

MATLAB 语言

——演算纸式的科学与工程计算语言

张培强 主编



中国科学技术大学出版社

TP312M1A

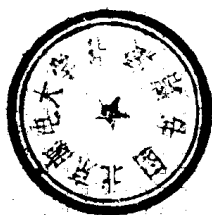
Z278

695947

MATLAB 语言

——演算纸式的科学与工程计算语言

张培强 主编



21113000951803

YD27/26

中国科学技术大学出版社

1995·合肥

内 容 简 介

Matlab 语言是一种“演算纸式”的用于科学工程计算的高级语言。它在矩阵代数、一般数值运算、数字信号处理、振动理论、系统识别、时序分析与建模、优化设计、神经网络控制、化学统计学、动态仿真系统、特殊函数和图形方面有着广泛的应用。亦可作为工程数学问题的研究与教学的辅助工具。与 BASIC、FORTRAN 和 C 等语言相比,编程效率高,语言简单,内涵丰富,易学易用,且扩充能力强。因此风行美国,流传世界。

本书参照国外关于在微机上应用 Matlab 语言最新版本的有关资料,结合我们实际应用中的体会,组织编写而成。书中深入浅出,循序渐进地介绍了 Matlab 语言的主要特点、安装方法和工作原理,着重表述了使用的基本语法规则等,并给出了若干典型算例。同时列出主要命令和函数的说明与举例。

本书可作为大中专院校师生、科研和工程技术人员学习使用 Matlab 语言的参考书。

图书在版编目(CIP)数据

MATLAB 语言——演算纸式的科学工程计算语言/张培强 主编.—合肥:中国科学技术大学出版社,1995 年 11 月
ISBN 7-312-00712-0/TP·113

I MATLAB 语言——演算纸式的科学工程计算语言
Ⅰ 张培强 主编
Ⅲ ①MATLAB ②计算机语言
Ⅳ TP

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号,230026)

合肥骆岗印刷厂印刷

全国新华书店经销

开本:787×1092/16 印张 20.25 字数:490 千字

1995 年 11 月第 1 版 1995 年 11 月第 1 次印刷

印数:1—5000 册 定价:22.00 元

前 言

Matlab 语言是一种高效率的用于科学与工程计算的高级语言。

与 Basic, Fortran, C 等语言比较, Matlab 的语法规则简单, 更加贴近人的思维方式。用 Matlab 写程序, 犹如在一张演算纸上排列公式和求解问题, 编程效率很高, 因此称为“演算纸式的”科学工程的算法语言。

Matlab 语言调试方便, 调试过程中可设置中断点, 存储多个中间结果, 把编辑、编译、连接和执行融为一体, 并能快速排除程序中的错误。可以说, Matlab 不仅是一种语言, 而且在广义上是一种语言开发系统。

Matlab 语言扩充能力强, 能够方便地扩展其功能和方便地调用 Fortran 语言和 C 语言的已有程序, 从而可充分利用已有的程序资源。

Matlab 语言在进行矩阵运算方面, 显得特别简捷、高效和方便。随着 Matlab 版本的不断更新, 其功能越来越强, 使之在诸如一般数值计算、数字信号处理、系统识别、自动控制、振动理论、时序分析与建模、优化设计、神经网络控制、化学统计学、动态仿真系统、特殊函数和图形等领域表现出一般高级语言难以比拟的优势, 并且可方便地用于几乎所有的科学和工程计算的各个方面。同时, 可以作为工程数学问题的研究与教学的辅助工具。近年来, 不同版本与用于不同机型和操作系统的 Matlab 语言已风行美国, 流行世界。

Matlab 语言易学易用, 不要求使用者有高深的数学和程序语言知识, 不需要使用者深刻了解算法和编程技巧。只要将数学方程算式按 Matlab 语言规则输入给计算机, Matlab 将如您所愿的那样给出该问题的相应解。

我们针对用于 PC 机的 Matlab 的 Window 环境下的最新版本 4.0, 结合使用中的心得与体会, 编写了此书。对 DOS 环境下或对其它版本 Matlab 语言使用者而言, 其主要使用方法是相通的。

