
实验一 MATLAB 的基本使用

【一】实验目的

- 1.了解 MATLAB 程序设计语言的基本特点，熟悉 MATLAB 软件的运行环境；
- 2.掌握变量、函数等有关概念，掌握 M 文件的创建、保存、打开的方法，初步具备将一般数学问题转化为对应计算机模型处理的能力；
- 3.掌握二维图形绘制的方法，并能用这些方法实现计算结果的可视化。

【二】MATLAB 的基础知识

一. MATLAB 简介

MATLAB 是由美国 Math Works 公司推出的软件产品。MATLAB 是“Matric Laboratory”的缩写，意及“矩阵实验室”。MATLAB 是一完整的并可扩展的计算机环境，是一种进行科学和工程计算的交互式程序语言。它的基本数据单元是不需要指定维数的矩阵，它可直接用于表达数学的算式和技术概念，而普通的高级语言只能对一个个具体的数据单元进行操作。因此，解决同样的数值计算问题，使用 MATLAB 要比使用 Basic、Fortran 和 C 语言等提高效率许多倍。许多人赞誉它为万能的数学“演算纸”。MATLAB 采用开放式的环境，你可以读到它的算法，并能改变当前的函数或增添你自己编写的函数。在欧美的大学和研究机构中，MATLAB 是一种非常流行的计算机语言，许多重要的学术刊物上发表的论文均是用 MATLAB 来分析计算以及绘制出各种图形。它还是一种有利的教学工具，它在大学的线性代数课程以及其它领域的高一级课程的教学过程中，已成为标准的教学工具。

二. MATLAB 的启动和退出

1.MATLAB 的启动有如下两种方式：

方式一：双击操作系统桌面上的 MATLAB 快捷方式，即可启动并打开 MATLAB 命令窗口。

方式二：单击【开始】菜单，依次指向【程序】→【MATLAB】→【MATLAB 6.5】即可启动并打开 MATLAB 命令窗口。

2. MATLAB 的退出

退出 MATLAB 非常简单，只需在 MATLAB 命令窗口内键入命令 quit 或单击命令窗口的【关闭】按钮即可。

三. MATLAB 使用界面简介

打开 MATLAB 后，缺省的操作桌面如图 1.1 所示。常用的有命令窗、历史命令窗、当前目录浏览器、工作空间浏览器、和交互界面分类目录窗 5 个窗口。如图 1.1 所示。

