

5 作业参考文档

5.1 MATLAB 入门

20 世纪 70 年代，美国新墨西哥大学计算机科学系主任克里夫·莫勒尔（Cleve Barry Moier, 1939—）为了减轻学生编程的负担，用 FORTRAN 编写了最早的 MATLAB，1984 年合作成立 MathWorks 公司正式把 MATLAB 推向市场。MATLAB 不但能编程还能用于数值计算以及图形显示，并用与控制系统以及工程设计。

1990 年，MathWorks 公司为 MATLAB 开发了一种新的用于图形控制及仿真模型建立的软件 Simulink，它是 Matlab 的一个扩展软件模块，这个模块为用户提供了一个用于建模仿真各种数学物理模型的软件，并且提供各种动态的结构模型，使用户可以快速方便地建模并仿真，而不必写任何程序。基于此优点，该工具很快被业界认可，并用于各种控制系统，成为国际控制界的标准计算软件。

MATLAB 编程工具不像 C 语言那样难以掌握，在仿真环境下用户只需简单地列出计算式，结果便会以数值或图形的方式显示出来。MATLAB 还具有工具箱函数库，能针对各个学科领域实现各种计算功能。此外，MATLAB 和其他高级语言也具有良好的接口，可以方便地与其他语言实现混合编程。

MATLAB 由主程序、Simulink 动态仿真系统和 MATLAB 工具箱三部分组成。主程序包括 MATLAB 语言、工作环境以及应用程序。Simulink 动态仿真系统是一个相互交互的系统，用户制作一个模拟系统，并动态控制它。工具箱就是 MATLAB 基本语句的各种子程序和函数库，分为功能性和学科性工具箱。功能性的工具箱主要用于扩展 MATLAB 的符号计算功能、图形建模功能、文字处理功能和与硬件的实时交互过程，如符号计算工具箱等；学科性工具箱则有较强的专业性，用于解决特定的问题，如信号处理工具箱和通信工具箱等。

如今，MATLAB 软件已成为对数值、线性代数以及其他一些高等应用数学课程进行辅助教学的有力工具。在工程技术界，MATLAB 软件也被用来构建与分析一些实际的数学模型，典型的应用包括数值计算、算法预设计与验证，以及一些特殊矩阵的计算应用，如统计、图像处理、自动控制理论、数字信号处理、系统识别和神经网络等。

5.1.1 MATLAB 安装

浙江大学的 MATLAB 校园版于 2017 年 11 月正式生效。MATLAB 校园版包括单机版和网络版两种形式，单机版适合用于个人电脑，网络版适合于机房、服务器等场所。下面介绍单机版的安装。

注册 MathWorks 帐户

登录注册页面：www.mathworks.com/login，选择“立刻创建一个（Create one!）”，如图 5.1 所示。

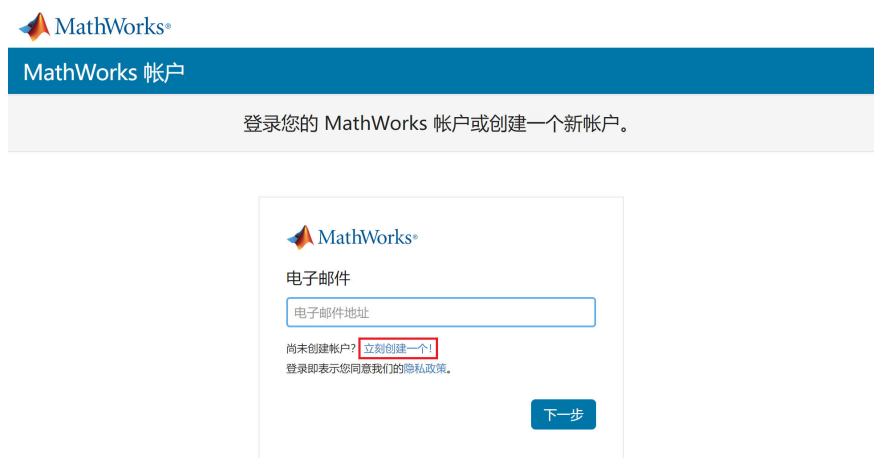


图 5.1 创建帐户

接下来填写帐户信息，请注意务必用学校的邮箱（@zju.edu.cn）作为电子邮箱地址（其它邮箱不识别），如图 5.2 所示。



图 5.2 填写帐户信息

登录账户，点击右上角名字，选择关联许可证（Associate License），如图 5.3 所示。

在“许可证号编号 * 或激活密钥”处填写浙江大学的 MATLAB 激活密钥。浙江大学学生的 MATLAB 激活密钥（Activation Key）为 11411-34616-78311-15943-52984，如图 5.4 所示。

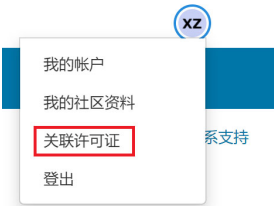


图 5.3 关联许可证

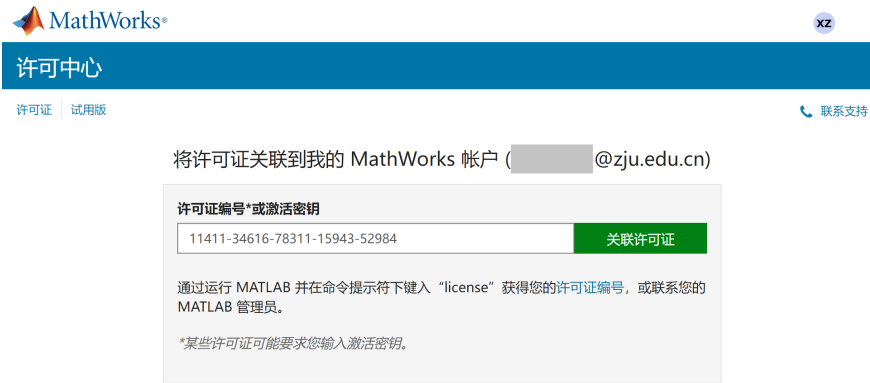


图 5.4 许可证编号

安装软件

在 MathWorks 帐号里，单击许可证号，如图 5.5 所示。选择“安装和激活（Install and Activate）”，下载安装程序（推荐在线安装），如图 5.6 所示。



图 5.5 安装软件

启动 MATLAB 安装软件后，选择“使用 MathWorks 帐户安装”。输入 MathWorks 帐户，电子邮件输入学校的邮箱，密码为 MathWorks 帐户密码。许可证选中列表中已经关联的许可证。安装路径一般建议选择默认的路径。

选择可能用到的工具箱。根据本课程作业需要，勾选 MATLAB、Simulink、Communication Toolbox 和 DSP System Tollbox，如图 5.7 所示。也可以全选。然后选择下一步进行安装。

安装 Simulink 和工具箱

Simulink 是美国 Mathworks 公司推出的 MATLAB 中的一种可视化仿真工具。Simulink 是一个模块图环境，用于多域仿真以及基于模型的设计。它支持系统设计、仿真、自动代码生成



