第一篇
MATLAB程序设计基础

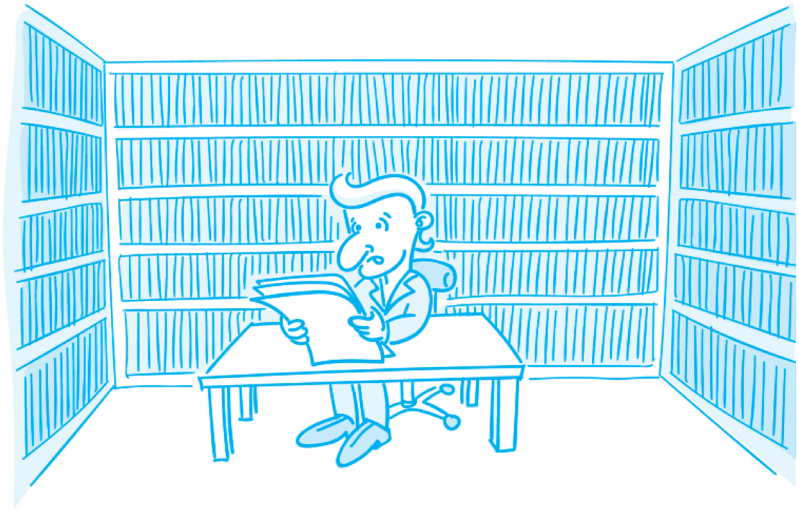
本篇主要介绍MATLAB程序设计的基本概念，并以MATLAB R2023a版本为平台，介绍MATLAB程序设计的基础知识。MATLAB程序设计的基本内容包括程序中的基本数据类型、变量及其基本运算、绘图、典型应用程序的基本结构，以及函数(子程序)的创建和使用方法。本篇将对这些问题进行详细介绍，具体包括如下章节。

第1章　MATLAB的基本应用

第2章　矩阵和矩阵运算

第3章　图形绘制与数据可视化

第4章　MATLAB程序设计



第1章 MATLAB的基本应用

第1集
微课视频

本章在MATLAB相关概念的基础上，重点介绍MATLAB集成窗口环境的基本使用方法，主要内容如下。

（1）MATLAB语言的基本知识，了解MATLAB基本界面操作。

（2）了解MATLAB命令行的基本操作。

（3）熟悉脚本命令和程序的基本概念，掌握创建、编辑、保存和运行脚本程序的基本方法。

1.1 MATLAB入门

MATLAB的名字取自矩阵实验室（Matrix Laboratory），是美国MathWorks公司出品的商业科学计算软件。MATLAB是主要面对科学计算、可视化以及交互式程序设计计算环境。MATLAB包括MATLAB和Simulink两大部分，在数据分析、无线通信、深度学习、图像处理与计算机视觉、信号处理、量化金融与风险管理、机器人、控制系统等领域有广泛应用。MATLAB图标如图1-1所示。

图1-1 MATLAB图标

借助MATLAB这一高级编程语言，用户能够直接进行矩阵和数组运算，进行科学计算、数据分析、算法开发、模型创建和可视化等工作，为科学研究、工程设

计及必须进行有效数值计算的众多科学领域提供一种全面的解决方案，在很大程度上摆脱了传统非交互式程序设计语言的编辑模式。相比于传统的编程语言，MATLAB有诸多的优点，主要包括如下几方面。

(1) 易用性。MATLAB是一种解释型语言，简单易用，可直接在命令行窗口输入命令行求解表达式的值，也可执行预先编写好的大型程序。在MATLAB集成开发环境下，可以方便地编写、修改和调试程序。

(2) 平台独立性。MATLAB支持许多操作系统，提供了大量的平台独立的措施。在某个平台上编写的程序，在其他平台上同样可以正常运行；在某个平台上编写的数据文件也可以在其他平台上编译。因此，用户可以根据需要把MATLAB编写的程序移植到新平台。

(3) 预定义函数和工具箱。MATLAB具有强大的预定义函数库，它提供了许多已测试和打包过的基本工程问题的函数，不需要重复工作，让编程更简单。同时，它还提供了多种经过专业开发、严格测试并拥有完善的帮助文档的工具箱，涵盖了众多领域，极大地方便了使用者。

(4) 强大的绘图功能。MATLAB提供了一系列的绘图函数，用户不需要过多地考虑绘图的细节，只需要提供一些基本参数就能得到所需图形，这类函数称为高层绘图函数。此外，MATLAB还提供了直接对图形句柄进行操作的低层绘图操作，这类操作将图形的每个图形元素(如坐标轴、曲线、文字等)作为一个独立的对象，系统给每个对象分配一个句柄，可以通过句柄对该图形元素进行操作，而不影响其他部分。

(5) 交互式应用程序。MATLAB程序可看到不同的算法如何处理数据。在获得所需结果之前反复迭代，自动生成MATLAB程序，以便对工作进行重现或自动处理，设计出便于操作的、功能复杂的数据分析程序。

(6) 扩展能力。只需更改少量代码，就能将MATLAB程序扩展到群集、图形处理单元(Graphics Processing Unit，GPU)和云上运行，无须重写代码或学习大数据编程和内存溢出技术。支持将MATLAB程序部署到云平台和企业系统，并能与数据源和业务系统集成。

(7) 支持在嵌入式设备中运行。MATLAB代码可以自动转换为C/C++和HDL代码，从而在嵌入式设备中运行。MATLAB与Simulink配合以支持基于模型的设计，用于多域仿真、自动生成代码，以及嵌入式系统的测试和验证，成为基于模型的设计(Model-based Design，MBD)方法。