1. 命令窗（Command Window）

该窗缺省地处在 MATLAB 桌面的右侧，该窗功能是进行各种 MATLAB 操作的主要窗

口。在该窗内，可键入各种送给 MATLAB 运作的指令、函数、表达式；显示除图形外的所有运算结果。

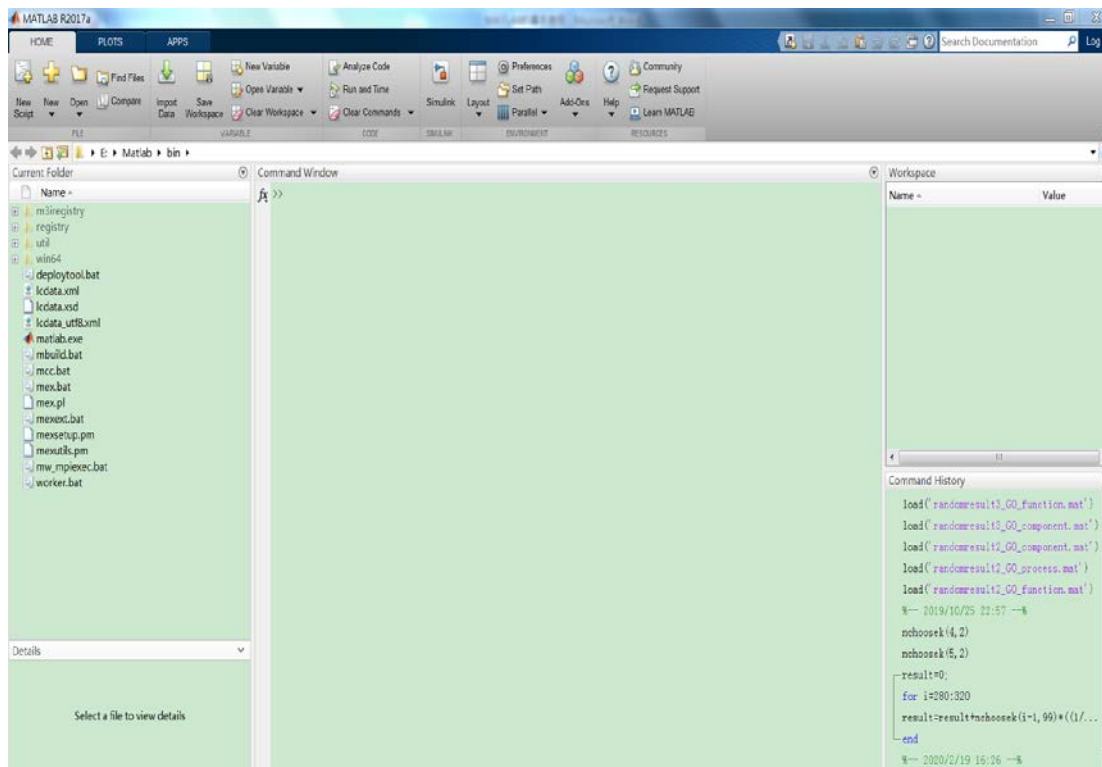


图 1.1 MATLAB 操作桌面的缺省外貌

2. 历史命令窗（Command History）

该窗缺省地处于 MATLAB 桌面的左下方前台。该窗记录已经运作过的指令、函数、表达式；允许用户对它们进行选择复制、重运行、及产生 M 文件。

3. 当前目录浏览器（Current Directory Browser）

该浏览器缺省地位于 MATLAB 桌面左下方的后台。在此交互界面中，可以进行当前目录的设置；展示相应目录上的 M、MDL 等文件；复制、编辑和运行 M 文件；装载 MAT 数据文件。

4. 工作空间浏览器（Workspace Browser）

该交互界面缺省地位于 MATLAB 桌面的左上方后台。该窗口列出 MATLAB 工作空间中所有的变量名、大小、字节数；在该窗中，可对变量进行观察、编辑、提取和保存。

5. 交互界面分类目录窗（Launch Pad）

该交互界面缺省地位于 MATLAB 桌面的左上方前台。该窗以可展开的树状结构列着 MATLAB 提供的所有交互界面，包括：帮助界面、演示界面、各种应用交互界面。用户若双击该树状结构上的分类图标，就可展现出相应的交互界面。

四. 帮助信息的获取

MATLAB 里有以下几种方法可获得帮助：帮助命令（help）、look for 命令、帮助窗口（help

window)、帮助桌面 (help desk)、在线帮助页或直接通过网络连接到 Mathworks 公司。

1. 帮助命令 (help)

帮助命令是查询函数相关信息的最直接方式，信息会直接显示在命令窗口中。如果你知道要找的题材 (topic) 是什么，直接键入 `help <topic>`。

例 1-1 例如在命令窗口键入 `help sin`，会出现如下解释：

```
SIN      Sine.

SIN(X) is the sine of the elements of X.

Overloaded methods

help sym/sin.m
```

2. lookfor 命令

它可以从你键入的关键字 (即使这个关键字并不是 MATLAB 的指令) 列出所有相关的题材，和 `help` 相比，`lookfor` 所能覆盖的范围更宽，可查找到包含在某个主题中的所有词组或短语。

例 1-2 例如在命令窗口键入 `lookfor sin` 会显示：

```
java.m: %Using Java from within MATLAB
syntax.m: %    You can enter MATLAB commands using either a FUNCTION
SUBSINDEX Subscript index.
ISINF    True for infinite elements.
ACOS     Inverse cosine.
ACOSH    Inverse hyperbolic cosine.
ASIN     Inverse sine.
ASINH    Inverse hyperbolic sine.
COS      Cosine.
COSH     Hyperbolic cosine.
SIN      Sine.
.....
```

3. 帮助窗口 (help window)

帮助窗口提供与帮助命令相同的信息，但窗口式的帮助界面更为方便直接。在 MATLAB 命令窗口中有三种方法进入帮助窗口：

- 选取帮助菜单里的 “help window” 项
- 双击菜单条上的问号按钮
- 键入 `helpwin`

4. 帮助桌面 (help desk)

通过在命令窗口中选择帮助菜单的 “Help Desk” 选项后键入 `help desk` 命令即可进入帮助

桌面。这两种方法都会自动启动浏览器，用户可利用浏览器的功能来浏览帮助。

5. 在线帮助页

帮助桌面的在线帮助均有相应的 PDF 格式文件，可以用 Adobe Acrobat Reader 浏览器打开。可以方便的翻页和查找。用户选中帮助桌面的关于 PDF 格式文件的选项，或是在命令窗口键入 doc，便会自动打开文件。

6. Mathworks 网站

对于已连入 Internet 的用户，可通过 HelpDesk 直接访问 Mathworks 公司的网站 www.mathworks.com 询问有关问题。

五.MATLAB 的数值计算功能

1. 变量

和其它高级语言一样，MATLAB 也是使用变量来保存信息。变量由变量名表示，变量的命名应遵循如下规则：

- 变量名必须以字母开头。
- 变量名可以由字母、数字和下划线混合组成。
- 变量名区分字母大小写。
- 变量名的字符长度不应超过 31 个。

在 MATLAB 内部还存在着一些系统默认的特殊变量和常数，如表 1.1

表 1-1 MATLAB 内部的特殊变量和常数

变量或常数	含义
i, j	虚数单位 $\sqrt{-1}$
ans	在没有定义变量名时,系统默认的变量名为 ans
pi	圆周率 π
inf	无穷大
NaN	表示不定值，由 Inf/Inf 或 0/0 产生
why	提供几乎任何问题的简捷回答

