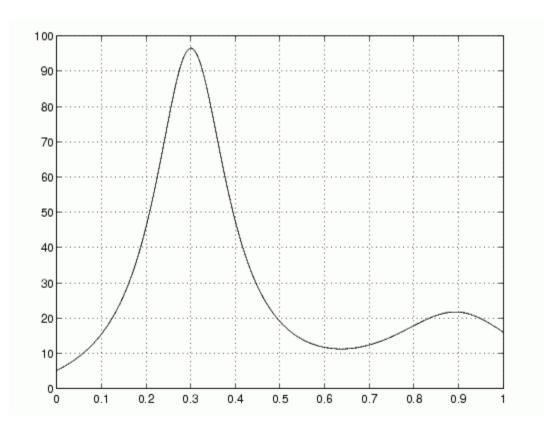
function 
$$y = humps(x)$$
  
 $y = 1./((x-.3).^2 + .01) + 1./((x-.9).^2 + .04) - 6;$ 

## 在区间0 ≤ x ≤ 1求此函数的值

x = 0:.002:1;y = humps(x);

#### 然后作出图象

plot(x, y)



以上图象表明函数在x=0.6附近有局部最小值。用函数fminsearch可以求出局部最小值,以及此时x的值。函数fminsearch第一个参数是(需要求出局部最小值的函数的)函数句柄,第二个是参数此时x的近似值。

```
p = fminsearch(@humps,.5)
p =
     0.6370
```

要求出此局部最小值,则

humps(p)
ans =
11.2528

数值分析家用求积分或积分法来区分定积分的近似值和常微分方程的数值积分。 MATLAB的积分程序是guad和guadl。语句

Q = quad1(@humps, 0, 1)

计算图象曲线下方的面积得

Q = 29.8583

最后, 图象说明函数在这区间无零点。故若用下面语句找零点

z = fzero(@humps, .5)

只能找到区间外的零点。

z = -0. 1316

● 函数的句柄

包含于MATLAB的程序示范▶

Getting Started





## if

if语句计算逻辑表达式并当表达式为真时执行一组语句。可选的elseif和else关键字规定后面的各组语句是否执行。与if搭配的关键字end结束最后一组语句。所有语句都用这4个关键字描绘,且不必用括号。

MATLAB的生成n阶魔方的算法有三种不同情形: n是奇数, n为偶数但不能被4整除, n能被4整除。用下列语句描述。

if 
$$rem(n, 2) \sim 0$$
  
 $M = odd magic(n)$ 

```
elseif rem(n, 4) \sim = 0
   M = single even magic(n)
else.
  M = double even magic(n)
end
```

在此例中,三种情形是互斥的,否则,就执行第一个真条件。

重要的是,要明白有关的算子和if语句是如何对矩阵起作用的。当你想检查两个 变量是否相等, 你或许会用

```
if A == B, ...
```

这是合法的MATLAB代码,当A,B都是标量时执行如你所愿的任务。但当A,B是矩阵 时,则A==B并不是测试它们是否相等,而是在哪些位置相等:这样,结果是另一 个含有0和1的矩阵,表示元素与元素是否相等。事实上,若A和B不是同尺寸的矩 阵,则A==B是错误语句。

检查两变量的等同性的正确方法是用isequal函数,

```
if isequal(A, B), ...
```

这里有另一个强调这点的例子。若A,B是标量,下面程序一定不会出现意料之外的 情形。但对于大多数矩阵对,包括有相互交换列的魔方,对所有元素,无论A>B. A < B还是A == B都非真, 因此执行了else子句。

```
if A > B
   'greater'
elseif A < B
   'less'
elseif A == B
  'equal'
   error ('Unexpected situation')
end
```

有些函数对把矩阵比较简化为标量条件以用if很有用。包括

isequal isempty all anv

**■** 用MATLAB编程

switch和case •



### switch和case

<u>switch</u>语句执行基于变量或表达式值的语句组。关键字<u>case</u>和<u>otherwise</u>描述语句组。 只执行第一个匹配的情形。用到switch则必须用end与之搭配。

魔方算法的逻辑也可以如下描述

```
switch (rem(n, 4) == 0) + (rem(n, 2) == 0)
    case 0
     M = odd_magic(n)
    case 1
     M = single_even_magic(n)
    case 2
     M = double_even_magic(n)
    otherwise
     error('This is impossible')
end
```