本书共分五章。第一章绪论介绍了 Matlab 的主要特点; 第二章介绍了 Matlab 的安装和工作原理, 第三章讲述了基本使用方法, 介绍有关使用 Matlab 的基本知识和语法规则; 第四章给出了若干典型算例, 以便于初学者对照练习; 第五章为主要函数和命令的说明与举例。

本书编写的目的是希望把这一新颖高效的语言, 系统地介绍给读者, 尤其是大学生、研究生和科技工作者。若此书能引起读者对 Matlab 语言的兴趣并对使用这种语言的读者有所帮助的话, 作者将感到十分荣幸。由于作者水平有限, 经验不足, 错误和不当之处恳请得到专家、读者的指正。

参加本书编写工作的师生有: 副教授王琪民, 博士生肖奇志以及韩海晏、黄岗、邱丹、董志梅等同学。高级工程师张培仁在百忙中审阅了全部书稿, 并提出了许多宝贵的意见和建议。

在本书编写过程中, 得到了东南大学吴镇杨副教授和武汉交通科大冯文琴副教授的大力支持, 并得到中国科技大学出版社的帮助, 在此一并表示衷心感谢。

张培强

1995 年 1 月

目 次

第一章 绪论	(1)
1.1 概述	(1)
1.2 Matlab 的主要特点	(3)
1.3 Matlab 作者、版本和组成	(16)
1.4 Matlab 的应用	(17)
第二章 Matlab 的安装和工作原理	(21)
2.1 Matlab 的安装和启动	(21)
2.2 系统要求	(23)
2.3 Matlab 语言的工作原理	(24)
一、Matlab 语言的结构	(24)
二、Matlab 的磁盘文件	(24)
三、Matlab 的库函数	(27)
四、Matlab 的执行	(27)
2.4 环境参数	(30)
2.5 命令与文件的编辑与建立	(31)
一、命令行的编辑	(31)
二、M 文件的编辑与建立	(32)
2.6 图形拷贝及文件打印	(34)
2.7 使用外部程序	(34)
2.8 调用 C 和 Fortran 子程序	(36)
第三章 基本使用方法	(37)
3.1 基础知识	(37)
一、输入简单矩阵	(37)
二、矩阵元素	(38)
三、语句和变量	(39)
四、who 和固定变量	(39)
五、数和算术表达式	(40)
六、复数与复数矩阵	(41)
七、输出格式	(41)
八、Help 命令	(42)
九、退出工作空间和变量的保存	(42)
十、Matlab 函数的一些内容	(43)

■

十一、磁盘操作命令	(14)
3.2 矩阵运算	(45)
一、矩阵的转置	(45)
二、矩阵加和减	(45)
三、矩阵乘法	(46)
四、矩阵除法	(46)
五、矩阵乘方	(47)
六、矩阵超越函数	(47)
七、建立矩阵函数	(47)
八、建立大矩阵	(48)
3.3 向量和下标	(49)
一、产生向量	(49)
二、下标	(50)
三、具有 0—1 向量的下标	(51)
四、空矩阵	(52)
3.4 矩阵函数	(52)
一、三角分解	(52)
二、正交变换	(54)
三、奇异值分解	(55)
四、特征值	(56)
五、秩和条件	(56)
3.5 数组运算	(57)
一、数组的加和减	(57)
二、数组的乘和除	(57)
三、数组的乘方	(57)
四、关系运算	(58)
五、逻辑运算	(60)
六、数学函数	(60)
3.6 数据处理	(62)
一、列定位分析	(63)
二、缺损值	(65)
三、消去无关项	(66)
四、回归和曲线拟合	(66)
3.7 绘图	(68)
一、X-Y 绘图	(69)
二、基本形式	(69)
三、多重线	(71)
四、图线形式和颜色	(71)