MATLAB 的变量分为字符变量和数值变量两种，字符变量必须用单引号括起来。

例 1-3 例如，用户可输入：

a='happy new year'

则表示将字符串 'happy new year' 赋值给字符变量 a。

若用户输入：

`b=365`

则表示将数值 356 赋值给数值变量 `b`。

和其它高级语言不同的是, **MATLAB** 使用变量时不需要先对变量类型进行说明, **MATLAB** 会自动根据所输入的数据来决定变量的数据类型和分配存贮空间。

2. 数值

在 **MATLAB** 内部, 每一个数据元素都是用双精度来表示和存储的, 大约有 16 位有效数字。其数值有效范围约为 $10^{-308} \sim 10^{+308}$ 。

但在其进行数据输入输出时, **MATLAB** 却可以用不同的格式。如果参加运算的每一个元素均为整数, 则 **MATLAB** 将用不加小数点的纯整数格式显示运算结果, 否则, 按默认的输出格式显示结果。**MATLAB** 的默认格式为 `short` 格式, 该格式显示运算结果为保留小数点后 4 位有效数字。用户可以通过 `format` 命令改变输出格式为 `long`, 以得到更多的有效数字 (小数点后 14 位)。需要注意的是, 数据输出格式的改变并不影响该数据在 **MATLAB** 内部的存储精度。设置为 `short` 和 `long` 输出格式的命令分别为:

`format short`

`format long`

MATLAB 通常用十进制数来表示一个数, 亦可用科学计数法来表示一个数。另外, **MATLAB** 还可以进行复数运算, 复数运算可以由如下语句来产生:

`c=a+j*b` (或 `c=a+i*b`) 将实部为 `a` 虚部为 `b` 的复数值赋值给复变量 `c`

`c=a*exp(j*b)` (或 `c=a*exp(i*b)`) 将模为 `a` 幅角为 `b` 的复数赋值给复变量 `c`

3. 矩阵

矩阵是 **MATLAB** 进行数据处理和运算的基本元素, **MATLAB** 的大部分运算或命令都是在矩阵运算的意义下执行的。我们通常意义上的数量 (标量) 在 **MATLAB** 系统中是作为 1×1 的矩阵来处理的, 而仅有一行或一列的矩阵在 **MATLAB** 中称为向量。

1) 矩阵的创建

MATLAB 中矩阵的输入一般有以下几种方法:

①直接输入矩阵元素。直接输入矩阵元素必须注意: 各元素之间用空格或逗号间隔, 用分号(“;”)或回车结束矩阵行, 用中括号 “[]” 把矩阵所有元素括起来。

例1-4 在命令窗口输入如下形式:

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

或 `A=[1 2 3`

`4 5 6`

`7 8 9]`

结果为

```
A =
     1     2     3
     4     5     6
     7     8     9
```

②MATLAB 中有许多特殊矩阵，可直接调用 MATLAB 函数产生。如表 1.2 所示。

表 1.2 MATLAB 中的特殊矩阵

特殊矩阵	调用格式	含义
ones	A=ones(m,n)	生成 $m \times n$ 大小的全 1 矩阵
zeros	A=zeros(m,n)	生成 $m \times n$ 大小的全 0 矩阵
eye	A=eye(m,n)	生成 $m \times n$ 大小的单位阵
rand	A=rand(m,n)	生成 $m \times n$ 大小的随机矩阵

2) 矩阵的下标

在 MATLAB 系统中，矩阵的元素是通过其行、列的标号来标识的，矩阵元素所处的行号和列号称为该元素的下标。矩阵元素可以通过其下标来引用， $A(i,j)$ 即表示矩阵 A 第 i 行第 j 列的元素。

例 1-5 定义矩阵 A 如下所示：

```
A=[1.1  1.2  1.3
    1.4  1.5  1.6
    1.7  1.8  1.9]
```

则 $A(2,3)$ 表示矩阵 A 第 2 行第 3 列的元素 1.6。若输入语句：

```
X=A(1,1)+1(2,2)+A(3,3)
```

则 X 即是矩阵 A 对角线元素之和，运行结果为

```
X= 4.5000
```

★注意：在 MATLAB 中，矩阵下表的行、列号都是从 1 开始的。

3) 利用外部数据文件 (*.mat) 保存和装载矩阵

在运行过程中，有时需要将矩阵数据长期保留下来，以备以后使用，这时就可以使用在 MAT 文件来对矩阵数据进行保存，在需要时又将其装载到 MATLAB 环境中。

MAT 文件(即扩展名为 mat 的文件)是 MATLAB 保存数据的一种标准格式的二进制文件。MAT 文件的生成和调用有专用命令 save 和 load 来进行。