想要获取正版的MATLAB，需要在MathWorks公司官网注册成为用户，先下载MATLAB试用版软件，后续根据进一步需求可以进行购买。安装好MATLAB软件后，双击桌面图标，可以打开如图1-2所示的默认布局界面。桌面主要包括下列区域。

当前文件夹：当前访问的文件路径中存储文件。

命令行窗口：用户可以直接在命令行中输入命令(命令行由提示符“>>”开始)。

工作区：显示用户创建或从文件导入的数据。

MATLAB界面的布局形式可以根据个人编程习惯，单击“布局”按钮进行修改，如图1-2所示。

以上是安装MALTAB软件后，显示给用户的界面基本情况，用户可在MATLAB命令行窗口中输入语句，实现相应的功能。

MATLAB集成了很多工具箱，不同版本的MATLAB工具箱更新程度不同，若要查询工具箱种类及工具箱版本，可直接在MATLAB命令行窗口输入ver命令，按Enter键即可得到MATLAB版本信息，如图1-3所示。

图 1-2　MATLAB 启动界面

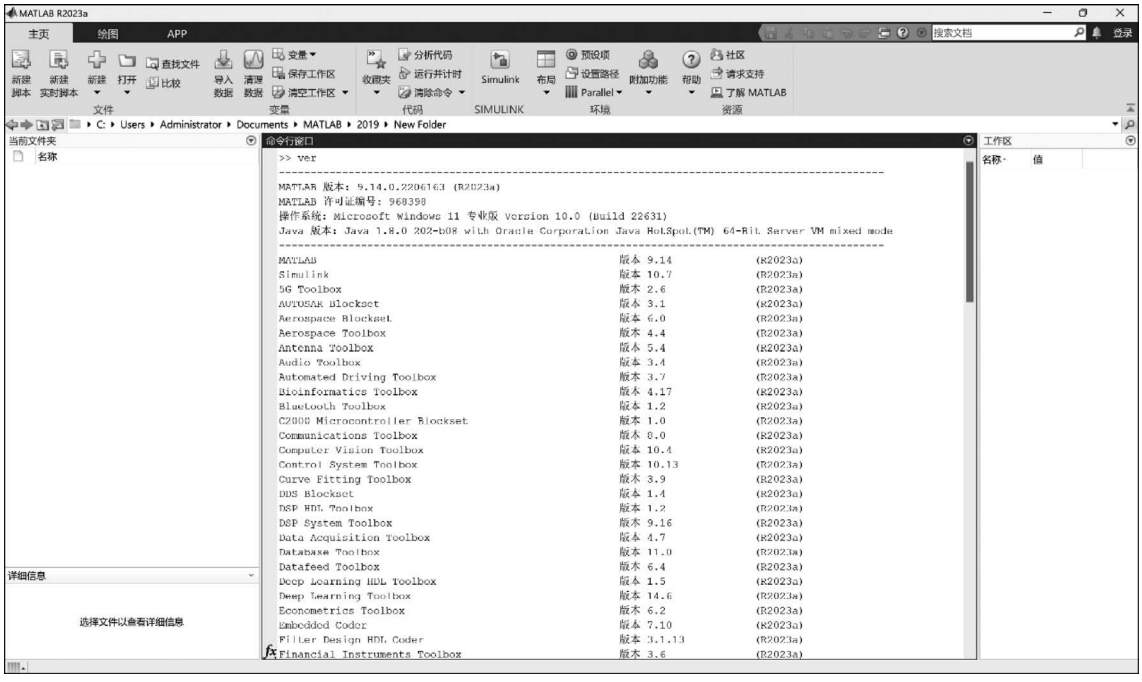


图 1-3　MATLAB 版本信息查看界面

1.2　命令行窗口

使用 MATLAB 时，用户直接在命令行窗口中输入各个语句即可输入命令并显示结果。例如，创建名为 a 的变量，并按 Enter 键执行。

```
a = 1
```

MATLAB 会将变量 a 添加到工作区，并在命令行窗口中显示结果。

```
a =
    1
```

同样，用户可以直接在命令行窗口创建更多变量，并进行运算，例如，

```
b = 2
```

```
b =
     2
```

```
c = a + b
```

```
c =
     3
```

```
d = cos(a)
```

```
d =
    0.5403
```

若未指定输出变量，MATLAB 将自动使用变量 ans（answer 的缩略形式）来存储最新的计算结果。

```
sin(a)
```

```
ans =
    0.8415
```

在多行上输入多个语句，可以在输入语句之间使用 Shift＋Enter 组合键进行分隔，输入完成后，按 Enter 键执行所有语句。当在多行上输入成对关键字语句（例如 for 和 end）时，不需要执行此操作。

此外，还可以使用逗号或分号在同一行上输入多个语句，以逗号结尾的命令会显示其结果，以分号结尾的命令不显示结果。例如，在命令行上输入以下三个语句，MATLAB 在命令行窗口中仅显示 A 和 C 的值。

```
A = magic(5),B = ones(5) * 4.7;C = A./B
```

```
A =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
C =
    3.6170    5.1064    0.2128    1.7021    3.1915
    4.8936    1.0638    1.4894    2.9787    3.4043
    0.8511    1.2766    2.7660    4.2553    4.6809
    2.1277    2.5532    4.0426    4.4681    0.6383
    2.3404    3.8298    5.3191    0.4255    1.9149
```

Tips：

- MATLAB 区分大小写，它的命令和函数全是小写的。MATLAB 中变量名是以字母开头，可以由字母、数字或下画线组成，最多 63 个字符。
- 注释符（%）后面的内容为注释，对 MATLAB 的计算不产生任何影响。

