

第一篇 准备实验

MATLAB 软件操作

0.1 MATLAB 软件的启动

启动 MATLAB 以后，就进入 MATLAB 的桌面。图 0.1 为 MATLAB R2013b 的默认桌面。第一行为菜单行，第二行为工具栏。下面是三个最常用窗口：右边最大的是指令窗口（Command Window），左上方前台为工作空间（Workspace），后台为当前目录（Current Directory），左下方为指令历史（Command History），左下角还有一个开始（Start）按钮，用于快速启动演示（Demo）、帮助（Help）和桌面工具等。

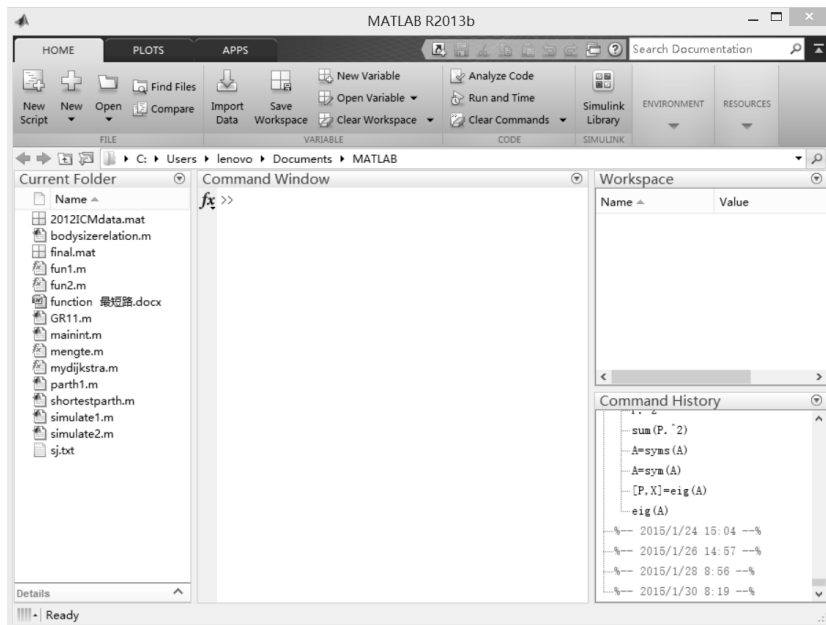


图 0.1 MATLAB R2013b 的默认桌面

0.1.1 窗口

窗 口	功 能
指令窗口 (Command Window)	MATLAB 进行操作的主要窗口, 窗口中的 >> 为指令输入的提示符, 其后输入指令, 按回车 (Enter) 键就执行运算, 并输出运算结果
工作空间 (Workspace)	列出内存中 MATLAB 工作空间的所有变量的变量名 (Name)、值 (Value)、尺寸 (Size)、字节数 (Byte) 和类型 (Class)
当前目录 (Current Directory)	用鼠标单击可以切换到前台。看到该窗口列出当前目录的程序文件 (.m) 和数据文件 (.mat)
指令历史 (Command History)	窗口列出在指令窗口执行过的 MATLAB 指令行的记录

0.1.2 菜单和工具栏

菜单和工具栏类似于 Word 软件。

菜单/工具栏	使用 说明
File: New: M-file	新建 M 文件
File: Import data	导入数据文件 (Mat 文件)
File: Save workspace as	将工作空间所有变量和数据保存为数据 Mat 文件
File: Set path	设置 MATLAB 文件搜索路径
File: Preference	设置 MATLAB 选项, 如数据显示格式, 字体等
Desktop: Desktop Layout	窗口布局选择, 一般使用默认 (Default)
Current Directory	设置 MATLAB 当前目录

【例 1】 在命令窗口输入:

```
a=1;b=2; c=a+b*pi
```

输出:

```
c =  
7.2832
```

可以看到在工作空间 (Workspace) 和指令历史 (Command History) 窗口, 对变量 a, b, c 和运算指令都有相应的记录。

0.2 MATLAB 常用命令、符号

0.2.1 命令窗口中的常用命令

指 令	含 义
clf	清除图形窗口
clc	清除命令窗中显示内容
clear	清除 MATLAB 工作内存中的变量

(续表)