五、虚数和复数	(72)
六、对数座标、极座标和条形图	(73)
七、三维作图	(73)
八、人工标注坐标轴尺寸	(74)
九、图形存储和打印	(75)
3.8 M 文件中的一些控制语句	(75)
一、控制流	(75)
二、echo, pause, input, keyboard, debug 语句	(78)
三、串和宏串	(80)
3.9 输入输出数据	(81)
一、输入数据	(81)
二、输出数据	(82)
第四章 算例汇编	(83)
4.1 复数运算	(83)
一、复数除法	(83)
二、复数的指数	(83)
三、复数的自然对数	(84)
四、复数的模	(84)
五、复数的平方根	(85)
六、复数的三角函数	(85)
七、复数的幂指数	(86)
4.2 插值和数值微商	(86)
一、一元全区间不等距插值	(86)
二、一元三点不等距插值	(88)
三、埃尔密特插值	(90)
四、第一种边界条件三次样条函数插值、微商与积分	(91)
五、第二种边界条件三次样条函数插值、微商与积分	(95)
六、第三种边界条件三次样条函数插值、微商与积分	(99)
七、二维光滑插值	(102)
4.3 数值积分	(108)
一、变步长辛普生方法求积	(108)
二、高斯方法求积(1)	(113)
三、高斯方法求积(2)	(114)
四、龙贝格方法求积(1)	(117)
五、龙贝格方法求积(2)	(119)
六、改进的龙贝格方法求积	(121)
七、牛顿—柯特斯方法求积	(123)
4.4 线性代数计算	(127)

一、求复系数线性方程组的解	(127)
二、LU 分解	(131)
三、对称正定矩阵的平方根法	(133)
四、LDL ^T 分解	(135)
五、求病态线性方程组的解	(137)
4.5 代数方程	(140)
一、二次方程的直接公式解法	(140)
二、三次方程的直接公式解法	(141)
三、四次方程的直接公式解法	(145)
四、牛顿法	(148)
五、特征值求根(1)	(150)
六、特征值求根(2)	(151)
4.6 常微分方程的数值积分及矩阵特征值	(153)
一、龙格—库塔方法(1)	(153)
二、龙格—库塔方法(2)	(159)
三、特征值和特征向量	(161)
四、QR 算法	(165)
第五章 主要命令和函数的注释与举例	(168)
5.1 一般命令与演示实例(GENERAL AND DEMO)	(168)
5.2 MATLAB 的文件 I/O 函数简集(IOFUN)	(170)
5.3 运算符与逻辑符(OPS)	(178)
5.4 字符串函数(STRFUN)	(187)
5.5 基本的数学函数(ELFUN)	(193)
5.6 初等矩阵和矩阵变换(ELMAT)	(202)
5.7 多项式和插值函数(POLYFUN)	(212)
5.8 数据分析和傅立叶变换(DATAFUN)	(220)
5.9 非线性数值方法(FUNFUN)	(228)
5.10 矩阵函数——数值线性代数(MATFUN)	(231)
5.11 二维图形(PLOTXY)	(242)
5.12 三维图形(PLOTXYZ)	(249)
5.13 常用图形函数(GRAPHICS)	(257)
5.14 彩色控制和亮度模式函数(COLOR)	(269)
5.15 声信号处理功能(SOUNDS)	(273)
5.16 特殊矩阵函数(SPECMAT)	(275)
5.17 稀疏矩阵函数(SPARFUN)	(279)
5.18 特殊数学函数(SPECFUN)	(289)
5.19 语言结构与程序调试(LANG)	(297)
5.20 命令索引	(307)

第一章 绪 论

1.1 概 述

在科学研究和工程应用中,往往要进行大量的数学计算,其中包括矩阵运算。这些运算一般来说难以用手工精确和快捷地进行,而要借助计算机编制相应的程序做近似计算。目前流行用 Basic、Fortran 和 C 语言编制计算程序,既需要对有关算法有深刻的了解,还需要熟练地掌握所用语言的语法及编程技巧。对多数科学工作者而言,同时具备这两方面才能有一定困难。通常,编制程序也是繁杂的,不仅消耗人力与物力,而且影响工作进程和效率。为克服上述困难,美国 Mathwork 公司于 1967 年推出了“Matrix Laboratory”(缩写为 Matlab)软件包,并不断更新和扩充。目前最新的 4.0 版本(Window 环境)是一种功能强、效率高便于进行科学和工程计算的交互式软件包。