- 按向上箭头键(↑)和向下箭头键(↓)可以重新调用以前的命令,在空白命令行中或在输入命令的前几个字符之后按箭头键。例如,要重新调用命令 b=2,可以输入 b,然后按向上箭头键。
- 可以计算已在命令行窗口中的任何语句。选择命令行窗口中想要执行的语句并右击,在下拉菜单中选择“执行所选内容”选项。

所有输入的变量会在工作区进行显示和存储,如图 1-4 所示。

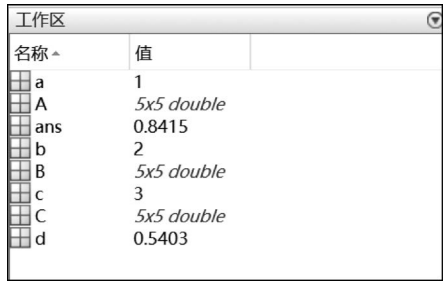


图 1-4 工作区变量

工作区变量包含在 MATLAB 中创建或从数据文件或其他程序导入的变量。使用 who 命令可以查看当前工作区中所有的变量名,使用 whos 命令可以查看工作区所有变量名及其数据结构、占用空间等信息。

```
whos
```

```
  Name        Size              Bytes  Class     Attributes
  A           5x5                 200  double
  B           5x5                 200  double
  C           5x5                 200  double
  a           1x1                   8  double
  ans         1x1                   8  double
  b           1x1                   8  double
  c           1x1                   8  double
  d           1x1                   8  double
```

要清除工作区中的某些变量,直接在 clear 命令后面列出这些变量名即可,变量名之间用空格分隔。想要清除工作区中所有变量,直接输入 clear 命令并按 Enter 键执行即可。

需要注意的是,退出 MATLAB 后工作区变量不会保留,但可使用 save 命令保存数据以供将来使用,保存后系统会以二进制的.mat 文件将工作区所有变量保存在当前工作文件夹中,默认文件名为 MATLAB.mat,也可以自定义文件名,实例中将工作区变量保存名为 myfile.mat。

```
save myfile.mat
```

使用 load 命令将 myfile.mat 文件中的数据还原到工作区,或者单击“主页→导入数据”按钮进行数据导入,单击“导入数据”按钮后会出现如图 1-5 所示的数据导入向导,按照向导提示即可导入.mat 文件数据到工作区。

```
load myfile.mat
```

图 1-5　数据导入向导

1.3　脚本和实时脚本

除了在命令行窗口中直接输入命令，还可以创建脚本和实时脚本以编辑和调试具有一定复杂功能的代码段。

1.3.1　脚本

脚本包含多行连续的 MATLAB 命令和函数调用的文件，脚本文件的扩展名为.m，一般称为.m 文件。创建脚本的方式有两种，第一种是单击“主页→新建脚本”按钮；第二种是直接在命令行中使用 edit 命令，例如要新建一个名为 mysphere.m 的脚本文件。

```
edit mysphere.m
```

编写代码时，可以使用%符号添加代码注释，方便其他人员理解代码，且有助于记忆，例如：

```
%绘制一个半径为 r 的球体.
[x,y,z] = sphere;                    %创建一个球体
r = 2;
surf(x*r,y*r,z*r)                    %给球体的三个坐标赋值并绘制
axis equal                           %设置坐标轴为等长.
%计算表面积和体积.
A = 4*pi*r^2;
V = (4/3)*pi*r^3;
```

将脚本文件保存为.m 文件后，单击“运行”按钮运行脚本，或在命令行窗口中直接输入脚本名称运行脚本。

1.3.2 实时脚本

实时脚本可以通过设置文本格式添加代码的说明和注释，进一步增强代码的可读性，有助于用户查看代码的输出并进行交互，实时脚本可以输入文本、方程和图像进行标注，并可分解进行运行。

创建脚本的方式有两种，第一种是单击“主页→文件部分→新建实时脚本”按钮；第二种是直接在命令行中使用 edit 命令。例如，要新建一个名为 mysphere.mlx 的脚本文件，编译环境如图 1-6 所示。

```
edit mysphere.mlx
```