指 令	含 义
who	列出 MATLAB 工作内存中驻留的变量名清单
whos	列出 MATLAB 工作内存中驻留的变量名清单以及属性
help	帮助命令
edit	打开 M 文件编辑器
↑(↓)	向前(后)调出已输入过的指令
format	定义输出格式(默认值), 等效于 format short
format short	输出用中小数点后面 4 位有效数字
format long	输出用 15 位数字表示
format short e	输出用 5 位科学计数法表示
format long e	输出用 15 位科学计数法表示
format rat	输出用近似的有理数表示
format compact	显示变量之间不加空行(紧凑格式)
format loose	显示变量之间加空行
demo	浏览 MATLAB 软件基本功能
funtool	打开函数简单操作的可视化交互界面, 显示三个可操作图形窗口(详见图 0.2)
Taylortool	打开可视化函数图形器, 观察不同次的泰勒多项式逼近函数的状态(详见实验九)

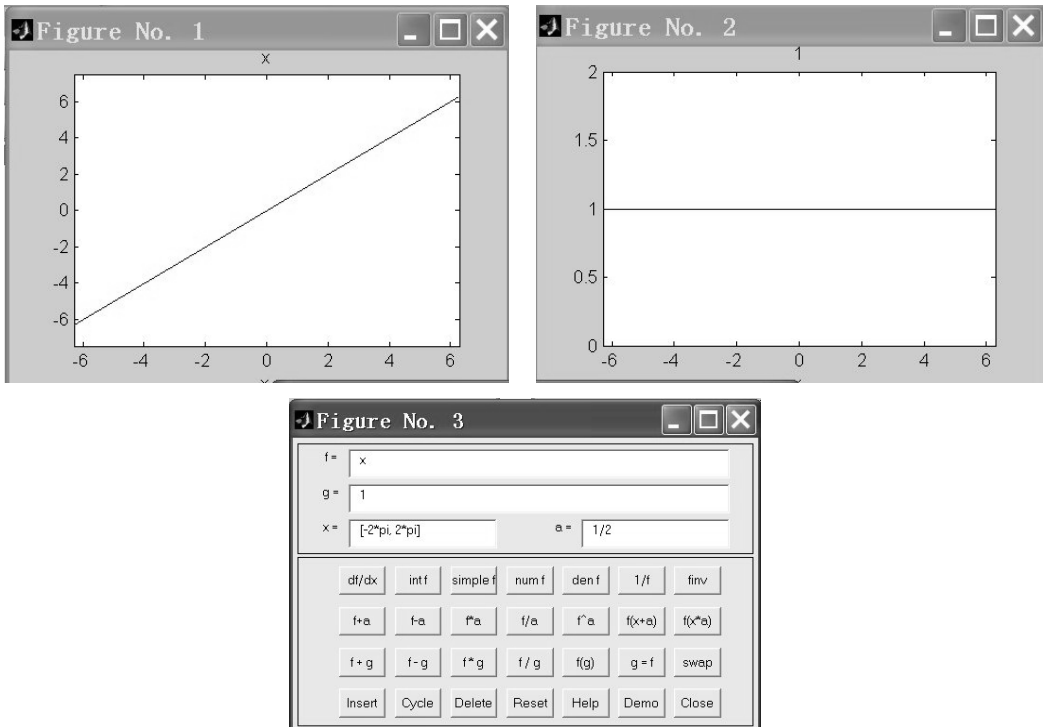


图 0.2 函数可视化交互界面

0.2.2 变量命名规则

变量名的第一个字符必须是英文字母，最多包含 31 个字符（包括英文字母、数字和下划线），变量中不得包含空格和标点符号，不得含有加减号。变量名和函数名区别字母的大小写。如 `matrix` 和 `Matrix` 表示两个不同的变量。要防止它与系统的预定义变量名（如 `I`, `j`, `pi`, `eps` 等）、函数名（如 `who`, `length` 等）、保留字（如 `for`, `if`, `while`, `end` 等）冲突。

变量赋值用 `=`（等于号）。

有一些变量永久驻留在工作内存中，不能再重新赋值。这些变量见下表。

变 量	含 义
<code>ans</code>	计算结果的默认变量名
<code>pi</code>	圆周率
<code>inf</code> 或 <code>Inf</code>	无穷大
<code>eps</code>	机器零阈值
<code>Flops</code>	浮点运算次数
<code>NaN</code> 或 <code>nan</code>	不是数 (Not a Number)