注意 与C语言的switch语句不同的是,MATLAB的switch是不会落空的。如果情形1是真,则其他语句不执行。因此,不需用break语句。

🚺 if

for 🕨

### Getting Started





#### for

for循环重复执行一组语句一个预先给定的次数。匹配的end描述该语句。

```
for n = 3:32
    r(n) = rank(magic(n));
end
r
```

内部语句末尾的分号隐藏重复的打印,循环后的r显示最终结果。

为可读性编排循环是个好主意,特别是当它们是嵌套时。

```
for i = 1:m
  for j = 1:n
    H(i, j) = 1/(i+j);
  end
end
```

## Getting Started





### while

while循环在一个逻辑条件的控制下重复一组语句一个不定的次数。匹配的end描述该语句。

这里有一个完整的程序,阐明了while, if, else, 和end, 这程序是用区间二分法寻找一个多项式的零点。

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
x
```

结果是多项式 $x^3 - 2x - 5$ 的根,即

```
x = 2. 09455148154233
```

在if语句一节提到的有关矩阵比较的注意事项,对while语句也适用。

for

continue 🗪



## Getting Started





### continue

continue 语句把控制传给下一个在其中出现的if或while循环的迭代,忽略任何循环体中保留的语句。在嵌套循环中,continue把控制传给下一个for或while循环所嵌套的迭代。

下面的例子展示了一个在magic.m文件代码中的计算行数的continue循环,跳过所有空行和注释。continue语句用于前进到magic.m的下一行而不用在每次遇到空行或注释行时都增大。

```
fid = fopen('magic.m','r');
count = 0;
while ~feof(fid)
    line = fgetl(fid);
    if isempty(line) | strncmp(line,'%',1)
        continue
    end
    count = count + 1;
end
disp(sprintf('%d lines',count));
```

**♦** while

break 🗪

#### Getting Started



# break

<u>break</u>语句让你提前从for或while语句中离开。在嵌套循环里,break只从最里面的循环离开。

这是上节例子的改进。为何用break是好主意呢?

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if fx == 0
        break
    elseif sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
x
```

**★** continue

其他数据结构▶

Getting Started





# 其他数据结构

本节介绍MATLAB中的一些其他数据结构,包括:

- 多维数组
- 单元数组
- 字符与文本
- 结构体

**♦** break

多维数组 ▶

Getting Started



## 多维数组

MATLAB中的多元数组MATLAB是下标多于两个的数组。可以调用带有两个以上变元的zeros, ones, rand, 或randn函数来产生。例如,

$$R = randn(3, 4, 5);$$

生成一个3\*4\*5的数组,共有3x4x5 = 60个正态分布的随机元素。

三维数组可能表示三维物理数据,譬如房间的温度,取样于一个长方形网格。或者,可表示一个矩阵序列 $A^{(k)}$ ,或依赖时间变化的矩阵A(t)。在后面这些情形中,第k个矩阵的第(i, j)个元素用A(i, j, k)表示。

4阶魔方的MATLAB和杜勒的版本不同在于交换了两列。许多魔方可以通过交换列产生。语句

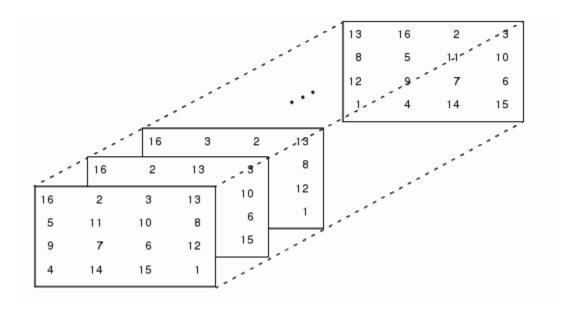
```
p = perms(1:4);
```

生成4! = 24个1:4的排列。第k个排列是行矩阵p(k,:),那么

```
\begin{array}{l} A = magic(4); \\ M = zeros(4,4,24); \\ for \ k = 1:24 \\ M(:,:,k) = A(:,p(k,:)); \\ end \end{array}
```

在三维数组M中保存有24项的魔方序列。M的大小为

```
size(M)
ans =
4 4 24
```



生成的序列中第三个是杜勒的。

语句

sum(M, d)

通过改变第d个下标计算和。因此

sum(M, 1)

是一个1\*4\*24的数组,包含24个相同的以下行向量

34 34 34 34

而

sum (M, 2)