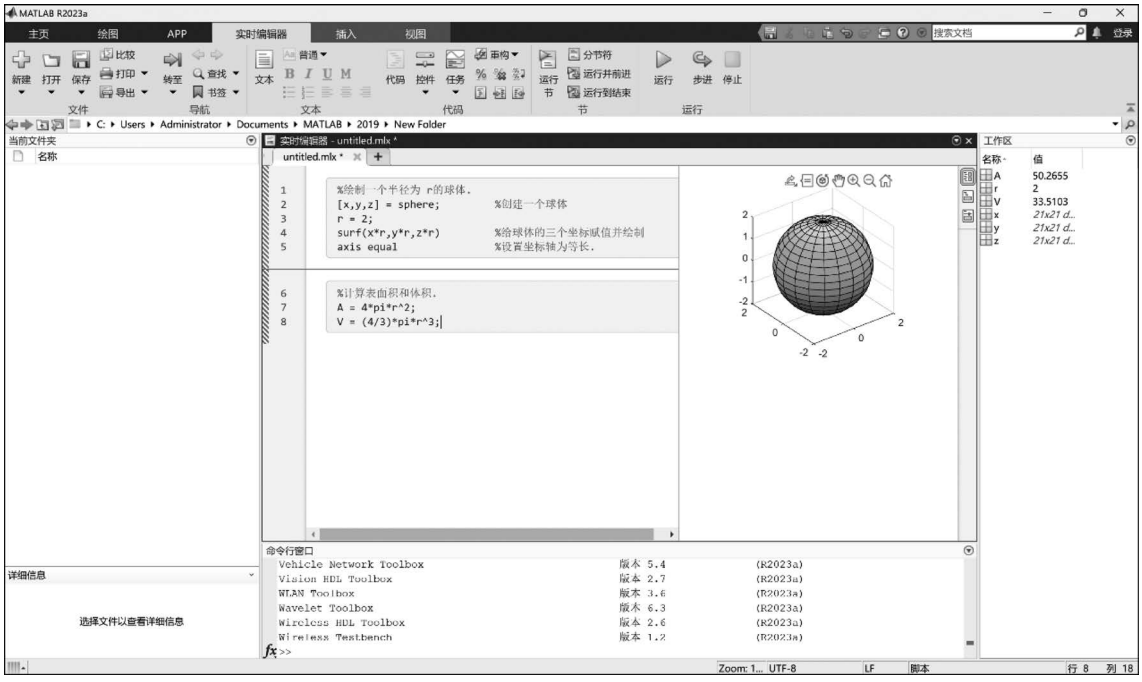


图 1-6 实时脚本编译环境

在实时脚本文件中，单击“实时编译器→代码”按钮，即可插入可执行的代码；单击“实时编辑器→文本”按钮，插入文本文字，可以在右侧下拉菜单中设置不同大纲级别和不同字体。若想要某一段代码实现一组功能并进行单独调试，可以单击“分节符”按钮添加小节；单独分节的代码可以单击“运行节”按钮或直接双击该代码段左侧的蓝色区域进行单独运行。

用户如果想要在文本中插入方程对代码内容进行详细注释，可以单击“插入→方程”按钮，编辑并插入方程。用户也可以根据需要插入图像、超链接等对象，如图 1-7 所示。

图 1-7 实时脚本中允许插入的对象

1.3.3 帮助和文档

所有 MATLAB 函数都有帮助文档，这些文档也包含一些介绍函数输入、输出和调用语法的示例，其强大而详细的功能足以支撑用户自学。从命令行访问帮助文档有多种方法，例如使用 doc 命令在单独的窗口中打开函数文档。

```
doc mean
```

在键入函数输入参数的左括号之后暂停，此时命令行窗口中会显示相应函数的提示框，在提示框右下角有蓝色字体“更多帮助...”，单击此链接也可以打开函数的帮助文档。

使用 help 命令可在命令行窗口中查看相应函数的简明文档。

```
help mean
```

直接单击主页右上角选项卡的帮助图标可以访问完整的产品文档，帮助窗口如图 1-8 所示。初学者一定要习惯使用 MATLAB 自带的帮助和文档功能，这将能够帮助大家持续性地学习。

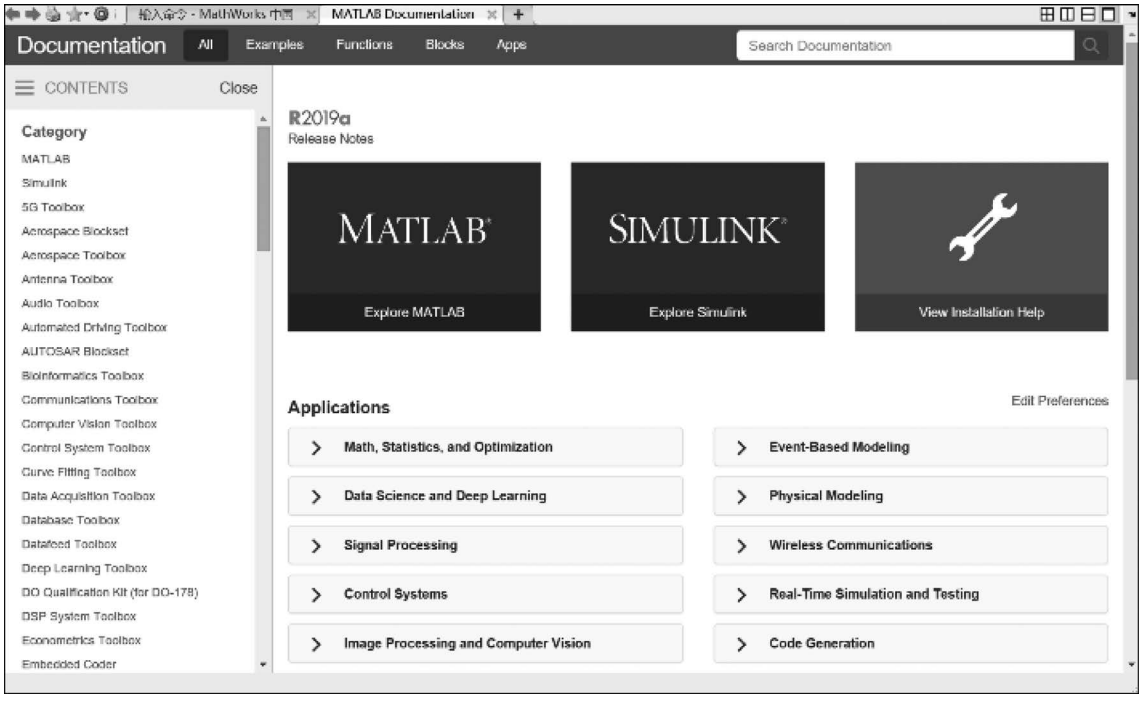


图 1-8　MATLAB 的帮助窗口

Tips：

- MATLAB 是在指定路径中查找脚本及其他文件的。要运行脚本，该文件必须位于当前文件夹或搜索路径中的某个文件夹内。在默认情况下，当前文件夹为安装程序创建的 MATLAB 文件夹。