用户可以将已定义过的矩阵(变量)以 MAT 文件的格式存入到磁盘上，命令格式为：

```
save 路径\文件名 变量名
```

save 命令可同时多个矩阵(变量)保存到一个 MAT 文件中，此时变量名之间需用空格分开。例如，用户输入命令：

```
save c:\my a b
```

则表示将变量a、b 以文件名my.mat保存到C盘根目录下。若路径默认，则MATLAB自动将变量保存到你默认的目录（work）中。

以MAT文件保存的矩阵，用户在使用MATLAB的任何时候均可用load命令装载到MATLAB的工作空间中。例如，要将上述保存的变量a、b重新装载到MATLAB的工作空间，只需键入命令：

```
load c:\my
```

4) 矩阵和数组的运算

在 MATLAB 中，数组也是一个非常重要的概念，矩阵在某些情况下可视为二阶的数值型数组。但是在 MATLAB 中，数组和矩阵运算规则却有着较大的区别。例如，两矩阵相乘和两数组相乘所遵循的运算规则就是完全不同的。具体运算规则如表 1.3 所示。

表 1.3 MATLAB 中关于矩阵和数组的算术运算符

运算符	名 称	指令示例	说 明
+	加	$A+B$	若 A、B 为同维矩阵，则表示 A 与 B 对应元素相加；若其中一个矩阵为标量，则表示另一矩阵的所有元素加上该标量
-	减	$A-B$	若 A、B 为同维矩阵，则表示 A 与 B 对应元素相减；若其中一个矩阵为标量，则表示另一矩阵的所有元素减去该标量
*	矩阵乘	$A*B$	矩阵 A 与 B 相乘，A 和 B 均可可是向量或标量，但 A 和 B 的维数必须符合矩阵乘法的定义
.*	数组乘	$A.*B$	矩阵 A 与 B 对应元素相乘，A 与 B 必须为同维矩阵或其中之一为标量
^	矩阵乘方	A^B	A、B 均为标量时，表示 A 的 B 次方幂；A 为方阵，B 为正整数时，表示矩阵 A 的 B 次乘积；A 为方阵，B 为负整数时，表示矩阵 A 逆矩阵的 B 次乘积；当 A、B 均为矩阵时无定义
.^	数组乘方	$A.^B$	矩阵 A 的各元素与矩阵 B 的对应元素的乘方运算，即 $[A(i,j)^B(i,j)]$ ，A、B 必须为同维矩阵
\	矩阵左除	$A\backslash B$	方程 $A*X=B$ 的解 X
.\	数组左除	$A.\backslash B$	矩阵 B 的各元素与矩阵 A 的对应元素，A 与 B 必须为同维矩阵或其中之一为标量
/	矩阵右除	A/B	方程 $X*B=A$ 的解 X
./	数组右除	$A./B$	矩阵 A 的各元素与矩阵 B 的对应元素，A 与 B 必须为同维矩阵或其中之一为标量
'	共轭转置	A'	矩阵 A 的共轭转置

另外，在 MATLAB 内部有一些特殊的函数用于对矩阵进行运算，如下所示。

●size()函数

size()函数用于计算矩阵的行数和列数，其调用格式为：

`d=size(a)` 将矩阵 `a` 的行数和列数赋值给变量 `d`

`[m,n]=size(a)` 将矩阵 `a` 的行数赋值给 `m`、列数赋值给 `n`

●`length()`函数

`length()`函数用于计算矩阵的长度（列数），其调用格式为：

`a=length(b)` 将矩阵 `b` 的列数赋值给变量 `a`

●`sum()`函数

`sum()`函数用于实现矩阵元素的求和运算，其调用格式为：

`sum(a)` 若 `a` 为向量，则该调用格式将计算出向量 `a` 所有元素之和。若 `a` 为矩阵，则该调用格式将产生一行向量，其元素分别为矩阵 `a` 的各列元素之和。

●`max()`函数

`max()`函数用于求出矩阵元素的最大值，其调用格式为：

`max(a)` 若 `a` 为向量，则该调用格式求出向量 `a` 所有元素的最大值。若 `a` 为矩阵，则该调用格式将产生一行向量，其元素分别为矩阵 `a` 的各列元素的最大值。