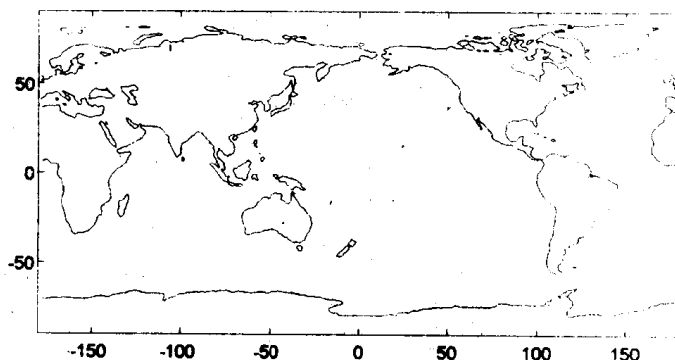


图 1.1

其中包括:一般数值分析、矩阵运算、数字信号处理、建模和系统控制和优化等应用程序,并集应用程序和图形于一便于使用的集成环境中。在此环境下所解问题的 Matlab 语言表述形式和其数学表达形式相同,不需要按传统的方法编程。不过,Matlab 作为一种新的计算机语言,要想运用自如,充分发挥它的威力,也需先系统地学习它。但由于使用 Matlab 编程运算与人进行科学计算的思路和表达方式完全一致,所以不象学习其它高级语言——如 Basic、Fortran 和 C 等那样难于掌握。实践证明,你可在几十分钟的时间内学会 Matlab 的基

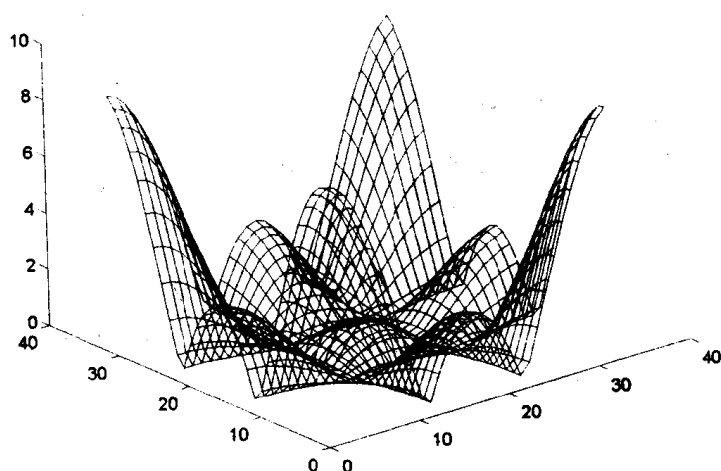


图 1.2

基础知识,在短短几个小时的使用中就能初步掌握它。从而使你能够进行高效率 and 富有创造性的计算。Matlab 大大降低了对使用者的数学基础和计算机语言知识的要求,而且编程效率和计算效率极高,还可在计算机上直接输出结果和精美的图形拷贝(如下面的图 1.1—图 1.4),所以它的确为一高效的科研助手。自推出后即风行美国,流传世界。