图 5.6 下载安装程序

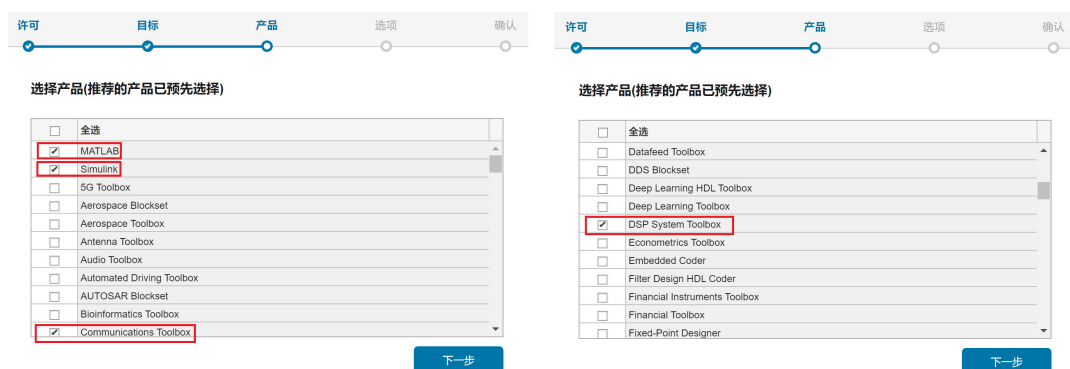


图 5.7 选择工具箱

以及嵌入式系统的连续测试和验证。Simulink 提供图形编辑器、可自定义的模块库以及求解器, 能够进行动态系统建模和仿真。

Simulink 与 MATLAB 相集成, 能够在 Simulink 中将 MATLAB 算法融入模型, 还能将仿真结果导出至 MATLAB 做进一步分析。Simulink 应用领域包括汽车、航空、工业自动化、大型建模、复杂逻辑、物理逻辑, 信号处理等方面。

如果初始安装 MATLAB 时没有安装 Simulink, 会导致找不到 Simulink 的按钮。可尝试两种方法: 一是在附加功能中搜索 Simulink 并安装; 二是在 MATLAB 命令行窗口中输入 Simulink, 将会出现安装提示。

因本课程作业需要, 需在 MATLAB/Simulink 中安装 DSP System Toolbox 和 Communications Toolbox。具体操作如下:

打开 MATLAB, 选择“附加功能”中的“获取附加功能”, 如图 5.8 所示。



图 5.8 获取附加功能

在弹出的“附加功能资源管理器”中左侧列表选中“应用程序（Applications）”下的“Signal Processing”，在右侧选中“DSP System Toolbox”并安装，如图 5.9 所示。

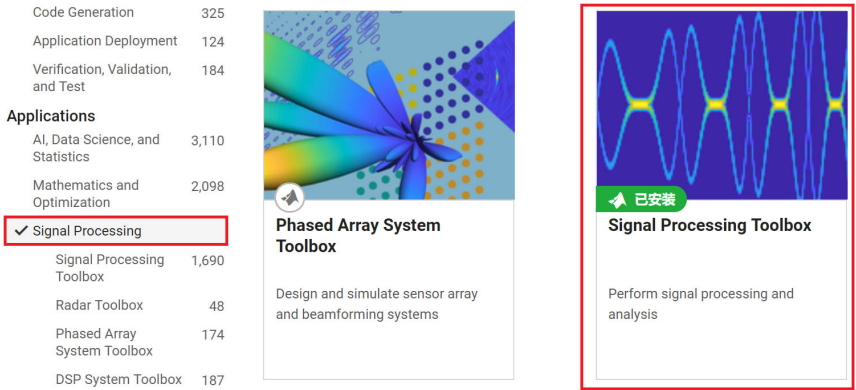


图 5.9 安装 DSP System Toolbox

在“附加功能资源管理器”中左侧列表选中“应用程序（Applications）”下的“Wireless Communications”，在右侧选中“Communications Toolbox”并安装，如图 5.10 所示。

MATLAB Campus-Wide License 更新

MATLAB Campus-Wide License 分为两种：个人版和机房版。个人版需要每年更新许可证文件，以恢复 MATLAB 使用。安装了个人版后，若在 MATLAB 软件中遇到即将过期的警告信息，可执行以下操作，更新许可证。

- 打开 MATLAB（如果许可证已经过期，则会启动激活客户端，你可以按照激活客户端的指示进行激活操作）。
- 在导航菜单栏，依次单击“帮助”“许可”“更新当前许可证”，如图 5.11 所示。
- 重新激活后，需重新启动 MATLAB 才能识别新许可证文件。

需要注意的是，成功完成激活后，在旧的许可证到期之前，你仍会在 MATLAB 中看到“许可证即将过期”的警告信息，你可以忽略该信息。旧的许可证到期后，MATLAB 将不会再发送此警告信息。

社区 1,080

按类别筛选

Using MATLAB

MATLAB 11,796

Using Simulink

Simulink 866

Physical Modeling 2,249

Event-based Systems 69

Real-Time Simulation and Testing 60

Workflows

Parallel Computing 196

Reporting and Database Access 205

Systems Engineering 21

Code Generation 325

Application Deployment 124

Verification, Validation, and Test 184

Applications

AI, Data Science, and Statistics 3,110

Mathematics and Optimization 2,098

Signal Processing 2,956

Image Processing and Computer Vision 4,194

Control Systems 1,434

Test & Measurement 782

RF and Mixed Signal 251

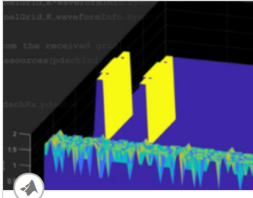
✓ Wireless Communications

Communications Toolbox 980

LTE Toolbox 88

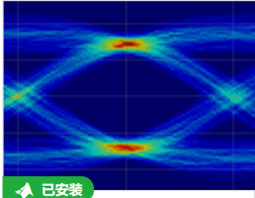
WLAN Toolbox 53

Wireless Communications (1,094)



5G Toolbox

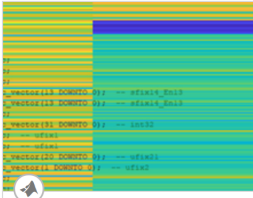
Simulate, analyze, and test the physical layer of 5G communications systems