4.运算符

MATLAB 的基本运算为算术运算、关系运算、逻辑运算和特殊运算等，每一类运算都有自己专用的运算符。关于矩阵和数组的算术运算符已在表 1.3 中列出，如下表 1.4、表 1.5、表 1.6 分别列出了 MATLAB 的关系运算符、逻辑运算符、特殊运算符和其对应的功能与示例。

表 1.4 MATLAB 的关系运算符表

运算符	名称
<code>==</code>	等于
<code>~=</code>	不等于
<code>></code>	大于
<code><</code>	小于
<code>>=</code>	大于等于
<code><=</code>	小于等于

表 1.5 MATLAB 的逻辑运算符

运算符	名称
<code>&</code>	逻辑与
<code> </code>	逻辑或
<code>~</code>	逻辑非

表 1.6 MATLAB 的特殊运算符

运算符	名称	说明
<code>:</code>	冒号	用于产生向量
<code>[]</code>	方括号	用于创建和表示矩阵
<code>;</code>	分号	用于分隔矩阵的行或不显示中间结果
<code>%</code>	百分号	用于注释语句
<code>,</code>	逗号	用于分隔矩阵列
<code>=</code>	等号	用于赋值

()	圆括号	用于函数调用和指定运算顺序
-----	-----	---------------

六.程序流程控制

MATLAB 为用户提供了丰富的程序结构语句用来实现用户对程序流程的控制。

1. 循环控制语句

当程序段中的某段指令需要根据一定能够的条件多次重复执行时，就需要用到循环控制。

在 MATLAB 中，循环控制由 for 语句和 while 语句实现。

1) for 循环语句

for 循环语句的格式为：

for 变量=表达式

语句组

end

在上述格式中，end 是必需的，不可默认。表达式是一个矩阵，语句组则是一组合法的 MATLAB 命令。

for 循环语句则执行过程是：从表达式的第一列开始，依次将表达式（矩阵）的各列之值赋值给变量，然后执行语句组中的语句，直到最后一列。

for 语句的典型格式是表达式为冒号运算的行向量，即：

for i=m:p:n

语句组

end

其中 m 为循环起始值，n 为循环终止值，p 为步长值。

例 1-6. 我们用 for 语句来实现求和运算 $s=1+3+5+7+\cdots+99$ ，对应的 MATLAB 命令如下：

s=0;

for i=1:2:99

s=s+i;

end

s

运行结果为： s=

2500

在上述格式中，步长 p 可以默认，系统将默认步长为 1。

例 1-7. 我们可用如下语句来实现求和运算 $s=1+2+3+4+\cdots+100$

s=0;

for i=1:100

s=s+i;

end

s

运行结果为:

s=

5050

for 语句也可实行多重循环的嵌套, 其格式为:

```
for 变量 1=表达式 1
    for 变量 2=表达式 2
        语句组
    end
end
```

★注意: 在用 for 语句实现多重循环时, for 和 end 必须成对出现。

2) while 语句

while 语句用来实现在某一逻辑关系控制下的循环。while 语句的格式为:

```
while 关系表达式
    语句组
end
```

在 while 语句中, end 也是必需的, 不可默认。while 语句的执行过程是: 首先判断关系表达式是否成立, 如果成立则运行语句组中的语句, 否则停止循环。通常是通过在语句中对关系表达式进行改变来控制循环是否结束。

例 1-8. 我们用 while 语句来实现下列级数求和: $s = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{100}$

命令如下:

```
s=0;
n=1;
while n<=100
    s=s+1/n;
    n=n+1;
end
s
```

运行结果为:

s=

5.1874

2. 条件控制语句

和其他高级程序语言一样, MATLAB 也为用户提供了方便的条件控制语句, 用以实现程

序的条件分支运行。实现条件控制的 MATLAB 命令是 if 语句和 switch 语句。

1) if 语句

if 语句有两种格式。当分支条件之有两种情况时，可采用 if 语句的第一种格式，即

```
if 表达式
    语句组 1
else
    语句组 2
end
```

上述 if 语句的运行结果是，如果表达式成立，则运行语句组 1，若表达式不成立，则运行语句组 2。

当程序运行的分支条件多于两个时，则可采用 if 语句的第二种格式，即：

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

上述 if 语句的运行结果是，若果表达式 1 成立，则运行语句组 1；若表达式 2 成立，则运行语句组 2；否则，则运行语句组 3。

2) switch 语句

当程序运行过程中需要根据某个变量的多种不同取值情况来运行不同的语句时，就要用到 switch 语句。

switch 语句适合于多个条件的分支程序，其基本格式为：

```
switch 控制变量
    case 变量 1
        语句组 1
    case 变量 2
        语句组 2
    case 变量 3
        语句组 3
    ...
    otherwise
        语句组 n
```

end

在上述格式中，switch 语句通过判断控制变量的取值情况来决定运行哪一个语句组，即当控制变量的值为变量值 1 时，则运行语句组 1；当控制变量的值为变量值 2 时，则运行语句组 2，依此类推。若所有条件均不满足，则运行 otherwise 后的语句组。