0.2.3 基本运算符及特殊符号

算术运算符

运 算 符	含 义
<code>+</code>	加
<code>-</code>	减
<code>*</code>	数之间乘，矩阵乘
<code>^</code>	数的乘幂，矩阵的幂
<code>\</code>	数的左除， <code>2\1</code> 得 0.5000，矩阵左除（见实验十三）
<code>/</code>	数的右除， <code>2/1</code> 得 2，矩阵右除

关系运算与逻辑运算

运 算 符	含 义
<code>==</code>	相等
<code>~=</code>	不等
<code><</code>	小于
<code>></code>	大于
<code><=</code>	小于等于
<code>>=</code>	大于等于
<code>&</code>	与
<code> </code>	或
<code>~</code>	非

0.2.4 命令行中的特殊符号

名 称	符 号	含 义
等号	=	赋值
空格		输入量与输入量之间的分隔符； 数组元素分隔符
逗号	,	输入量与输入量之间的分隔符； 数组元素分隔符
句点	.	数值运算中的小数点； 结构域的存取
分号	;	不显示计算结果命令的结尾标志； 数组行与行之间的分隔符
冒号	:	生成一维数值数组； 单下标索引时，表示全部元素构成的长列； 多下标索引时，表示所在维上的全部元素
注释号	%	在它后面的文字、命令等不被执行，用于注释
单引号对	''	字符串标记符
单撇号	'	矩阵转置
方括号	[]	输入数组标记符
圆括号	()	用于紧随函数名后； 用于运算式中的结合与次序
续行号	...	用于长表达式的续行

0.2.5 基本初等函数的表示

函 数 类 别	函 数 名 称
三角函数	sin(x) cos(x) tan(x) cot(x) sec(x) csc(x)
反三角函数	asin(x) acos(x) atan(x) acot(x) asec(x) acsc(x)
双曲函数	sinh(x) cosh(x) tanh(x) coth(x) sech(x) csch(x)
反双曲函数	asinh(x) acosh(x) atanh(x)
x 的平方根	sqrt(x)
以 e 为底的 x 指数	exp(x)
以 e 为底的 x 的对数	log(x)
以 10 为底的 x 的对数	log10(x)

0.2.6 几个特殊的函数

函 数 名	含 义
round(x)	四舍五入函数，取最接近 x 的整数
floor(x)	取整函数，取不超过 x 的最大整数
ceil(x)	取整函数，取不小于 x 的最小整数
Fix(x)	取 x 的整数部分
abs(x)	取 x 的绝对值（模）

(续表)

函 数 名	含 义
real(z)	取复数 z 的实部
imag(z)	取复数 z 的虚部
angle(z)	取复数 z 的辐角
conj(z)	取复数 z 的共轭
mod(m,n)	输出 m 除以 n 的余数