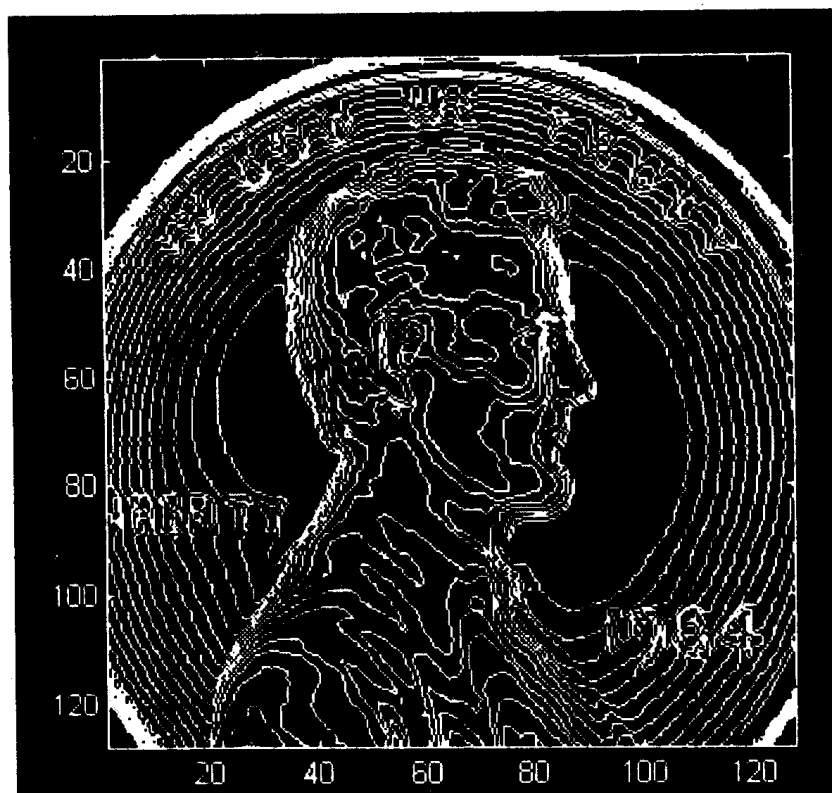


图 1.3

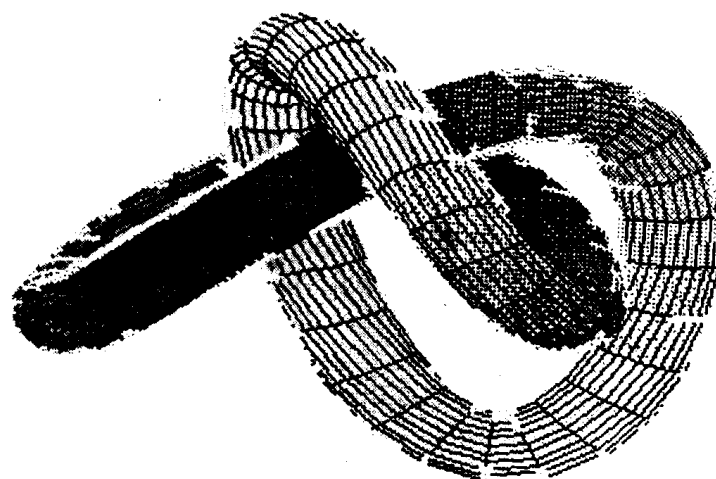


图 1.4

1.2 Matlab 的主要特点

Matlab 最明显的特点是功能强,可以进行各种科学和工程运算,特别容易学习和使用,而且可根据自己的实际需要扩充其功能。

Matlab 的基本数据单元是不需要指定维数的矩阵,所以在 Matlab 环境下数组(向量或矩阵)的操作同数的操作一样简单;而 Basic、Fortran 和 C 等语言的基本数据单元是数,因而要先定义数组,然后才能进行有关操作。一般说来,数组操作要比数的操作复杂得多。请看下面的例子:

例 1 已知矩阵

$$A = \begin{bmatrix} 5 & 3 & -21 & 42 \\ 44 & 15 & 0 & 6 \\ 97 & 6 & 81 & 2 \end{bmatrix}, B = \begin{bmatrix} 22 & 7 \\ 97 & -53 \\ 45 & 0 \\ 72 & 1 \end{bmatrix}$$

求 A、B 矩阵的积 C。

Basic 程序如下:

```
DIM a(3,4),b(4,2),c(3,2)
FOR i=1 TO 3
  FOR j=1 TO 4
    READ a(i,j)
  NEXT j
NEXT i
FOR i=1 TO 3
  FOR j=1 TO 2
    READ b(i,j)
  NEXT j
NEXT i
DATA 5.,3.,-21.,42.,44.,15.,0.,6.,97.,6.,81.,2.,22.,7.,97.,-53.,45.,
0.,72.,1.
FOR i=1 TO 3
  FOR j=1 TO 2
    c(i,j)=0
    FOR k=1 TO 4
      c(i,j)=c(i,j)+a(i,k)*b(k,j)
    NEXT k
  NEXT j
NEXT i
PRINT"a"
```

```

FOR i=1 TO 3
PRINT a(i,1),a(i,2),a(i,3),a(i,4)
NEXT i
PRINT"b"
FOR i=1 TO 4
PRINT b(i,1),b(i,2)
NEXT i
PRINT"c"
FOR i=1 TO 3
PRINT c(i,1),c(i,2)
NEXT i
RETURN

```

在 DOS5.0 系统提供的 QBASIC 下运行上述程序,屏幕上显示如下结果:

a			
5	3	-21	42
44	15	0	6
97	6	81	2
b			
22	7		
97	-53		
45	0		
72	1		
c			
2480	-82		
2855	-481		
6505	363		