- 若要将程序存储在其他文件夹，或者要运行其他文件夹中的程序，需要将该文件夹路径添加到搜索路径。具体方法为：单击“主页→设置路径”按钮 设置路径，将弹出如图 1-9 所示的设置路径对话框，单击“添加文件夹...”按钮，即可根据提示将用户需要的文件夹添加到搜索路径中。

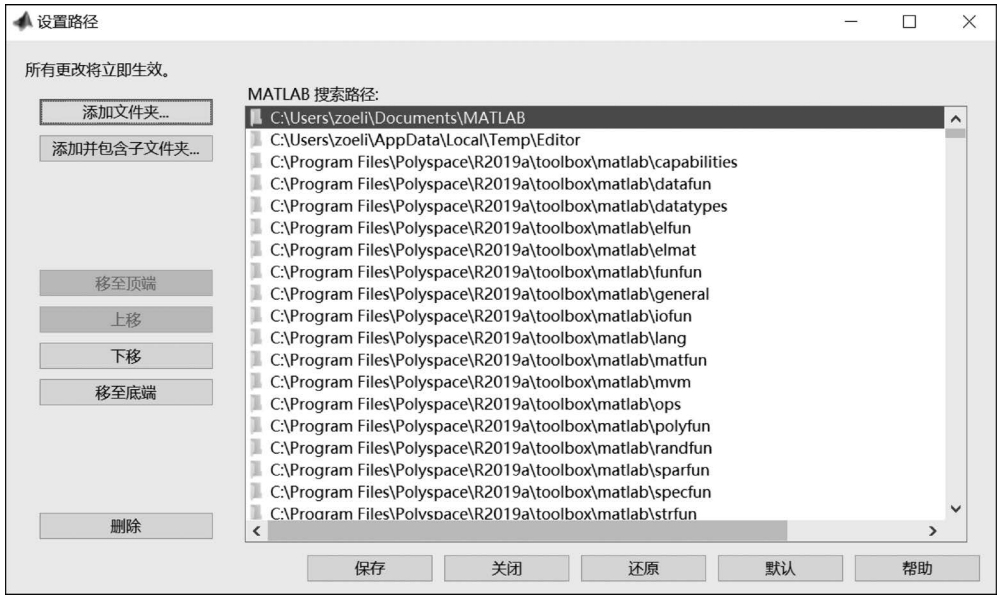


图 1-9　设置路径对话框

第2章 矩阵和矩阵运算

第2集
微课视频

矩阵运算是 MATLAB 的核心操作，MATLAB 提供了易于使用的矩阵操作语法，不需要烦琐的循环和索引操作便可以高效地处理大规模矩阵运算，极大地提高了编程效率。主要内容如下。

(1) 掌握 MATLAB 矩阵及特殊矩阵创建及存储方法，掌握矩阵元素位置索引及删除方法。

(2) 掌握 MATLAB 矩阵运算和实际矩阵运算的共同点和区别。

2.1 矩阵的创建

2.1.1 矩阵的构建

矩阵是 MATLAB 最基本的数据结构。为了尽可能简化运算，MATLAB 将所有变量都看作矩阵或多维数组。MATLAB 将标量看作 1×1 维矩阵，将向量看作只包含 1 行或 1 列的矩阵。矩阵是按行和列排列的元素的二维矩形数组，矩阵的元素可以是数字、逻辑值(true 或 false)、日期和时间、字符串或者其他 MATLAB 数据类型，即使一个数字也能以矩阵的形式存储。例如，包含值为 100 的变量 A 被存储为 double 类型的 1×1 维矩阵。

```
A = 100;
whos A
```

```
Name        Size              Bytes  Class     Attributes
A           1×1                   8  double
```

矩阵以方括号为分隔符，一行数据的元素之间用空格或逗号分隔，行与行之间用分号分隔。例如，创建只有一行的矩阵(行向量)，包含 4 个数字元素。用 size()函数可以测取返回的矩阵行和列的数量。

```
A = [12 62 93 -8]
sz = size(A)
```

```
A = 1×4
    12    62    93    -8
sz = 1×2
     1     4
```

再创建一个 2 行 2 列的矩阵 $\boldsymbol{B}$。

```
B = [12  62; 93  -8]
sz = size(B)
```

```
B = 2×2
    12    62
    93    -8
sz = 1×2
     2     2
```

另外，可以创建复数矩阵，其中，复数位可以由 i 或 j 代表，例如：

```
C = [i+9i,2+8i,3+7j;4+6j  5+5i,6+4i;7+3i,8+2j  1i]
```

```
C =
   0.0000 +10.0000i   2.0000 + 8.0000i   3.0000 + 7.0000i
   4.0000 + 6.0000i   5.0000 + 5.0000i   6.0000 + 4.0000i
   7.0000 + 3.0000i   8.0000 + 2.0000i   0.0000 + 1.0000i
```

可以通过 R＝real(C)，I＝imag(C)函数提取复数矩阵实部和虚部。

```
R = real(C)
I = imag(C)
```

```
R =
     0     2     3
     4     5     6
     7     8     0
I =
    10     8     7
     6     5     4
     3     2     1
```

Tips：

使用 size()命令与 length()命令可以测量矩阵或向量的维度，size(A)返回一个行向量，其元素是矩阵 A 的相应维度的长度。length(A)返回矩阵 A 中最大维度的值：对于向量，长度仅仅是元素数量；对于具有更多维度的矩阵，即为 max(size(A))；空数组的长度为零。