已安装

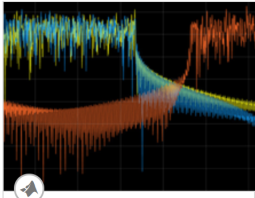
Communications Toolbox

Design and simulate the physical layer of communications systems



Wireless HDL Toolbox

Model LTE communications subsystems for FPGAs and ASICs



WLAN Toolbox

Simulate, analyze, and test the physical layer of WLAN communications systems



图 5.10 安装 Communications Toolbox

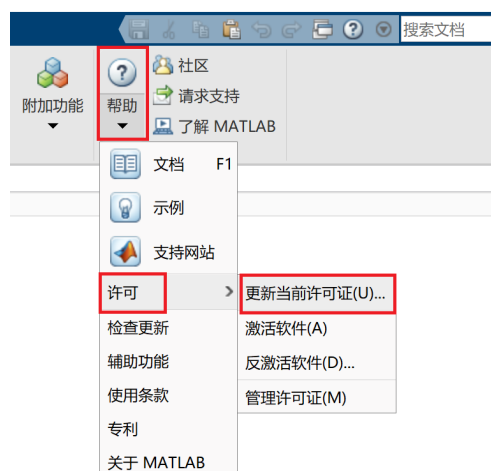


图 5.11 更新个人版 MATLAB 许可证

5.1.2 界面认识

启动 MATLAB 时，桌面会以默认布局显示，如图 5.12 所示，分别有菜单栏与工具栏、当前文件夹、命令行窗口和工作区等。

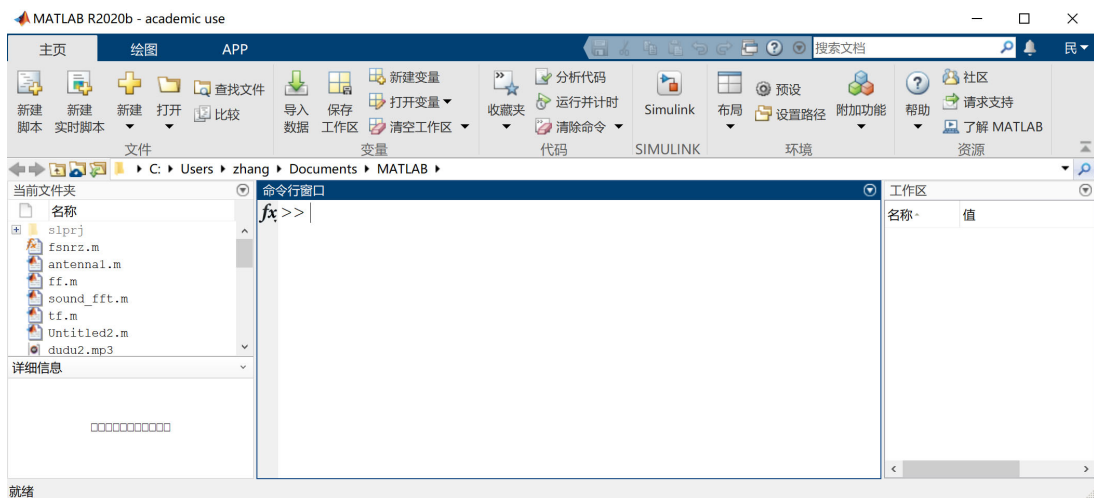


图 5.12 MATLAB 默认界面

菜单栏与工具栏

常用的操作与工具显示。

在工具栏中依次单击“主页”“预设”，弹出预设项窗口。单击“颜色”菜单，可以修改 IDE 中的语法高亮的颜色风格等；单击“字体”菜单，可以修改各类显示字体的类别和大小等，完成设定后，单击“应用”或“确定”保存修改。

在工具栏中依次单击“主页”“布局”，可以对界面布局进行修改。

当前文件夹

展示当前 MATLAB 代码保存位置以及路径下面有什么文件与文件夹。

在工具栏中单击“设置路径”可对 MATLAB 的搜索路径进行设置。

命令行窗口

写简单代码与显示运行结果。在“>>”后面输入语句，并敲击回车，语句就会即时执行（和 C、Pascal 等编译执行的语言不同）。

MATLAB 自带有非常详细的帮助文件。在命令行中输入“help xxx”即可获得这些信息，“xxx”是需要查询的函数、关键字以及运算符等。

工作区

显示变量与变量值、维度等信息，在此区域可以查看程序运行中用到的变量。

MATLAB 的变量采用矩阵或元胞矩阵的形式存储，双击对应的变量名就可以查看这个矩阵各元素的值。

5.1.3 MATLAB 变量

变量名

在 MATLAB 中，变量名是以字母或者下划线开头，后接字母、数字或下画线的字符序列，最多 63 个字符。变量名区分字母的大小写。标准函数名以及命令名一般用小写字母。图 5.13 中代码分别对变量 `a1` 赋值 `1`，`A1` 赋值 `'isee'`。如果变量名不符合命名规则，会报错，提示表达式无效。

变量赋值

Matlab 语句的通常形式为

变量 = 表达式;

表达式是用运算符将有关运算量连接起来的式子，其结果被赋给赋值号“=”左边的变量。若不想在屏幕上输出结果，可以在语句最后加分号“;”，如图 5.13 中的“`x=8;`”“`y='ZJU';`”。

如果语句很长，可用续行符“...”（三个点）续行，续行符的前面最好留一个空格，如图 5.13 中的“`w=sin(0.1*pi)+2*sin(0.2*pi)+3*sin(0.3*pi)+ ...`”。

工作区中列出了工作空间中的所有变量。可使用 `who` 命令显示工作空间中的所有变量，`whos` 命令查看工作空间中变量的详细属性。

`clear` 命令可清空工作空间中的所有变量，或“`clear 变量名 变量名`”清除指定的变量（变量名之间用空格分隔）。

一些变量系统已预定义，应尽量避免给系统预定义变量重新赋值，如图 5.14 所示。科学计算中通常需要用到很多基本常数，当然也可以使用手工定义的方法来使用这些常数，但是使用 MATLAB 内置的常数通常更加方便和精确。

- `pi`: 圆周率 π ;
- `inf`、`Inf`: 无穷大;
- `nan`、`NaN`: Not a Number, 一个不定值，如 `0/0`;
- `eps`: 浮点运算相对精度，实际应用中基本可以视为无穷小量（真正的无穷小量应该是 `realmin`）;
- `i`、`j`: 虚部单位，为了避免混淆通常写成 `1i`、`1j`;
- `ans`: 一个特殊变量，例如如果计算 `1+2` 但结果没有赋予一个变量，MATLAB 就会把结果默认赋予 `ans`。