注意，在 switch 语句中，end 是必需的，不可默认。

七. M 文件

MATLAB 为用户提供了专用的 M 文件编辑器，用业帮助用户完成 M 文件的创建、保存及编辑等工作。

1. 创建新 M 文件

利用 M 文件编辑器创建新 M 文件有如下两种方法：

方法一：启动 MATLAB，选中命令窗口菜单栏【File】菜单下【New】菜单选项的【M-File】命令，打开 MATLAB 的 M 文件编辑窗口。

方法二：单击 MATLAB 命令窗口工具栏的“New M-File”图标按钮，也可打开 M 文件编辑器。

2. 保存 M 文件

当用户的 M 文件已输入或编辑完毕后，则可将 M 文件保存到磁盘上。保存 M 文件的方法是，单击 M 文件编辑器窗口的工具栏中的“Save”图标按钮或选中 M 文件编辑器窗口菜单栏【File】菜单的【Save】命令，打开 M 文件保存对话框。

如果是新建 M 文件，则系统默认的文件名为“Untitled1.m”，用户可以对要保存的文件进行重新命名。系统默认的文件保存目录为“work”，用户也可以通过保存对话框对文件保存位置进行更改和设置。当保存的文件名和保存位置设定完成后，按下【保存】按钮，即可将 M 文件保存到指定位置。

3. 打开 M 文件

若需要对已保存过的 M 文件进行修改和编辑，则可单击 MATLAB 命令窗口工具栏的“Open file”图标按钮或选中命令窗口菜单栏【File】菜单下的【Open】命令，系统即启动 M 文件编辑器并打开用户指定的 M 文件。

4. 搜索与执行 M 文件

M 文件中的命令是通过在 MATLAB 命令窗口中键入 M 文件的文件名并按下回车键来执行的。当用户在命令窗口中键入 M 文件的文件名并按下回车键后，系统先搜索该文件，若该文件存在，则以解释方式按顺序逐条执行 M 文件语句。

“work”目录是系统 M 文件的默认保存目录。若用户的 M 文件保存在“work”目录以外的其他位置，则可通过如下两种方式在 MATLAB 命令窗口中直接调用该 M 文件。

方式一：启动 MATLAB 后，用“CD”命令将当前工作目录更改为 M 文件的保存目录，如：
cd A:\mydata

方式二：用“path”命令将 M 文件的保存目录添加到 MATLAB 的默认搜索路径中。设待执行的 M 文件的保存位置为“D:\myfile”，则添加的搜索路径的命令为：

```
path(path,'D:\myfile')
```

运行该命令后，即可直接在 MATLAB 命令窗口中直接调用并执行 D:\myfile 目录下的所有 M 文件。

八. 函数文件

函数文件是 M 文件的另一种类型，它也是由 MATLAB 语句构成的 ASCII 码文本文件，扩展名为 m。用户可用前述的 M 文件的创建、保存及编辑的方法来进行函数文件的创建、保存与编辑，但特别需要注意以下几点：

函数文件必须以关键字“function”开头。

函数文件的第 1 行为函数说明语句，其格式为：

```
function [返回参数 1, 返回参数 2, ...]=函数名(传入参数 1, 传入参数 2, ...)
```

其中函数名为用户自己定义的函数名（与变量的命名规则相同）。

函数文件保存的文件名应与用户定义的函数名一致，例如，若函数文件说明语句中定义的函数名为“example”，则该函数文件保存的文件名为“example.m”。

用户可通过函数说明语句中的返回参数及传入参数来实现函数参数传递。返回参数和传入参数并不是必需的。下面是函数文件调用及参数传递的例子。

例 1-9. 首先创建如下所示的函数文件并保存。

```
function [s,m]=pjz(a) %定义函数文件 pjz.m, a 为传入参数, m、s 为返回参数
l=length(a); %计算传入向量长度
s=sum(a); %对传入向量 a 求和并赋值给 s
m=s/l; %计算传入向量的平均值并赋值给返回向量 m
```

上述函数文件定义了 1 个新的函数 pjz，其作用是对指定向量求和及平均值，并通过向量 s、m 返回计算结果。用户可通过如下所示的命令调用该函数：

```
a=1:9;
```

```
[s,m]=pjz(a)
```

运行结果为：

```
s =
    45
m =
     5
```

九. MATLAB 的可视化

MATLAB 在数据可视化方面的表现力很强。它的图形处理能力不仅功能强大，而且充分考虑了高低不同层次用户的不同需求，系统具有两个层次的绘图指令：一个层次是直接对图