2.1.2 创建特殊矩阵

MATLAB 中有许多函数可以帮助用户创建具有特定值或特定结构的矩阵。例如，zeros()函数和 ones()函数可以创建元素全部为 0 或全部为 1 的矩阵，函数的参数分别是矩阵的行数和列数。

```
A = zeros(3,2)
```

```
A = 3×2
     0     0
     0     0
     0     0
```

```
B = ones(2,4)
```

```
B = 2×4
     1     1     1     1
     1     1     1     1
```

eye()函数可以创建单位阵，输入参数为单位阵的维度。

```
C = eye(3)
```

```
C = 3×3
     1     0     0
     0     1     0
     0     0     1
```

diag()函数将输入元素放在矩阵的对角线上。例如，创建一个行向量 ***D***，其中包含 4 个元素。然后创建一个 4×4 矩阵 ***E***，其对角元素是 ***D*** 的元素。

```
D = [12 62 93 -8];
E = diag(D)
```

```
E = 4×4
    12     0     0     0
     0    62     0     0
     0     0    93     0
     0     0     0    -8
```

magic()函数可以创建每行、每列，以及主、副对角线上各 n 个元素之和都相等的魔方阵，输入参数为魔方阵的维数。

```
F = magic(4)
```

```
F =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

rand()函数可以返回一个在区间(0,1)内均匀分布的随机数；rand(n)返回 n×n 的随机数矩阵；rand(s)返回由随机数组成的矩阵，大小由向量 s 指定。

```
r1 = rand(3)                          % 生成一个由介于 0～1 的均匀分布的随机数组成的 3×3 矩阵
r2 = -5 + (5 + 5) * rand(1, 3)        % 生成(-5,5)内均匀分布的数字组成的 1×3 行向量
r3 = randi([10 50],1,3)               % 生成在 10～50 均匀分布的 1×3 随机整数矩阵
r4 = rand + i * rand                  % 生成一个实部和虚部位于区间(0,1)内的随机复数
```

```
r1 =
    0.6797    0.1190    0.3404
    0.6551    0.4984    0.5853
    0.1626    0.9597    0.2238
r2 =
    3.3083    0.8526    0.4972
r3 =
    47    21    41
r4 =
   0.7537 + 0.3804i
```

2.1.3 矩阵的串联

矩阵的串联是使用方括号将现有矩阵连接在一起。例如，使用逗号将两个行向量水平串联起

来，形成一个更长的行向量。

```
A = ones(1,4);
B = zeros(1,4);
C = [A B]                          % 或使用 C = [A,B]
```

```
C = 1×8
     1     1     1     1     0     0     0     0
```

也可以使用分号将 ***A*** 和 ***B*** 垂直串联为一个矩阵的两行。

```
D = [A;B]
```

```
D = 2×4
     1     1     1     1
     0     0     0     0
```

要串联两个矩阵，它们的大小必须兼容。即水平串联矩阵时，它们的行数必须相同；垂直串联矩阵时，它们的列数必须相同，否则将会报错。

2.1.4 冒号表达式

冒号表达式是创建元素连续且均匀分布的矩阵的便捷方式。冒号表达式的格式为 v=s1:s2:s3，该函数将生成一个行向量 v，其中，s1 为向量的起始值，s2 为步距，该向量将从 s1 出发，每隔步距 s2 取一个点，直至不超过 s3 的最大值，构成一个向量。例如，创建一个行向量，其元素是 1～10 的整数。

```
A = 1:1:10
B = 0:0.2:pi
C = 13:13:1000;                    % 1～1000 内所有能被 13 整除的整数
```

```
A = 1×10
     1     2     3     4     5     6     7     8     9    10
B =
  0      0.2000    0.4000    0.6000    0.8000    1.0000    1.2000    1.4000    1.6000
1.8000   2.0000    2.2000    2.4000    2.6000    2.8000    3.0000
```

可以省略 s2，使用冒号表达式创建一个在任何范围内以步距为 1 的数字序列。

```
D = -2.5:2.5
```

```
D = 1×6
   -2.5000   -1.5000   -0.5000    0.5000    1.5000    2.5000
```

更改序列的增量值，需要在该范围的起始值和结束值之间指定增量值，以冒号分隔，例如创建 0～10 的范围内间隔为 2 的行向量。

```
E = 0:2:10
```

```
E = 1×6
     0     2     4     6     8    10
```

如果要建立一个递减的向量，步距使用负数即可。

```
F = 6:-1:0
```

```
F = 1×7
     6     5     4     3     2     1     0
```

可以按非整数值递增。如果增量值不能均分指定的范围，MATLAB 会在超出范围之前，在可以达到的最后一个值处自动结束序列。

```
G = 1:0.2:2.1
```

```
G = 1×6
    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
```

MATLAB 还提供了另外两个函数生成不同的等间距行向量。

若要生成线性等间距点向量，可以使用 MATLAB 函数 v=linspace(n1,n2,N)，其中向量的点数为 N(默认值为 50)，起始值为 n1，终止值为 n2。例如，创建一个由区间[−5,5]中的 7 个等距点组成的向量。

```
y1 = linspace(-5,5,7)
```

```
y1 =
   -5.0000   -3.3333   -1.6667         0    1.6667    3.3333    5.0000
```

若要生成对数等间距点向量，在某些特定的领域要求以对数等间距生成行向量，可以使用 logspace()函数生成行向量，其调用格式为 w=logspace(n1,n2,N)，该语句将生成数据点的行向量，其第一个值为 lgn1，最后一个值为 lgn2。若想在[10^{-3},10^{4}]频率段内生成 5 个点，则需要给出命令。

```
W = logspace(-3,4,5)
```

```
w = 1×5
104 ×
    0.0000    0.0000    0.0003    0.0178    1.0000
```

2.1.5 矩阵位置索引

MATLAB 中的每个变量都可以认为是一个矩阵，可以使用索引访问数组的某个元素或子矩阵。最常见的方法是按照元素位置进行索引，即 B=A(v1,v2)，其中 v1 表示子矩阵要保留的行号构成的向量，v2 表示要保留的列号构成的向量，这样从矩阵 A 中提取有关的行和列，就可以构成子矩阵 B 了。以 4×4 魔方阵 ***A*** 为例，找到 ***A*** 中第 4 行第 2 列的元素。