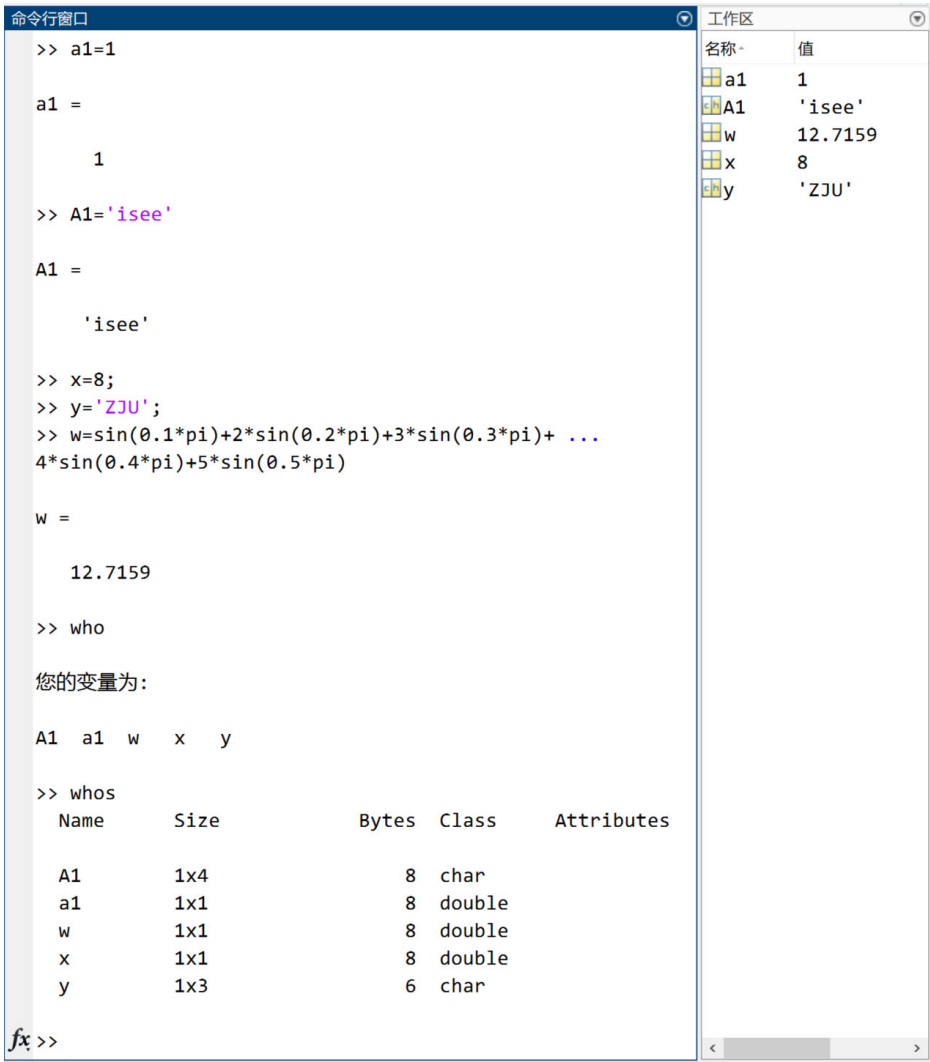


图 5.13 MATLAB 变量

变量保存

save 命令创建内存变量文件。**load** 命令导入内存变量文件。

save 文件名 变量名列表
load 文件名 变量名列表

变量名列表中各变量之间用空格分隔，如果不写变量名，则存储（导入）所有变量。保存后，在当前文件夹中产生文件名后缀为“**.mat**”的数据文件。

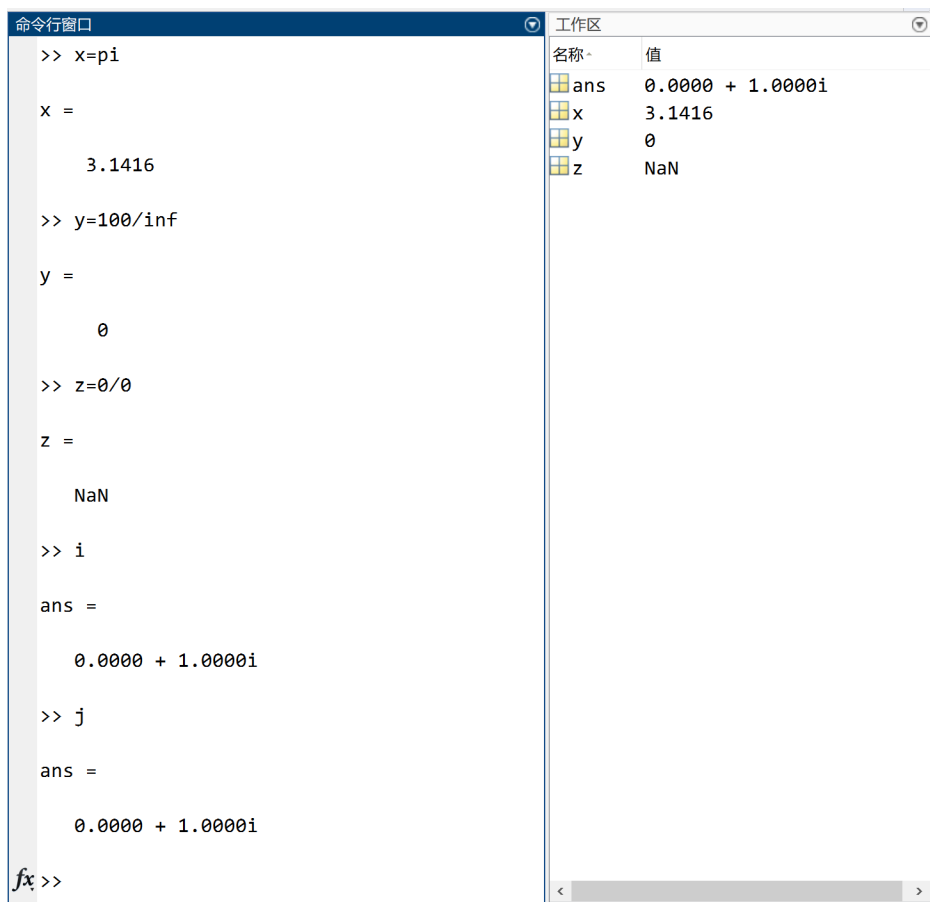


图 5.14 系统预定义变量

5.1.4 数组与矩阵

数组是计算机程序设计领域的概念，是在程序设计中，为了处理方便，把具有相同类型的若干变量按有序的形式组织起来的一种形式。这些按序排列的同类数据元素的集合称之为数组。在 MATLAB 中，一个数组可以分解为多个数组元素，这些数组元素可以是基本数据类型或是构造类型。因此按数组元素的类型不同，数组又可以分为数值数组、字符数组、单元数组、结构数组等各种类别。

矩阵是数学的概念，最早来自于方程组的系数即常数所构成的方阵。作为一种变换或映射算符的体现，矩阵运算有着明确而严格的数学规则。而数组运算是 MATLAB 软件定义的规则，其目的是为了使数据管理方便，操作简单，命令形式自然，执行计算有效。

对于 MATLAB 来说，数组或向量与二维矩阵在本质上没有区别。MATLAB 的变量都是以数组的形式储存的：一维的数组就是向量，二维的数组就是矩阵，三维或以上的数组就是多维数组。特殊的 1×1 的数组就是普通变量，还有 `[]` 是空数组，大小是 0。

MATLAB 的数据结构只有矩阵一种形式，所有的返回维数都是 2，所有的数据都以矩阵的形式保存。单个的数就是 1×1 的矩阵，数组或向量就是 $1 \times n$ 或 $n \times 1$ 的矩阵。

矩阵创建

利用直接输入法建立矩阵如图 5.15 所示。将矩阵的元素用方括号 “[]” 括起来，按矩阵行的顺序输入各元素，同一行的各元素之间用逗号 (,) 或空格分隔，行与行之间用分号 (;) 分隔。直接输入法中分号可以用回车代替。

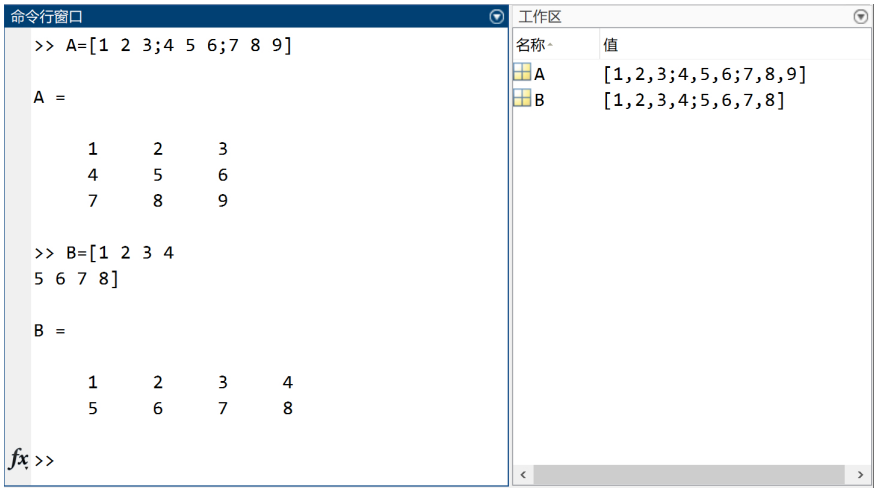


图 5.15 直接输入法建立矩阵

可利用冒号表达式创建矩阵或数组，如图 5.16 所示。**a:b:c** 产生一个由等差序列组成的向量，**a** 是首项，**b** 是公差，**c** 确定最后一项。若 **b=1**，则 **b** 可以省略。