是一个4\*1\*24的数组,含有24个相同的以下列向量。

34

34

34 34

最后.

$$S = sum(M, 3)$$

把序列中的24个矩阵相加。结果的尺寸是4\*4\*1,故看起来象4\*4数组。

## ▲ 其他数据结构

单元数组▶

Getting Started





# 单元数组

MATLAB中的单元数组是多维数组,其元素是其他数组的副本。空矩阵的单元数组 可以用cell函数生成。但是更多时候,单元数组是通过用大括号引把各种东西括起 来产生的。大括号也用下标来访问不同单元的内容,例如,

$$C = \{A \text{ sum}(A) \text{ prod}(prod(A))\}$$

产生一个1\*3的单元数组。三个单元分别包含魔方,列和的行矩阵,及所有元素的 积。若把C写出来,就是

$$C = [4x4 \text{ double}] [1x4 \text{ double}] [20922789888000]$$

这是因为,前两个单元太大以至不能在此有限空间输出,但第三个元素只包含一 个数16!,因而有位置输出。

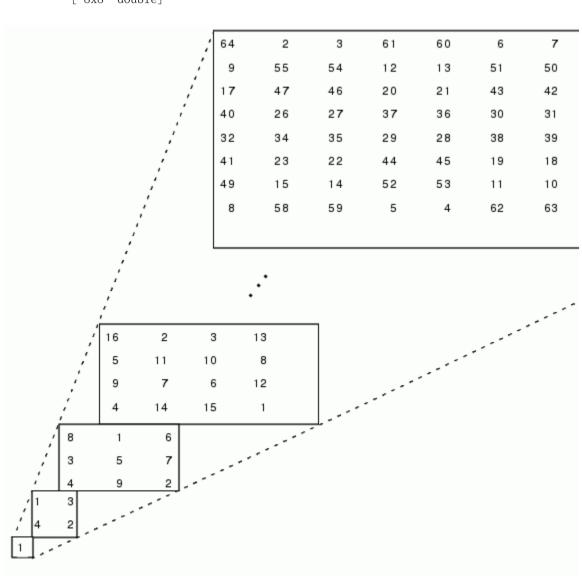
这里有两点要记住。第一,要找回其中一个单元的内容,就用大括号中的下标。 例如, C{1}返回魔方, C{3}则是16!。第二,单元数组包含其他数组的副本,而不是 指向这些数组的指针。若你随后改变A.则C不变。

三维数组可以用来保存一个同尺寸的矩阵序列。单元数组可以用来保存不同尺寸 矩阵的序列。例如,

$$M = cell(8, 1);$$
  
for  $n = 1:8$ 

```
M\{n\} = magic(n); end M
```

## 生成一个不同阶的魔方序列。



用下面语句可以找回我们的老朋友。

 $M\{4\}$ 

## ● 多维数组

字符与文本▶

Getting Started



## 字符与文本

用单引号输入文本至MATLAB。例如,

s = 'Hello'

结果不是我们迄今为止处理过的数字矩阵或数组。它是一个1\*5的字符数组。

在内部, 字符是以数字存储的, 而不是用浮点格式。语句

a = double(s)

把字符数组转换为包含每个字符ASCII码的浮点表示的数字矩阵。

a = 72 101 108 108 111

语句

s = char(a)

把变化逆转。

把数字变成字符使你能在计算机上调查不同字体。在ASCII字符集中可打印的字符是由32:127的整数表示的。(小于32的整数表示不可打印的控制字符。)这些整数编排在一个恰当的6\*16数组里。

F = reshape(32:127, 16, 6)';

在扩展ASCII码中的可打印字符用F+128表示。当这些整数翻译成字符时,结果决定于当前所用字形。输入语句

char (F) char (F+128)

然后改变用于MATLAB命令窗口的字形。从File菜单选择Preferences。一定要尝试

Symbol和Wingdings字体,若你的计算机中存在该字体。这是可能得到的某种输出的例子。

```
!"#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmno
pqrstuvwxyz{|}^\x7f

°¢≤§•¶ß®©™'"≠ÆØ
\xb0±≤≥¥μ¹²³π½²°¾æØ
¿;¬√Ýý«»...þÄÃÕ
---""`;÷Þÿ\xd9\xda X\xde\xdf
•ÂÊÁËÈÍÎÏÌÓÔ
ÒÚÛ\xf5~\xf9\xfa°,\xfd\xfe\xff
```