【例 2】 输入：

```
x=1.2; y=exp(x)*sin(x)^2 , z=fix(y)
```

执行后输出：

```
y =  
2.8842  
z =  
2
```

0.3 数组及其运算

0.3.1 数组的输入与分析

最常用的数组是双精度数值数组。一维数组是向量，二维数组是矩阵。一维数组可以视为二维数组的特例。

输入、分析各种数组的技巧可以从下面的例子中学习。例如

1. 输入：

```
clear; a=[1,2,3;4,5,6;7,8,9] %输入 3 行 3 列矩阵 a, 逗号可以用空格代替
```

输出：

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

2. 输入：

```
b=0:3:10 %构造一维等差数组，初值：增量：终值
```

输出：

```
b =  
0 3 6 9
```

3. 输入:

```
b=0:5 %增量 1 可以省略
```

输出:

```
b =  
    0    1    2    3    4    5
```

4. 输入:

```
b=10:-3:0 %输入递减等差数列
```

输出:

```
b =  
   10    7    4    1
```

5. 输入:

```
b=linspace(0,10,4) %将区间[0,10]等分为 4-1=3 份
```

输出:

```
b =  
    0    3.3333    6.6667   10.0000
```

6. 输入:

```
length(b) %查询一维数组 b 的长度 (元素的个数)
```

输出:

```
ans =  
     4
```

7. 输入:

```
size(a) %查询二维数组 a 的尺寸, a 是 3 行 3 列的矩阵
```

输出:

```
ans =  
     3     3
```

8. 输入:

```
b(3) %查询、提取 b 的第 3 个元素
```

输出:

```
ans =  
    6.6667
```

9. 输入:

```
a(3,2) %查询、提取 a 的第 3 行, 第 2 列的元素
```

输出:

```
ans =  
8
```

10. 输入:

```
a(:) %将 a 的所有元素按单下标顺序排列为列向量, 注意排列方式  
是先第 1 列, 再第 2 列……, 与通常的认为相反
```

输出:

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

11. 输入

```
a(4) %查询、提取将 a 的所有元素按单下标顺序排列后的第 4 个元素,
```

输出:

```
ans =  
2
```

数组的部分元素可以按其编址提取和拼接。例如输入

```
b([1,end]) %提取 b 的首和尾元素  
c=a([1 3],[2 3]) %提取 a 的第 1, 3 行, 第 2, 3 列  
d=a(2,1:3) %提取 a 的第 2 行的 1 至 3 列  
d1=a(2,:) %提取 a 的整个第 2 行  
e=[a;d1] %数组 a 与数组 d 拼接  
e(3,4)=15 %修改 e 的 3 行 4 列元素的值为 15,e 的其余元素不变
```

请输入以上命令, 自己观察输出结果, 体会命令的功能。

0.3.2 数组的运算

运算符号	功能
+	数组加, A+B 为 A,B 两个数组对应元素相加
-	数组减, A-B 为 A,B 两个数组对应元素相减
*	数组乘, A.*B 为 A,B 两个数组对应元素相乘
^	数组的幂, A.^2 为数组 A 的每个元素平方; A.^B 为 A,B 两个数组对应元素作乘幂
.\	数组左除, A.\2 为数组 A 的每个元素去除 2, A.\B 意义类似
./	数组右除, A./2 为数组 A 的每个元素除以 2, A./B 意义类似

注意数组运算与矩阵运算在符号上和结果上的区别。

另外，MATLAB 已经有定义的数学函数具有对数组运算的功能。例如，输入：

```
x=1:5, sin(x)
x =
    1     2     3     4     5
```

输出：


```
ans =
    0.8415    0.9093    0.1411   -0.7568   -0.9589
```

0.4 MATLAB 文件与编程


0.4.1 数据文件的存储和调用

在清除变量或退出 MATLAB 时，变量不复存在。为了保存变量的值，可以把它们存储在数据文件中。例如，输入：

```
Clear; A=2,B=1,C=A-B
```

执行以后在 File 菜单选 Save Workspace As 存入数据文件，取文件名（如 ABC.mat）。则在以后的操作中可以调用这个数据文件。只要在 File 菜单中单击 Open 等操作，则可以打开这个文件。在工具栏中单击相应的打开文件的图标（），也同样能找到要打开的文件。

0.4.2 M 文件

在进行复杂运算时，在指令窗口（Command Window）调试程序或修改指令是不方便的。因此，需要从指令窗口工具栏的  按钮或菜单 File: New: M-File 可进入 MATLAB 的程序编辑器窗口，以编写自己的 M 文件。

M 文件分为两类：M 脚本文件和 M 函数文件。

将多条 MATLAB 语句写在编辑器中，以扩展名为 m 的文件保存在某一目录中，就得到一个脚本文件。例如，在 M 文件编辑器中输入（见图 0.3）：

```
clear;
n=1:50;
m=sum(n)           %sum 是求和命令
```

单击工具栏中的保存按钮，保存以后选择 Debug: run 按钮（运行），则在指令窗口输出：

```
m =
    1275
```

注 文件名与变量名的命名规则相同，M 文件一般用小写字母。尽管 MATLAB 区分变量名的大小写，但不区分文件名的大小写。也要注意不与其他变量名及文件名冲突。

M 脚本文件没有参数传递功能，而 M 函数文件具有参数传递功能。因此 M 函数文件用得更为广泛。M 函数文件的格式有严格规定。M 函数文件必须以 `function` 开头，详细格式为

```
function 输出变量=函数名称(输入变量)
语句;
```