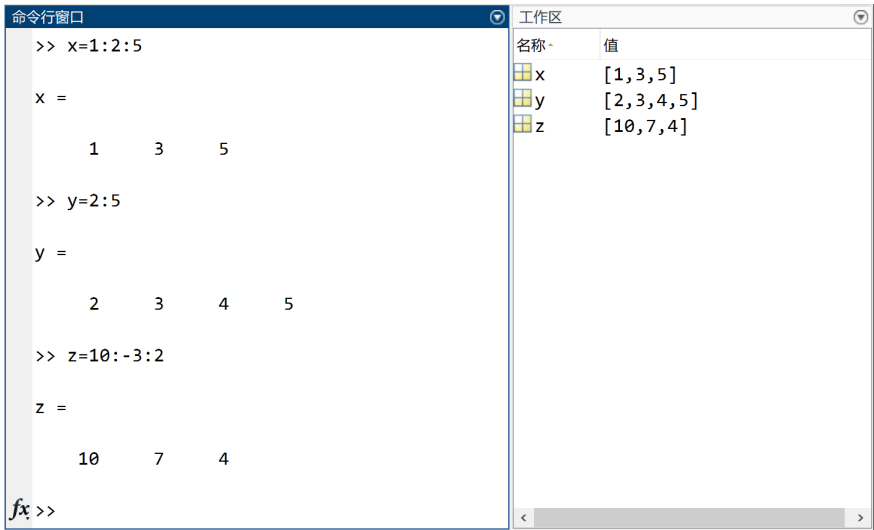


图 5.16 等差序列组成的向量

数组索引

如果矩阵 **A** 是一个 1×2 矩阵，第一个元素是 1，第二个元素是 2，可赋值 **A(1)=1**；**A(2)=2**。

大矩阵可以把小矩阵作为其元素，如 **B = [A; 3 4]**。

引用数组中的特定元素有两种方法。最常见的方法是指定行和列下标，例如 **A(i,j)** 矩阵 **A** 中的第 **i** 行，第 **j** 列元素。

使用单一下标引用数组中特定元素的方法称为线性索引。这种方法不太常用，但有时非常有用，即使用单一下标按顺序向下遍历每一列。例如 **A(1)** 引用 **A** 的第一个元素；**A(i)** 引用向量 **A** 中的第 **i** 个元素。按列、行顺序排列，如图 5.17 所示。

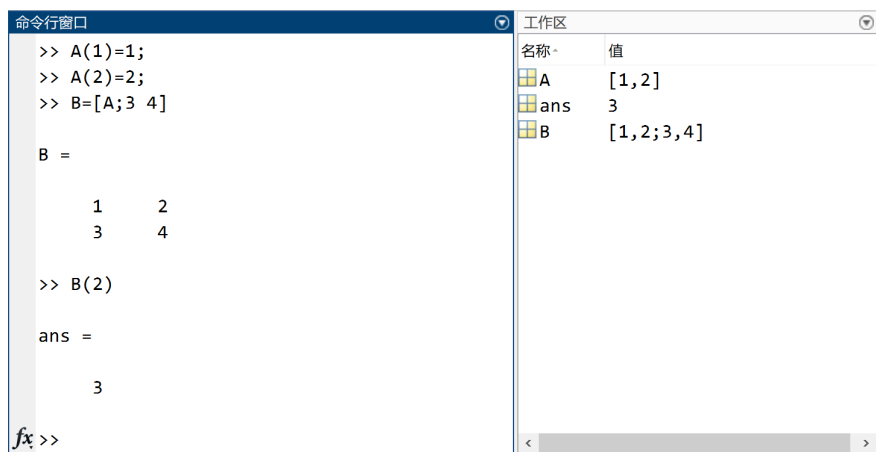


图 5.17 矩阵元素

如果尝试在赋值语句右侧引用数组外部元素，MATLAB 会引发错误。不过，可以在赋值语句左侧指定当前维外部的元素。数组大小会增大以便容纳新元素。

MATLAB 提供了许多生成和操作矩阵的函数。例如：**reshape** 函数用于建立数值矩阵，**diag** 函数用于产生对角阵等。

A(i:j,m:n) 表示由矩阵 **A** 的第 **i** 到第 **j** 行和第 **m** 到第 **n** 列交叉线上的元素组成的子矩阵。可利用冒号提取矩阵的整行或整列。如 **A(1,:)**；**A(:,1:3)**；**A(:,:)**。

5.1.5 基本运算

算术运算

基本算术运算符有：**+**（加）、**-**（减）、*****（乘）、**/**（右除）、****（左除）、**^**（幂运算）等。

一个标量也可以和矩阵进行加减运算，这时把标量和矩阵的每一个元素进行加减运算。

矩阵 **A** 和 **B** 进行乘法运算，要求 **A** 的列数与 **B** 的行数相等，此时则称 **A**、**B** 矩阵是可乘的。在 MATLAB 中，有两种矩阵除法运算：右除/和左除\。如果 **A** 矩阵是非奇异方阵，则 **B/A** 等效于 **B*inv(A)**，**A\B** 等效于 **inv(A)*B**。**inv** 为矩阵求逆函数。

数组运算

数组的运算是指数组对应元素之间的运算，也称点运算。点运算符为 `.*`、`./`、`.\` 和 `.^`。两数组进行点运算是指它们的对应元素进行相关运算，要求两数组同型。图 5.18 为一个例子。