```
A = magic(4)
B = A(4,2)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
B = 14
```

用户还可以在一个矩阵中指定多个元素的索引，从而一次引用多个元素。例如，访问矩阵 ***A*** 的第二行中的第一个和第三个元素。

```
C = A(2,[1 3])
```

```
C = 1×2
     5    10
```

要访问某个行范围或列范围内的元素，可以使用冒号表达式。例如，访问矩阵 ***A*** 中第 1 到 3 行、第 2 到 4 列中的元素。

```
D = A(1:3,2:4)
```

```
D = 3×3
     2     3    13
    11    10     8
     7     6    12
```

若不知道矩阵 ***A*** 中有多少列，可以使用关键字 end 指定第二直至最后一列，结果与上面一致。

```
E = A(1:3,2:end)
```

```
E = 3×3
     2     3    13
    11    10     8
     7     6    12
```

若要访问所有行或所有列，可以使用冒号运算符。例如，返回矩阵 ***A*** 的整个第三列。

```
F = A(:,3)
```

```
F = 4×1
     3
    10
     6
    15
```

2.1.6 从矩阵中删除行或列

删除矩阵的行或列，最简单的方法是将该行或列设置为等于空方括号[]。例如，创建一个 4×4 矩阵并删除第 2 行和第 3 列。

```
A = magic(4);
A(2,:) = []
A(:,3) = []
```

```
A = 3×4
    16     2     3    13
     9     7     6    12
     4    14    15     1
A = 3×3
    16     2    13
     9     7    12
     4    14     1
```

此方法可以扩展到任何维度的矩阵。例如，创建一个随机的 3×3×3 矩阵，然后删除第 3 维第 1 个矩阵中的所有元素。请自行输入程序，得到结果。

```
B = rand(3,3,3)
B(:,:,1) = []
```

2.2 矩阵的基本运算

2.2.1 矩阵的代数运算

变量之间的有限次的加、减、乘、除、乘方和开方等运算称为代数运算。

1. 加减法运算

MATLAB矩阵的加减法运算与正常代数运算含义相同，参与加减的两个矩阵要有相同的维度。

```
A = [1 1 1];
B = [1 2 3];
C = A+B
```

```
C =
     2     3     4
```

参与加减法运算的两个矩阵之一为标量，则会将其逐一加(减)于另一个矩阵的每个元素中。

```
A = [1 1 1];
D = 3;
E = D-A
```

```
E =
     2     2     2
```

若一个矩阵为 $n\times m$ 矩阵，另一个为 $n\times 1$ 列向量或 $1\times m$ 行向量，允许将列向量或行向量逐一加或减到另一个矩阵的各列或各行中，以得出新的和矩阵或差矩阵。例如，