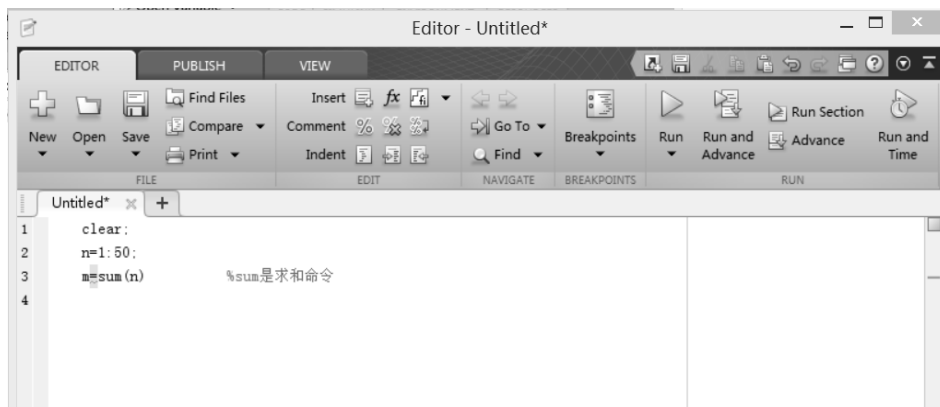


图 0.3 M 文件编辑器

例如，单击指令窗口工具栏的  按钮，进入 MATLAB 的程序编辑器窗口后，输入：

```
%M 函数 eg_1fun.m
function y=eg_1fun(c,t)
y=c(1)*exp(c(2)*t); %函数 eg_1fun 有两个参数 c 与 t，参数 c 是二维的，
                    参数 t 是一维的
```

在保存以后（保存为 `eg_1fun.m`），在指令窗口输入：

```
Clear;
eg_1fun([1,2],3) %M 函数可以传递参数
```

执行后得到：

```
ans =
    403.4288
```

又输入：

```
syms h u v %令 h,u,v 为符号变量
eg_1fun([u,v],h) %M 函数可以传递参数
```

执行后得到：

```
ans =  
u*exp(v*h)
```

因为 M 函数必须给输入参数赋值，所以编写 M 函数必须在编辑器窗口，而执行 M 函数要在指令窗口，并给输入参数赋值。M 函数不能像 M 脚本文件那样在编辑器窗口通过 Debug: run 菜单执行。M 函数可以被其他 M 函数文件或 M 脚本文件调用。为了以后调用时的方便，文件名最好与函数名同名。

0.4.3 inline 函数和匿名函数

比较简单的函数可以不用写成外部 M 函数文件，而是用更简捷的 inline 函数或匿名函数方式（MATLAB 7.0 有此功能）。inline 函数的使用格式为

```
fun=inline('expr',arg1,arg2,...) %fun 为函数名,expr 为表达式,arg1 为变量 1,...
```

匿名函数的使用格式为

```
fun=@(arg1,arg2,...)expr, %fun, expr, arg1 意义同上, MATLAB7.0 的新功能
```

先举 inline 函数的例子。从 MATLAB 指令窗口输入：

```
fname=inline('sum(1./(1:m).^2)','m')  
%定义 inline 函数 fname, 从表达式看, 是求  $1/1^2+1/2^2+1/3^2+\dots+1/m^2$  的函数, m 是变量。
```

执行后得：

```
fname =  
Inline function:  
fname(m) = sum(1./(1:m).^2)
```

输入：

```
fname(100)
```

执行后得：

```
ans =  
1.6350
```

再看匿名函数的例子。在指令窗口输入：

```
fname=@(m) sum(1./(1:m).^2)
```

执行后得到：

```
fname  
= @(m) sum(1./(1:m).^2)
```

输入：

```
feval(fname,100)
```

输出:

```
ans =  
    1.6350
```

注 (1) 注释。在编写程序时,为了增加可读性,常常使用注释语句。注释语句用 % 开头,使本行后面的字符不参与运算,只起说明作用。M 文件开头一般应有一段注释,说明文件的功能和使用方法。这部分注释在使用 help 时可以看到。