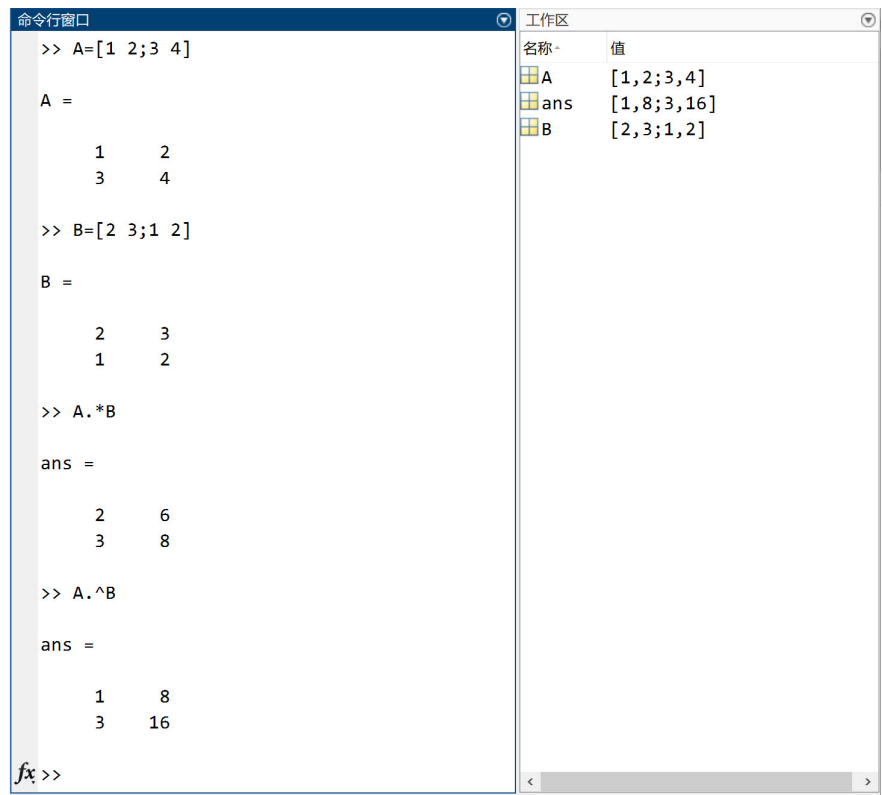


图 5.18 点运算

关系运算

关系运算符有：`<`（小于）、`<=`（小于或等于）、`>`（大于）、`>=`（大于或等于）、`==`（等于）、`~=`（不等于）等。

逻辑运算

逻辑运算符有：`&`（与）、`|`（或）和`~`（非）。设参与逻辑运算的是两个标量 **a** 和 **b**，那么运算规则为

- **a&b**: **a**、**b** 全为非零时，运算结果为 1，否则为 0。
- **a|b**: **a**、**b** 中只要有一个为非零时，运算结果为 1。
- **~a**: 当 **a** 为零时，运算结果为 1；当 **a** 为非零时，运算结果为 0。

基本函数

MATLAB 提供了大量标准初等数学函数, 包括 `abs`、`sqrt`、`exp`、`sin`、`cos`、`log`、`log10` 和 `log2` 等。生成负数的平方根或对数不会导致错误, 系统会自动生成相应的复数结果。

MATLAB 还提供了许多其他高等数学函数, 包括贝塞尔函数和 `gamma` 函数, 其中的大多数函数都接受复数参数。

有关初等数学函数的列表, 请在命令行键入

```
help elfun
```

有关更多高等数学函数和矩阵函数的列表, 请在命令行键入

```
help specfun  
help elmat
```

5.1.6 线图

二维线图

要创建二维线图, 使用 `plot` 函数。例如, 绘制从 0 到 2π 的正弦函数值, 运行以下代码可得到如图 5.19 所示的曲线图。

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

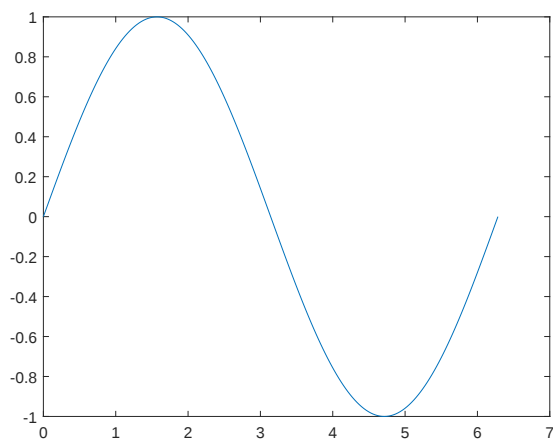


图 5.19 二维线图

坐标区属性

可以标记轴并添加标题，如图 5.20 所示。

```
xlabel('x')  
ylabel('sin(x)')  
title('Plot of the Sine Function')
```

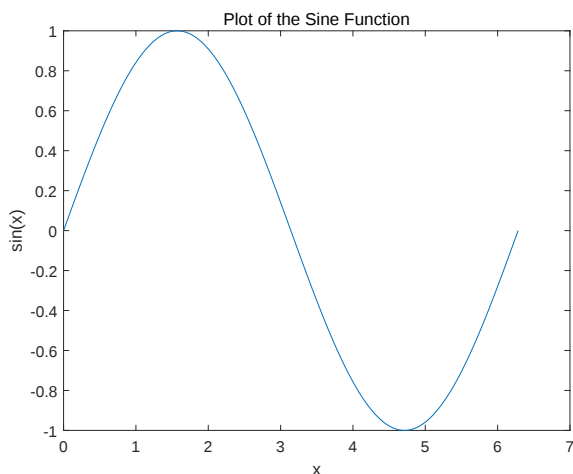


图 5.20 标记坐标轴并添加标题

通过向 `plot` 函数添加第三个输入参数，您可以使用红色虚线绘制相同的变量，如图 5.21 所示。

```
plot(x,y,'r--')
```

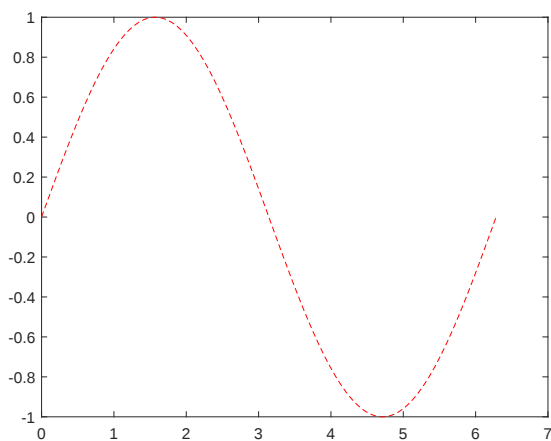


图 5.21 设定线条颜色、样式和标记

'r--' 为线条设定。每个设定可包含表示线条颜色、样式和标记的字符。标记是在绘制的

每个数据点上显示的符号，例如，+、o 或 * 等。例如，'g:*' 请求绘制使用 * 标记的绿色点线。

'linewidth' 为线条宽度设定，结果如图 5.22 所示。

```
plot(x,y,'r--','linewidth',2)
```

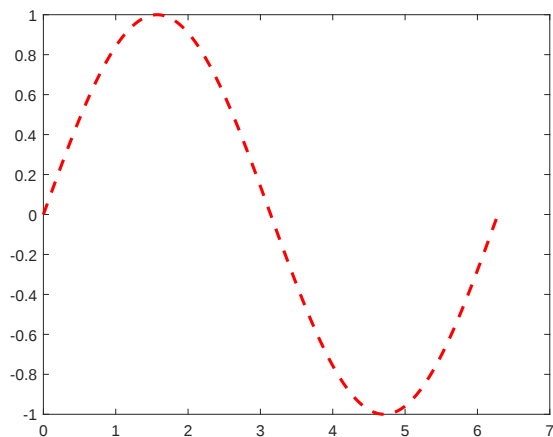


图 5.22 设定线条宽度

可设定坐标区字体大小，如图 5.23 所示。

```
set(gca,'fontsize',15)
```

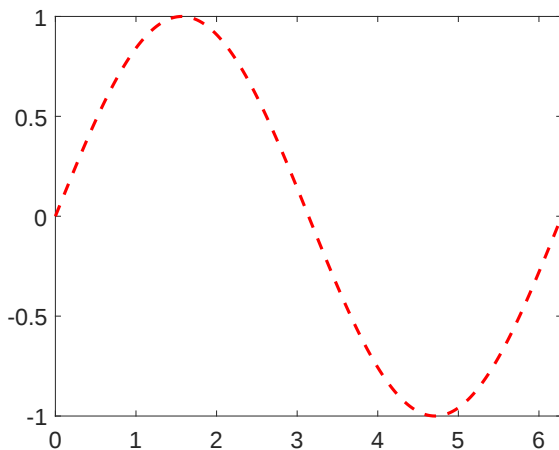


图 5.23 设定坐标区字体大小

添加曲线

请注意，为第一幅绘图定义的标题和标签不再被用于当前的图窗窗口中。默认情况下，每次调用绘图函数、重置坐标区及其他元素以准备新绘图时，MATLAB 都会清空图窗。

要将绘图添加到现有图窗中，请使用 **hold on**。在使用 **hold off** 或关闭窗口之前，当前图窗窗口中会显示所有绘图，如图 5.24 所示。

```
x = 0:pi/100:2*pi;  
y = sin(x);  
plot(x,y)  
hold on  
y2 = cos(x);  
plot(x,y2,':')  
legend('sin','cos')  
hold off
```