```
A = [5;6];
B = [1 2;3 4];
C = A+B
D = B-A'
```

```
C =
     6     7
     9     10
D =
    -4    -4
    -2    -2
```

其他情况，若参与加减运算的两个矩阵维数不匹配，MATLAB将给出错误信息，提示不能相加或相减。

2. 矩阵乘法

矩阵乘法要满足参与运算的两个矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$，矩阵 $\boldsymbol{A}$ 的列数与矩阵 $\boldsymbol{B}$ 的行数相等，或其一为标量，则称矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$ 是可乘的或称矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的维数是相容的。

假设 $\boldsymbol{A}$ 为 $n\times m$ 矩阵，其各元素为 a_{ij}，$i=1,2,\cdots,n$，$j=1,2,\cdots,m$，$\boldsymbol{B}$ 为 $m\times r$ 矩阵，其各元素为 b_{ij}，$i=1,2,\cdots,m$，$j=1,2,\cdots,r$，$\boldsymbol{C}=\boldsymbol{AB}$ 为 $n\times r$ 矩阵，其各个元素为

$$c_{ij}=\sum_{k=1}^{m}a_{ik}b_{kj},\quad i=1,2,\cdots,n,\ j=1,2,\cdots,r$$

MATLAB 语言中两个矩阵的乘法由 C=A * B 直接求出，不需要指定 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 矩阵的维数。若维数相容，则可以准确无误地获得乘积矩阵；如果二者的维数不相容，则将给出错误信息。

3. 矩阵的左除和右除

MATLAB 中用“\”运算符号表示两个矩阵的左除，代码 A\B 为求方程 $\boldsymbol{AX}=\boldsymbol{B}$ 的解 $\boldsymbol{X}$。若 $\boldsymbol{A}$ 为非奇异方阵，则 $\boldsymbol{X}=\boldsymbol{A}^{-1}\boldsymbol{B}$。如果矩阵 $\boldsymbol{A}$ 不是方阵，也可以求出 $\boldsymbol{X}=\boldsymbol{A}\backslash\boldsymbol{B}$，这时将使用最小二乘解法来求取 $\boldsymbol{AX}=\boldsymbol{B}$ 中的 $\boldsymbol{X}$ 矩阵。

MATLAB 中用“/”运算符号表示两个矩阵的右除，相当于求方程 $\boldsymbol{XA}=\boldsymbol{B}$ 的解。$\boldsymbol{A}$ 为非奇异方阵时，代码 B/A 表示求解 $\boldsymbol{X}=\boldsymbol{B}\boldsymbol{A}^{-1}$，但在计算方法上存在差异，更精确的计算方法为 $\boldsymbol{B}/\boldsymbol{A}=(\boldsymbol{A}'\backslash\boldsymbol{B}')'$。

4. 矩阵的乘方与开方

只有方阵才能进行乘方与开方运算，长方形矩阵是不能进行乘方与开方的。矩阵的乘方与开方在数学上可以统一地表示成 $\boldsymbol{A}^x$，但处理方法可能存在差异。

(1) 矩阵乘方运算。一个矩阵的乘方运算可以在数学上表述成 $\boldsymbol{A}^x$。如果 x 为正整数，则乘方表达式 $\boldsymbol{A}^x$ 的结果可以将 $\boldsymbol{A}$ 矩阵自乘 x 次得出。如果 x 为负整数，则可以将 $\boldsymbol{A}$ 矩阵自乘 $-x$ 次，然后对结果进行求逆运算，就可以得出该乘方的结果。如果 x 是一个分数，如 $x=n/m$，其中 n 和 m 均为整数，相当于将 $\boldsymbol{A}$ 矩阵自乘 n 次，然后对结果再开 m 次方。在 MATLAB 中的代码统一表示成 F=A^x。

(2) 矩阵开方运算。从数学公式上看，矩阵 $\boldsymbol{A}$ 自乘几次是可以得出唯一解的，其结果再作 m 次开方则应该有 m 个不同的根。考虑 $\sqrt[3]{-1}$ 的一个根是 -1，对该根在复数平面内旋转 120°可以得到第二个根，再旋转 120°则可以得出第三个根。可以将结果乘以复数标量 $\delta=e^{2\pi j/3}$ 实现 120°旋转。如果想开 m 次方，则可以将结果 $A^{n/m}$ 乘以 $\delta_k=e^{2\pi j/m}$，其中，$k=1,2,\cdots,m-1$。

2.2.2 矩阵的点运算

两个相同维数的矩阵 $\boldsymbol{A}$ 与 $\boldsymbol{B}$，如果对其相应元素单独进行乘法运算，则可以得出一个新的矩阵 $\boldsymbol{C}$，其中，$c_{ij}=a_{ij}b_{ij}$，矩阵 $\boldsymbol{C}$ 称为 $\boldsymbol{A}$、$\boldsymbol{B}$ 矩阵的 Hadamard 乘积，又称为点乘积。

矩阵的点乘积可以由点运算直接计算出来，代码为 C=A. * B，这种运算和普通乘法运算是不同的。此外，可以定义两个矩阵之间或矩阵与标量之间其他的点运算，即在实际运算符号前面加一个圆点(.)，如点乘方算符(.^)、点除算符(./与.\)等。如果参与运算的两个变量 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 都是矩阵，则要求这两个矩阵的维数相同或其一为标量，否则将给出错误信息。

点运算在 MATLAB 中起着很重要的作用，如绘制 $\boldsymbol{x}^5$ 的函数曲线需要生成一个向量 $\boldsymbol{x}$，再对其每个元素单独求 5 次方。例如，[x^5]不能直接写成 x^5，必须写成 x.^5。

根据下面例子，A.^A 实际上是完成了对应元素各取乘方运算。

```
A = [1,2,3;4 5,6;7,8 0];
B = A.^A                    % 对应元素单独运算可以求点乘方
```

```
B =
           1           4          27
         256        3125       46656
      823543    16777216           1
```

实际上该语句完成的是下面矩阵的计算。

$$\boldsymbol{B}=\begin{bmatrix}1^1 & 2^2 & 3^3\\ 4^4 & 5^5 & 6^6\\ 7^7 & 8^8 & 0^0\end{bmatrix}$$

2.2.3 矩阵的逻辑运算与比较运算

1. 矩阵的逻辑运算

在MATLAB语言中，如果一个数的值为0，可以认为它为逻辑0，否则为逻辑1。假设矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 均为 $n\times m$ 矩阵，MATLAB定义了如下逻辑运算。

(1) 矩阵的与运算。在MATLAB中用 & 表示矩阵的与运算。A&B表示矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 相应元素的与运算。若两个矩阵相应元素均非零，则该结果元素的值为1，否则该元素为0。

(2) 矩阵的或运算。在MATLAB中用A|B表示矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$ 的或运算，如果两个矩阵相应元素存在非零值，则该结果元素的值为1，否则该元素为0。

(3) 矩阵的非运算。在MATLAB中用～A表示矩阵的非运算。若矩阵元素均为0，则结果为1，否则为0。

(4) 矩阵的异或运算。矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的异或运算可以表示为xor(A,B)。若相应的两个数一个为0一个为非0，则结果为1，否则为0。

2. 矩阵的比较运算

MATLAB定义了各种比较关系，如C=A>B，当矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 满足 $a_{ij}>b_{ij}$ 时，$c_{ij}=1$，否则 $c_{ij}=0$。MATLAB支持等于关系(用“==”表示)、大于或等于关系(用“>=”表示)和不等于关系(用“～=”表示)。例如，

```
A = [1,2,3;4 5,6;7,8 0];
K = A>=5
```

```
k =   3×3
    0    0    0
    0    1    1
    1    1    0
```

得出的结果为逻辑型矩阵，逻辑条件A>=5满足的位置上的元素为逻辑1，不满足的位置上的值为逻辑0。

2.2.4 矩阵的转置、翻转与旋转

MATLAB提供了对矩阵进行转置、翻转和旋转的运算函数。

1. 矩阵转置

矩阵的转置记作 $\boldsymbol{A}^{\mathrm{T}}$，假设矩阵 $\boldsymbol{A}$ 为一个 $n\times m$ 矩阵，则其转置矩阵 $\boldsymbol{B}$ 的元素定义为 $b_{ji}=a_{ij}$，