用方括号把文本变量连接成更长的字符串。语句

$$h = [s, 'world']$$

把字符串水平地连接起来并产生

语句

$$v = [s; 'world']$$

把字符串另起一行连接并产生

```
v =
Hello
world
```

注意,在h里,字母'w'前必须插入空格,在v中,两个词必须等长。得到的两个数组都是字符数组;h是1\*11的,v是2\*5的。

要改变包含不同长度行的文本体,有两个选择一填补字符的数组或字符串地单元数组。<u>char</u>函数允许任何长度的语句,添加空格到每行以使它们具有相同长度,生成一个独立成行的字符串数组。例如,

```
S = char('A', 'rolling', 'stone', 'gathers', 'momentum.')
```

生成一个5\*9的字符数组。

```
S = A rolling stone gathers momentum.
```

S的前四行里,每行都有足够的空格使得各行等长。另一种选择是,把文本保存到单元数组中。例如,

```
C = {'A'; rolling'; stone'; gathers'; momentum.'}
```

是一个5\*1的单元数组。

用下面语句可以把一个填补字符数组转换成字符串单元数组

```
C = cellstr(S)
```

用下面语句把过程逆转

```
S = char(C)
```

# ● 单元数组

Getting Started

1 十九级组



结构体▶



# 结构体

结构体是MATLAB的多维数组,其元素取决于原文的域指示符。例如,

```
S. name = 'Ed Plum';
S. score = 83:
```

S. grade = 
$$'B+'$$

创建一个含三个域的标量结构体。

grade: 'B+'

和MATLAB的其他一切一样,结构体是数组,故可以插入附加元素。在此情况下,数组的每个元素都是带有若干域的结构体。域可以逐个加起来,

```
S(2).name = 'Toni Miller';
S(2).score = 91;
```

S(2).grade = 'A-';

或者, 完整元素可以由单个语句相加得到。

现在,结构体足够大,可以输出摘要了。

```
S =
1x3 struct array with fields:
   name
   score
   grade
```

有几种方法可以把不同域集合到其他MATLAB数组里。它们都是基于逗号分开的列表的符号。若输入

S. score

## 等于输入

S(1).score, S(2).score, S(3).score

这就是逗号分开的列表。没有任何其他标点,它不是很有用。这样,它把三个分数值依次赋给默认变量ans,并正确无误地输出每个赋值结果。但当你用方括号把表达式括起来时,

[S. score]

## 就等同于

[S(1).score, S(2).score, S(3).score]

生成一个包含所有分数的行向量。

类似地,输入

S. name

只依次把姓名赋值给ans。但若嵌入花括号,

{S. name}

则产生一个包含三个名字的1\*3单元数组,

ans =
'Ed Plum' 'Toni Miller' 'Jerry Garcia'

而

char(S. name)

调用char函数,用三个变元从姓名域创立一个字符数组,

ans =
Ed Plum
Toni Miller
Jerry Garcia

# ● 字符与文本

脚本与函数▶

Getting Started



# 包含于MATLAB的程序示范

MATLAB包含许多突出不同特征和功能的程序示例。要查看示例列表,在命令提升 符输入

help demos

要看详细文件,例如airfoil,键入

edit airfoil

要运行示例,则在命令提示符输入文件名。例如,运行螺旋桨示例,输入

airfoil

注意: 许多示例用到多窗口及需要你在MATLAB命令窗口按键继续运行。

下表列出了一些可用的现成的示例,它们编排在以下类别里:

- 矩阵示例
- 数字示例
- 可视化示例语言示例
- 常微分方程组示例
- 图库示例
- 游戏示例
- 其他示例
- 帮助函数示例

## MATLAB矩阵示例

例。
0

## MATLAB数字示例

bench	MATLAB基准。
census	美国2000年人口预报。
e2pi	如下问题的二维直观解答 "e <sup>π</sup> 和π <sup>e</sup> 哪个较大?"
fftdemo	光谱分析的快速傅立叶变换法使用。
fitdemo	用单一算法作非线性曲线拟合。
fplotdemo	函数作图的示例。

funfuns	函数嵌套示例。
lotkademo	常微分方程解的示例。
quaddemo	适应性积分。
quake	Loma Prieta震荡。
spline2d	二维ginput和spline的示例。
sunspots	MATLAB中快速傅立叶变换示例,用于分析太阳黑子活动性的变化。
zerodemo	用fzero找零点。