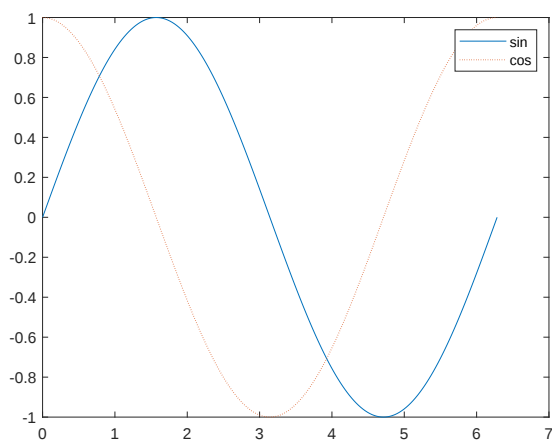


图 5.24 添加曲线

5.1.7 编程和脚本

脚本

脚本是一个包含多行连续的 MATLAB 命令和函数调用的文件，在命令行中键入脚本名称即可运行该脚本。用 MATLAB 语言编写的脚本，文件名后缀为 “.m”，也称为 M 文件。

M 文件有命令文件和函数文件两类。命令文件没有输入参数，也不返回输出参数。函数文件可以输入参数，也可返回输出参数。

单击工具栏的“主页”“新建脚本”，可以建立一个脚本文件。在脚本文件中编写相应的代码，编写完成后单击工具栏的“保存”或利用快捷键 Ctrl+S 保存该文件。注意，一般不建议将文件保存到 MATLAB 的安装目录下。保存脚本文件后，单击工具栏的“编辑器”“运行”即可运行这个脚本文件。

也可使用 **edit** 命令创建脚本。

```
edit mycurves
```

该命令会打开一个名为 `mycurves.m` 的空白文件。编写代码时，最好添加描述代码的注释。注释能够让其他人员理解你的代码，并且有助于你在稍后返回代码时再度记起。使用百分比 (%) 符号添加注释，如图 5.25 所示。

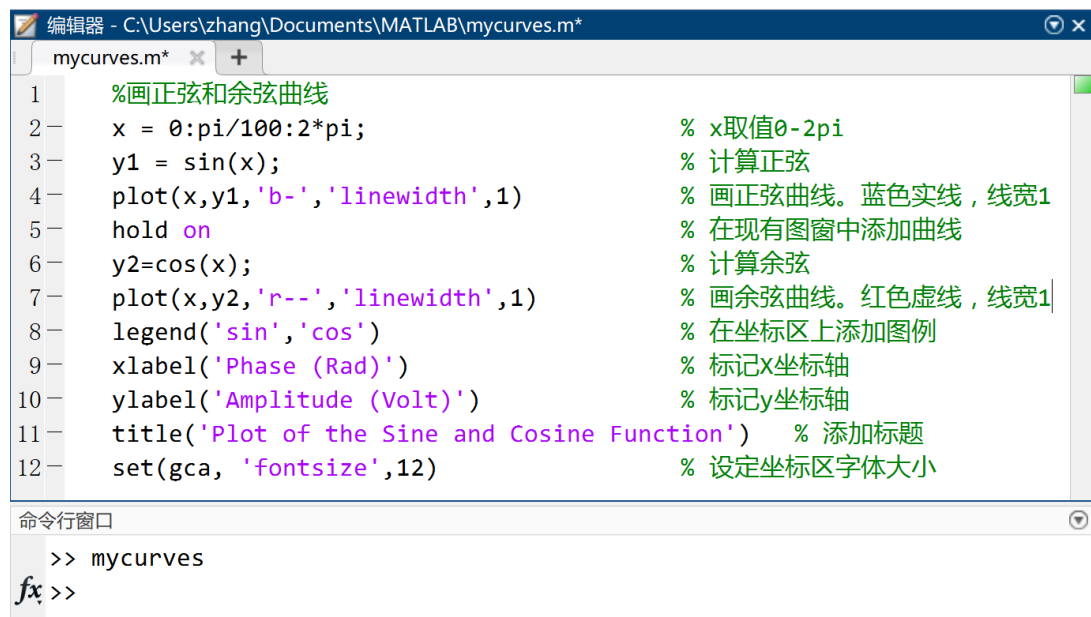


图 5.25 脚本文件

将文件保存在当前文件夹中。要运行脚本，在命令行中键入脚本名称，也可以使用编辑器中的运行按钮运行脚本。输出结果如图 5.26 所示。

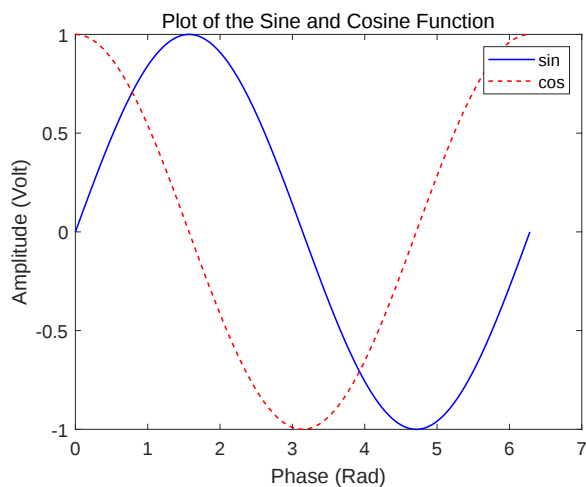


图 5.26 mycurves 输出

循环及条件语句

在任何脚本中,都可以定义按循环重复执行或按条件执行的代码段。循环使用 **for** 或 **while** 关键字, 条件语句使用 **if** 或 **switch**。

for 语句的格式为

```
for 循环变量=表达式1:表达式2:表达式3
    循环体语句
end
```

其中 **表达式 1** 的值为循环变量的初值, **表达式 2** 的值为步长, **表达式 3** 的值为循环变量的终值。步长为 1 时, **表达式 2** 可以省略。

循环在创建序列时很有用。例如, 创建一个名为 **fibseq** 的脚本, 该脚本使用 **for** 循环来计算斐波那契数列 (Fibonacci Sequence, 又称黄金分割数列) 的前 100 个数。在这个序列中, 最开始的两个数是 1, 随后的每个数是前面两个数的和, 即 $F_n = F_{n-1} + F_{n-2}$ 。

```
N = 100;
f(1) = 1;
f(2) = 1;
for n = 3:N
    f(n) = f(n-1) + f(n-2);
end
f(1:10)
```