(2) 全程变量与局部变量。M 函数中的所有变量为局部变量,不进入工作空间(Workspace),M 脚本文件中所有变量在执行后进入工作空间,即为全程变量。

0.4.4 循环语句、分支语句与简单编程

(1) 循环语句 for

语 法
for 循环变量=数组 指令组; end

使用说明
对于循环变量依次取数组中的值,循环执行指令组直到循环变量遍历数组。数组常采用的形式是“初值:增量:终值”

(2) 循环语句 while

语 法
while 条件式 指令组; end

使用说明
当条件式满足,循环执行指令组直到条件式不满足。使用 while 语句要注意避免出现死循环

(3) 分支语句 if

语 法
if 条件式 1 指令组 1; elseif 条件式 2 指令组 2; ; else, 指令组 k; end

使用说明
如果条件式 1 满足,则执行指令组 1,且结束该语句;否则检查条件式 2,若满足执行指令组 2,且结束该语句;……;若所有条件式都不满足,则执行指令组 k,并结束该语句。

最常用的格式是

if 条件式, 指令组; end

注意语句中标点符号的使用。

【例 3】 计算 $s = \sum_{n=1}^{100} \frac{1}{n^2}$ 。

在指令窗口输入：

```
clear;s=0;
for n=1:100
s=s+1/n^2;
end
s
```

执行后得到：

```
s=
1.6350
```

也可以利用 **while** 循环语句。输入：

```
clear;s=0;n=1;
while n<=100
s=s+1/n^2;
n=n+1;
end
s
```

输出结果相同。

注 在使用循环语句时，如果不小心陷入了死循环，可以使用快捷键 Ctrl+C 强行中断。

0.4.5 其他

语 句	含 义
pause	表示中断语句，使程序暂停执行，直到击键盘
break	中断语句，用在循环句内表示跳出循环
input	用在交互式执行程序中提示键盘输入
disp	用于屏幕显示

0.5 符号运算初步

0.5.1 字符串的定义方法

MATLAB 用单引号对来定义字符串。例如，在指令窗口输入：

```
A='Olympics'
```

输出:

```
A=  
Olympics
```

0.5.2 命令 `syms` 定义符号变量与符号表达式

在 MATLAB 指令窗口, 输入的数值变量必须提前赋值, 否则提示出错。只有符号变量可以在没有提前赋值的情况下合法地出现在表达式中。但是符号变量必须预先定义。

命令

```
syms x y w p
```

表示将 x , y , w 和 p 定义为符号变量。继续输入:

```
z=sin(x)+cos(y)
```

执行以后 z 就表示符号表达式 $z=\sin(x)+\cos(y)$ 。

0.5.3 命令 `sym` 将数值表达式转换为符号表达式

命令 `sym` 的格式是

```
sym('数值表达式')
```

例如, 输入:

```
ss=sym('2008+sqrt(2)')
```

输出:

```
ss=  
2008+sqrt(2)
```

它是一个数值符号表达式, 而不是数值表达式。

0.5.4 命令 `eval` 可以计算符号表达式的值

继续前面的输入, `ss=sym('2008+sqrt(2)')` 是一个符号表达式, 如果要得到 `ss` 的近似值, 则需要计算它的值。输入:

```
eval(ss)
```

输出:

```
ans =  
2.0094e+003
```

由于 $ss = \text{sym}('2008 + \sqrt{2}')$ 实际上是一个符号常数，所以也可以用 `vpa` 命令计算。

【例 4】 计算定积分 $\int_1^4 \frac{3\sin(x^2)}{x} dx$

这个积分的原函数不是初等函数，使用没有解析解。输入：

```
syms x
g=int(3*sin(x^2)/x,1,4)      %命令 int 是符号积分命令
```

输出：

```
g =
3/2*sinint(16)-3/2*sinint(1)
```

输出中有特殊函数 `sinint(a)` 表示积分 $\text{int}(\sin(a*x)/x, 0, 1)$ ，为了得到近似解，输入：

```
eval(g)
```