# MATLAB可视化程序示例

colormenu	在当前图象添加颜色条示例。
cplxdemo	单复变函数映像。
earthmap	地球地形的图形演示。
graf2d	MATLAB中的二维坐标图
graf2d2	MATLAB中三维坐标图。
grafcplx	MATLAB中复函数图象。
imagedemo	MATLAB的图形功能演示。
imageext	变换与旋转图像颜色图演示。
lorenz	洛仑兹混沌吸引子附近轨道的图象演示。
penny	便士数据几种图。
vibes	震荡L形膜电影。
xfourier	傅立叶级数展开的图形演示。
xpklein	克莱因瓶演示。
xpsound	MATLAB声音功能演示。

# MATLAB语言程序示例

graf3d	表面图的句柄图演示。
hndlaxis	轴的句柄图演示。
hndlgraf	线图的句柄图演示。

# MATLAB常微分方程组程序演示

a2ode	带实特征值的线性刚性问题。
a3ode	带实特征值的线性刚性问题。
b5ode	带复特征值的线性刚性问题。
ballode	用于BALLDEMO中的弹球运动方程。
besslode	用于BESSLDEMO的零阶Bessel方程。
brussode	模拟化学反应(Brusselator)的刚性问题。
buiode	Bui的刚性问题的分析解。
chm6ode	从Enright到Hull的化学加工6的刚性问题。
chm7ode	从 Enright到Hull的化学加工7的刚性问题。
chm9ode	从 Enright到Hull的化学加工9的刚性问题。
d1ode	带非线性实特征值的刚性问题。
fem1ode	带有依赖于时间的块矩阵的刚性问题。
fem2ode	带有不依赖于时间的块矩阵的刚性问题。
gearode	由van der Houwen引用的归于Gear的刚性问题。
hb1ode	Hindmarsh和Byrne的刚性问题1。
hb2ode	Hindmarsh和Byrne的刚性问题2。
hb3ode	Hindmarsh和Byrne的刚性问题3。
odedemo	常微分方程组积分器的示例。
orbitode	用于ORBITDEMO的限制三物体问题。
orbt2ode	Hull et al的非刚性问题D5。
rigidode	不受外力的刚体的欧拉方程。
sticode	用于STICDEMO的固定于表面的弹簧驱动质量。
vdpode	可参数化的van der Pol方程(对大μ刚性)。

# MATLAB图库程序示例

cruller	麻花炸面圈的图形演示。

_	
klein1	克莱因瓶的图形演示。
knot	包围三维绳结的管。
logo	MATLAB的L形膜标识语的图形示例。
modes	L形膜的12种模式的图形演示。
quivdemo	震动方程的图形演示。
spharm2	球面谐波的图形演示。
tori4	四连接,未成结的圆环面的图形演示。
finddemo	查找单独工具箱的可用示例的命令。
helpfun	方便地显示帮助文本的效用函数。
membrane	MathWorks标识语。
peaks	双变量的样本函数。
pltmat	在figure窗口显示矩阵的命令。

# MATLAB游戏演示程序

bblwrap	泡泡游戏。
life	Conway的生命游戏。
soma	彩色魔方块。
xpbombs	扫雷游戏。

# MATLAB其他演示程序

codec	字母变换编码器/解码器。
crulspin	扭纹圈自转电影。
logospin	MathWorks标识语的自转电影。
makevase	表面旋转的演示。
quatdemo	四元数旋转。
spinner	彩色线在空间自转。
travel	"货郎担"问题。
truss	弯桥构架动画。
wrldtrv	绕地球大圆的航空线。

xphide	运动物体的可视感知。
xpquad	二次超曲面的作图演示。

## MATLAB帮助函数示例

bucky	Buckminster Fuller的球顶网格图。
cmdlnbgn	命令行示例的设定。
cmdlnend	命令行示例演示后清除。
cmdlnwin	运行命令行演示的通行路径。
finddemo	对单独工具箱查找可用示例的命令。
helpfun	方便地显示帮助文本的效用函数。
membrane	MathWorks标识符。
peaks	双变量样本函数。
pltmat	在figure窗口显示矩阵的命令。

## 更多信息

MathWorks的网站(www.mathworks.com)包含众多的由用户和MathWorks员工编写的M-文件。可以选择下载以使用。另外,从技术支持网站(www.mathworks.com/support)可访问到的Technical Notes包含大量的示例,如作图,数学,应用编程接口,Simulink等等。

┫函数的函数