形句柄进行操作的底层绘图指令，它具有控制和表现数据图形能力强，控制灵活多变等优点，对于有较高和特殊需求的用户而言，该层次能够完全满足他们的需求；另一层次是在底层指令基础上建立的高层绘图指令，它的指令简单明了，易于掌握，适用于普通用户。

常用的 MATLAB 绘图语句有 figure、plot、subplot、stem，等，图形修饰语句有 title、axis、text 等。

1.figure 语句

figure 语句有两种用法，只用一句 figure 命令，会创建一个新的图形窗口，并返回一个整数型的窗口编号。figure(n)表示将第 n 号图形窗口作为当前的图形窗口，并将其显示在所有窗口最前面；如果该图形窗口不存在，则新建一个窗口，并辅以编号 n。

2. plot 语句

线型绘图函数，将点和点之间用直线段连接。用法为 plot(x,y,'s')，参数 x 为横轴变量，y 为纵轴变量，s 用以控制图形的基本特征如颜色、粗细，，通常可以省略。常用方法如表 1.7 所示。

表 1.7 plot 命令的参数及其含义

参数	含义	参数	含义	参数	含义
y	黄色	.	点	—	实线
m	紫色	o	圆	:	虚线
c	青色	×	打叉	-.	点划线
r	红色	+	加号	--	破折线
g	绿色	*	星号	^	向上的三角形
b	蓝色	s	正方形	<	向左的三角形
w	白色	d	菱形	>	向右的三角形
k	黑色	v	向下的三角形	p	五角星形

3. stem 语句

绘制离散序列图，常用格式 stem(y)和 stem(x,y)分别和相应的 plot 的函数绘图规则相同，只是用 stem 命令绘制的是离散序列图。

4. subplot 语句

subplot(m,n,i) 图形显示时分割窗口命令，把一个图形窗口分为 m 行，n 列，m× n 个小窗口，并指定第 i 个小窗口为当前窗口。

5. 绘图修饰命令

在绘制图形时，我们通常需要为图形添加各种标注以增加可读性。在 plot 语句后使用 title（‘标题’）可以在图形上方添加标题；使用 xlabel（‘标记’）或 ylabel（‘标记’）为 X 轴和 Y 轴添加说明；使用 text(X 值、Y 值、‘想加的标示’）可以在图形中任意位置添加标示；使用 axis([xmin,xmax,ymin,ymax])设置坐标轴的最小最大值；gtext（‘string’）利用鼠标添加说明性文本

信息：grid on(off)给当前图形标记添加（取消）网格。

例 1-10 用 MATLAB 语句绘正弦图，其运行结果见图 1.2

```
x=0:0.01*pi:2*pi;    %定义 x 向量
figure(1);            %创建一个新的图形窗口,编号为 1
subplot(2,2,1);        %将窗口划分为 2 行 2 列,在第 1 个窗口中作图
plot(x,sin(x));        %画图
title('正弦线');       %给图形加标题
subplot(2,2,2);        %在第 2 个窗口中作图
plot(x,sin(x),'r');     %画一正弦波,红色
xlabel('X');            %给 x 轴加说明
ylabel('SIN(X)');      %给 y 轴加说明
subplot(2,2,3);        %在第 3 个窗口中作图
plot(x,sin(x),'--');    %画一正弦波,破折线
subplot(2,2,4);        %在第 4 个窗口中作图
plot(x,sin(x),'r+');    %画一正弦波,红色加号
text(4,0,'注记');      %在位置(4,0)处加一注记
```

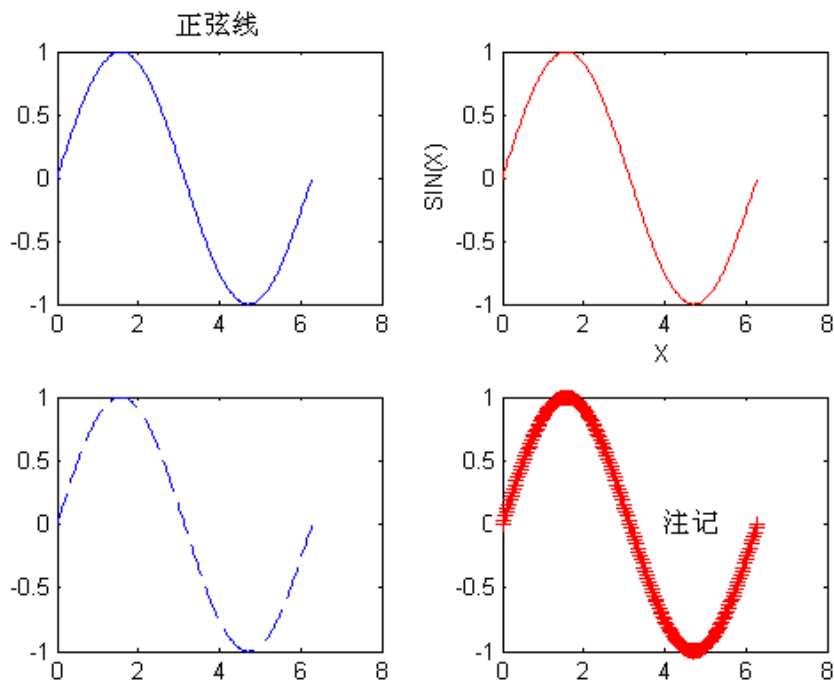


图 1.2 例 1-10 的运行结果

6. 添加图例函数 legend()