执行后得：

```
ans =
1.0278
```

也可以输入：

```
vpa(g,6)      %数字 6 表示有效数字位数
```

得到：

```
ans=
1.02783
```

0.6 MATLAB 作图初步

0.6.1 二维曲线的绘制

这里只列出常用作图命令格式，更多内容与例子见实验一与实验六等。

命令格式	含义
<code>plot(x,y,'s')</code>	作出以数据 $(x(i), y(i))$ 为结点的折线图，其中 x, y 为同长度的向量。当向量 y 省略时， y 就是向量 x 的下标所组成的向量。其中 s 是参数，用来指定绘制曲线的线型，颜色，数据点的形状等
<code>plot(x1,y1,'s1',x2,y2,'s2',...)</code>	同时作出多条折线，分别由向量对 $(x1,y1), (x2,y2), \dots$ 构成
<code>fplot('fun',[a,b],'s')</code>	作出符号表达式表示的函数 <code>fun</code> 在 $[a,b]$ 上的图形
<code>ezplot('fun',[a,b],'s')</code>	是与 <code>plot</code> 命令对应的符号表达式作图命令

当区间 $[a,b]$ 默认时，默认的绘制区间是 $[-2\pi, 2\pi]$ 。`ezplot` 还可以用于隐函数作图，二维参数方程确定的曲线作图（见实验一）。

0.6.2 三维曲线绘制

命令格式	含义
<code>plot3(x,y,z,'s')</code>	绘制空间曲线, 其中 x, y, z 是同长度的向量。 s 是参数, 意义类似于 <code>plot</code> 中的参数
<code>plot3(x1,y1,z1's1',2,y2,z2 's2', ...)</code>	同时绘制多条绘制空间曲线
<code>ezplot3('x1','y1','z1',[a,b],'s1')</code>	是与 <code>plot3</code> 对应的符号表达式作图命令

0.6.3 三维网面图与曲面图

命令格式	含义
<code>meshgrid</code>	生成 X - Y 平面上的网格数据
<code>[x,y]=meshgrid(xa,ya)</code>	当 xa, ya 分别为 m 维和 n 维和向量, 得到 x 和 y 均为 n 行 m 列矩阵: x 的各列相同, y 的各行相同
<code>z=f(x,y)</code>	计算网格数据(格点矩阵)上的函数值。 z 是与 x, y 同阶的矩阵
<code>mesh(x,y,z)</code>	分别以 x, y, z 对应的元素为横、纵、竖坐标, 绘制网面图。是最基本的曲面图形指令。可以增加参数项, 以改变网线的颜色、线型等
<code>ezmesh('f(x,y)')</code>	对符号表达式 $f(x,y)$ 表示的曲面作网面图
<code>ezmesh('x(s,t)','y(s,t)','z(s,t)')</code>	对由符号参数方程表示的曲面作网面图
<code>surf(x,y,z)</code>	分别以 x, y, z 对应的元素为横、纵、竖坐标, 绘制曲面图
<code>ezsurf('f(x,y)')</code>	对符号表达式 $f(x,y)$ 表示的曲面作曲面图
<code>ezsurf('x(s,t)','y(s,t)','z(s,t)')</code>	对由符号参数方程表示的曲面作曲面图
<code>contour(x,y,z)</code>	绘制等值线图, 用法与 <code>mesh</code> 类似
<code>ezcontour</code>	绘制等值线图, 用法参照 <code>ezsurf</code> 等

0.6.4 图形的说明和定制

命令格式	含义
<code>title('字符串')</code>	标记图形标题的命令
<code>xlabel,ylabel,zlabel</code>	标记坐标轴的命令, 用法类似于 <code>title</code> 命令
<code>text(x,y,'字符串')</code>	在二维图形指定位置 (x,y) 处加文本字符串
<code>text(x,y,z,'字符串')</code>	在三维图形指定位置 (x,y,z) 处加文本字符串
<code>grid on/of</code>	显示/不显示格栅
<code>box on/of</code>	使用/不使用坐标框
<code>hold on/of</code>	保留/释放现有图形
<code>axis of/on</code>	不显示/显示坐标轴
<code>axis([a,b,c,d])</code>	定制 2 维坐标轴范围 $a < x < b, c < y < d$
<code>axis([a,b,c,d,e,f])</code>	定制 3 维坐标轴范围 $a < x < b, c < y < d, e < z < f$
<code>subplot(m,n,k)</code>	将图形窗口分为 $m \times n$ 个子图, 并指向第 k 幅图