运行该脚本时, **for** 语句定义一个名为 **n** 的计数器, 该计数器从 3 开始。然后, 该循环重复为 **f(n)** 赋值, **n** 在每次执行中递增, 直至达到 100。脚本中的最后一条命令 **f(1:10)** 显示 **f** 的前 10 个元素。

```
ans =
     1     1     2     3     5     8    13    21    34    55
```

while 语句的格式为

```
while (条件)
    循环体语句
end
```

使用 **for** 实现规定次数的循环, 使用 **while** 实现不确定次数的循环。

条件语句仅在给定表达式为 **true** 时执行。使用 **if...elseif...else...** 实现选择结构, 使用 **switch...case...otherwise...** 实现多分支结构。例如, 根据随机数的大小为变量赋值: 'low'、'medium' 或 'high'。在下面的例中, 随机数是在 1 和 100 之间的一个整数。

```
num = randi(100)
if num < 34
```

```
sz = 'low'
elseif num < 67
    sz = 'medium'
else
    sz = 'high'
end
```

语句 `sz = 'high'` 仅在 `num` 大于或等于 67 时执行。

函数文件

函数文件是另一种形式的 M 文件, 每一个函数文件都定义一个函数。函数文件由 `function` 语句引导, 其格式为

```
function 输出形参表=函数名(输入形参表)
    注释说明部分
    函数体
```

其中函数名的命名规则与变量名相同。输入形参为函数的输入参数, 输出形参为函数的输出参数。当输出形参多于 1 个时, 则应该用方括号括起来。

例: 编写函数文件求小于任意自然数 `n` 的斐波那契数列各项。

```
function f=ffib(n)           %用于求Fibonacci数列的函数文件
    f=[1,1];
    i=1;
    while f(i)+f(i+1)<n
        f(i+2)=f(i)+f(i+1);
        i=i+1;
    end
```

函数调用格式为

```
[输出实参表] = 函数名(输入实参表)
```

全局变量和局部变量

在 MATLAB 中, 全局变量用命令 `global` 定义。函数文件的内部变量是局部的, 与其他函数文件及 MATLAB 工作空间相互隔离。但是, 如果在若干函数中, 都把某一变量定义为全局变量, 那么这些函数将公用这一个变量。全局变量的作用域是整个 MATLAB 工作空间, 即全程有效。

5.1.8 进阶

MATLAB 快速入门

MATLAB 官网上对内置函数、运算符、语法乃至信号处理、图论和微分方程求解等许多问题均有大量的介绍。网址为

https://ww2.mathworks.cn/help/matlab/getting-started-with-matlab.html?s_tid=CRUX_lftnav

在线课程与学习资源

MATLAB 免费在线课程 <https://matlabacademy.mathworks.com/>，使用交互式课程和练习，学习 MATLAB 基础支持。课程时长 2-3 小时、自定进度，通过 Web 浏览器访问，提供自动在线评估和反馈。需要 Campus-Wide License 账户登录（请访问学校内网确认账户注册步骤）。

MATLAB 入门之旅：120 分钟，零基础入门，快速学习 MATLAB 基础知识。

Simulink 入门之旅：180 分钟，快速学习 Simulink 基础支持，需 R2018b 及以上版本支持。

机器学习入门之旅：120 分钟，学习使用机器学习的方法。

深度学习入门之旅：120 分钟，学习使用深度学习的方法。

强化学习入门之旅：120 分钟，学习了解面向控制问题的强化学习方法。

图像处理入门之旅：120 分钟，快速掌握 MATLAB 图像处理入门实践。

信号处理入门之旅：120 分钟，学习面向频谱分析的信号处理实用方法。

Simscape 入门之旅：120 分钟，学习掌握使用 Simscape 仿真物理系统的基础知识。

Stateflow 入门之旅：180 分钟，快速学习在 Stateflow 中创建、编辑和仿真状态机。

Simulink 控制设计入门之旅：120 分钟，学习如何使用 Simulink 进行控制设计。

提升程序运行效率

MATLAB 计算循环的效率非常低下，然而其进行矩阵运算的效率很高，因此为了提高程序的运行效率，应当尽量使用矩阵运算代替循环。

利用线性代数和离散数学的知识，一些需要使用循环的操作可以被转化为矩阵运算，下面给出这种等效的一个例子。

```
a=[1,2,3,4,5,6,7];  
b=ones(1,7);  
c=a*b' % a和b的逆矩阵相乘
```

```
a=[1,2,3,4,5,6,7];  
b=0;  
c=0;  
for b=1:7  
    c=c+a(b);
```

end

两段代码都完成了对 7 维向量 **a** 的各维度求和，但是第一种方法使用矩阵运算进行，第二种方法使用循环进行。在向量 **a** 的维数较小时，这两种方法的效率相差不大；随着维数的增加，第一种方法计算运行效率就会很明显。

在多数情况下，MATLAB 提供的内置函数的执行速度都会比用户自行编写的函数的执行速度更快。在刚才的例子中，我们用较快的矩阵运算替代了较慢的循环累加求和。事实上，MATLAB 提供了求和函数 **sum** 计算矩阵的和，因此我们也可以使用 **sum** 函数替代先前的矩阵运算。

总体而言，在 MATLAB 提供了内置函数的情况下，调用内置函数大概率可以同时降低编程复杂度和时间复杂度。当没有合适的内置函数可供调用时，尽量考虑使用矩阵运算实现需要的功能。当无法找到合适的矩阵运算时，则使用循环、递归等相对低效的方法（但循环和递归也不总是低效的，特别是在分治算法中）。

分节运行

现在考虑这样一个问题：有一段比较长的代码，这段代码从前往后分为 A、B、C、D 四个部分。为了实现功能 1，需要 A 部分和 B 部分；为了实现功能 2，需要 A 部分和 C 部分；为了实现功能 3，需要 A、C、D 这三个部分。为了实现功能 4，A、B、C、D 四个部分需要依次运行。怎样可以较为高效的解决这个问题？

首先，将 A、B、C、D 分别保存为 4 个脚本大概不是个好办法，因为这样做的话，每次运行时都需要切换脚本文件。

其次，利用自定义的子程序或者函数可能会使得代码更简洁些，但是需要考虑参数传递等问题，并且需要对程序进行较复杂的改写。

是否存在一个办法，可以允许我们每次运行部分或全部脚本？

分节运行或许是解决这个问题的一个比较好的办法。在 MATLAB 中，独占一行的连续的两个百分号%% 表示一个分节符。在分节符所在行的上方将出现一条细线，细线将程序分成 2 节或多节。

当光标处于某一节时，单击“编辑器”“运行节”或者利用快捷键 **Ctrl+Enter** 可以运行当前光标所在的节，这就意味着程序将会从本节开始的行运行至下一个分节符出现的位置或是文件末尾。

回到刚才的问题，我们只需要将 A、B、C、D 四个部分用分节符分成 4 节，再根据需要运行对应的节就可以了。

除此以外，分节对提高程序可读性和代码调试都很有帮助。将比较长的程序中相对独立的几个部分分成几节，不仅使程序各段的功能清晰，同时也便于他人阅读。按功能块阅读代码段不仅可以加快阅读代码的速度，同时对理解代码也很有帮助。

在代码调试过程中，错误的定位一般来说会消耗大量的时间。如果将代码按照功能分为几节，就可以依次运行每一节并检查结果，当某节的结果与预期结果不符时，就可以将错误定位在这一节。