基本格式：legend('string1','string2',...)

legend(...,pos)

功能：给图加图例，对于每个已经画出的曲线，可以配置想要的图例。

参数: 'string1','string2',...——解释图例的字符串

pos —— 决定放置图例的位置

pos= -1 将图例放在轴边距外右边

pos= 0 将图例放在轴边距内右边占用尽量少的点

pos= 1 将图例放在轴的右上方（缺省值）

pos= 2 将图例放在轴的左上方

pos= 3 将图例放在轴的左下方

pos= 4 将图例放在轴的右下方

例1-11. legend 用法举例

```
x=-pi:pi/20:pi;
```

```
plot(x,cos(x),'-ro',x,sin(x),'-b')
```

```
h=legend('cosx','sinx','Location','NW')
```

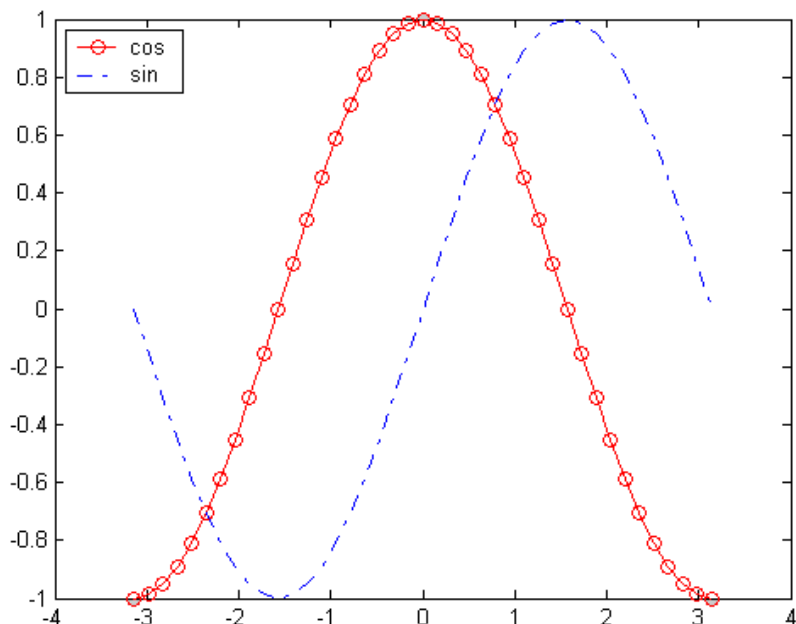


图 1.3 例 1-11 的运行结果

【三】上机练习

1. 仔细预习第二部分内容，关于 MATLAB 的基础知识。
2. 熟悉 MATLAB 环境，将第二部分所有的例子在计算机上练习一遍

3. 已知矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, $B = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$ 。求 $A*B$, $A.*B$, 比较二者结

果是否相同。并利用 MATLAB 的内部函数求矩阵 A 的大小、元素和、长度以及最大值。

-
4. Fibonacci 数组的元素满足 Fibonacci 规则: $a_{k+2} = a_k + a_{k+1}, (k = 1, 2, \dots)$; 且 $a_1 = a_2 = 1$ 。现要求该数组中第一个大于 10000 的元素。

- 1) 在命令窗口中完成;
- 2) 利用 M 文件完成;
- 3) 自己定义一个函数文件, 并在命令窗口中调用该函数完成。

5. 在同一个图形窗口的两个子窗口中分别画出 $\cos(\frac{\pi}{4}x)$ (红色、虚线) 和 $\cos(\frac{\pi}{8}x)$

(蓝色、星号) 的波形。要求有标题, x、y 轴有标注。

【四】思考题

1. 在语句末加分号 “;” 和不加分号有什么区别?
2. M 文件和函数文件有什么异同之处?
3. 矩阵乘 (*) 和数组乘 (.*) 有何不同?

【五】实验报告要求

1. 简述实验目的;
2. 实验内容及结果分析:
 - 1) 附上源程序清单, 要求可读性好, 必要处要加注释;
 - 2) 实验结果, 包括运行的数值结果或图形;
 - 3) 结果分析, 正确与否, 误差原因。
3. 简要回答思考题;
4. 简述本次实验的体会和建议。