详细格式可参照实验六, 或使用帮助菜单。

【例 5】 作由参数方程

$$\begin{cases} x = e^{-0.1t} \cos(t), \\ y = e^{-0.1t} \sin(t), & 0 < t < 20 \\ z = \sqrt{t}, \end{cases}$$

表示的空间曲线。

在编辑器窗口编写 M 脚本文件，保存并执行后得到图 0.4。

```
clear;close;
t=0:0.1:20;
r=exp(-0.1*t);
x=r.*cos(t);y=r.*sin(t);z=sqrt(t);
subplot(1,2,1)
plot3(x,y,z);
title('螺旋线');
text(x(end),y(end),z(end),'end')
xlabel('X轴');ylabel('Y轴');zlabel('Z轴');
subplot(1,2,2);
plot3(x,y,z);
grid on
```

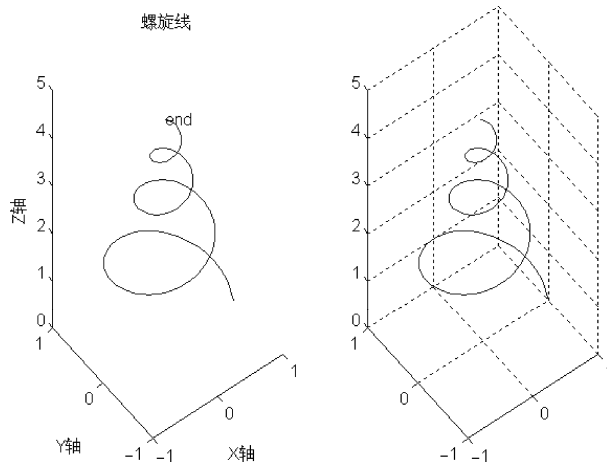


图 0.4 参数方程表示的空间曲线

0.7 MATLAB 帮助系统

充分利用在线帮助是边学边用 MATLAB 的最有效方法。常用命令有 help, lookfor 等。

命 令	功 能
help	显示 MATLAB 及其工具箱的主题目录（目录很多）
help 子目录名	显示子目录中所有 MATLAB 系统指令及函数
help 指令或函数	显示该指令或函数的说明部分
lookfor 关键字	显示与该关键字有关的指令和函数
type M 文件主名	显示该 M 文件程序代码
which M 文件主名	显示该 M 文件的 MATLAB 的路径
doc 指令或函数	打开该指令或函数的帮助窗口（HTML 文件形式）
demo	演示 MATLAB 功能

例子列述如下。

```

help mesh           %显示 M 函数 mesh 的用法说明（其 M 文件的注释部分）
which mesh          %显示 M 函数 mesh 所在的目录
type mesh           %显示函数 mesh 的 M 文件代码
doc mesh            %打开函数 mesh 的帮助窗口（HTML 文件形式）
lookfor surface     %显示 MATLAB 搜索路径中凡是第一行注释含有 surface 的 M 函数或指令

```

当不能确切地知道函数名的拼法时，还可以用在指令窗口输入模糊函数名加双击 Tab 键来寻找。

例如，要寻找与多项式有关的命令，只知道模糊名 pol，在指令窗口输入：

```
pol
```

再双击 Tab 键就得到与 pol 有关的命令：

```

PolynomEvalSFnV1 poly2ac      poly2th      polydec  polymerge  polyvalm
PolynomJacobSFnV1 poly2lsf    poly2trellis polyder  polyscale  polyxpoly
pol2cart          poly2mask    polyarea    polyeig   polysplit
polar             poly2rc      polybool    polyfit   polystab
polcmap           ly2str       polycon     polyform  polytool
polsim           poly2sym     polyconf    polyint   polyval
poly             poly2tfd     polycut     polyjoin  polyval_mex

```

0.8 实验作业

1. 执行下列指令，观察其运算结果，理解其意义，

- (1) $[1 \ 2; 3 \ 4]+10-2i$
- (2) $[1 \ 2; 3 \ 4].*[0.1 \ 0.2; 0.3 \ 0.4]$
- (3) $[1 \ 2; 3 \ 4].\backslash[20 \ 10; 9 \ 2]$
- (4) $[1 \ 2; 3 \ 4].^2$
- (5) $\exp([1 \ 2; 3 \ 4])$