Fortran 程序如下:

```

dimension a(3,4),b(4,2),c(3,2)
data a/5.,44.,97.,3.,15.,6.,-21.,0.,81.,42.,6.,2./
data b/22.,97.,45.,72.,7.,-53.,0.,1./
do 10 i=1,3
do 10 j=1,2
c(i,j)=0.
do 10 k=1,4
10 c(i,j)=c(i,j)+a(i,k)*b(k,j)
write(*,'(a)')a'
do 20 i=1,3

```

```

20  write( * , * )(a(i,j),j=1,4)
    write( * , ' (a)' )' b'
    do 30 i=1,4
30  write( * , * )(b(i,j),j=1,2)
    write( * , ' (a)' )' c'
    do 40 i=1,3
40  write( * , * )(c(i,j),j=1,2)
    stop
    end

```

C 语言程序如下:

```

main( )
{
int  a[3][4]={{5,3,-21,42},{44,15,0,6},{97,6,81,2}};
int  b[4][2]={{22,7},{97,-53},{45,0},{72,1}};
int  c[3][2],i,j,k;
    for(i=0;i<3;i++)
    {
        for(j=0;j<2;j++)
        {
            c[i][j]=0;
            for(k=0;k<4;k++)
                c[i][j]+=a[i][k]*b[k][j];
        }
    }
    for(i=0;i<3;i++)
    {
        for (j=0,j<2;j++)
            printf("c[%d][%d]=%d," ,i,j,c[i][j]);
        printf("\n");
    }
}

```

而如下 Matlab 程序要简短得多:

```

>>a=[5 3 -21 42;44 15 0 6;97 6 81 2];
>>b=[22 7;97 -53;45 0;72 1]
>>c=a * b;
>>a

```

```

5    3   -21   42
44   15    0    6
97    6    81    2

```

```
>>b
```

```

22    7
97   -53
45    0
72    1

```

```
>>c
```

```

2480   -82
2855  -481
6505   363

```

程序中第一行是输入矩阵 a , 第二行输入矩阵 b , 前两行为输入部分; 第三行求 a, b 矩阵之积, “ $*$ ”表示矩阵相乘的运算符, “ $>>$ ”表示输入命令提示符。矩阵的乘积, 赋值给 c 矩阵, 若没有指定变量时, 赋值给 ans 。这就是最简单的 Matlab 程序, 又称 M 文件。

在 Matlab 环境下, 许多复杂的数学运算, 如求矩阵的行列式值、求矩阵的逆及求特征值, 求函数的微分和积分, 以及进行多项式插值等, 都可以直接调用库函数。

例 2 输入矩阵 a :

```
>>a=[1.5 1.1 1.3;2.1 2.3 2.5;3.3 3.5 3.1]
```

```
a=
```

```

1.5000  1.1000  1.3000
2.1000  2.3000  2.5000
3.3000  3.5000  3.1000

```

求 a 的转置:

```
>>a'
```

```
ans=
```

```

1.5000  2.1000  3.3000
1.1000  2.3000  3.5000
1.3000  2.5000  3.1000

```

程序中“ $'$ ”表示矩阵的转置运算, 由于未指定赋值变量, 因此屏幕上显示出运算结果, 并赋值给 ans 。

求 a 的逆:

```
>>inv(a)
```

```
ans=
```

```

1.9565  -1.3766  0.2899
-2.1014  -0.4348  1.2319
0.2899    1.9565  1.3768

```

程序中“inv()”表示矩阵的逆运算。

求 a 的行列式值：

```
>>det(a)
```

```
ans =
```

```
-0.8280
```

程序中“det()”表示求矩阵的行列式。

求 a 的特征值：

```
>>eig(a)
```

```
ans =
```

```
6.9000
```

```
0.3464
```

```
-0.3464
```

程序中“eig()”表示求矩阵的特征值。

求 a 的秩

```
>>rank(a)
```

```
ans =
```

```
3
```

程序中“rank()”表示矩阵秩的计算。

对 a 作 LU 分解。

```
>>[L,U]=lu(a)
```

```
L =
```

```
0.4545    1.0000    0
```

```
0.6364   -0.1481    1.0000
```

```
1.0000    0         0
```

```
U =
```

```

3.3000    3.5000    3.1000
0         -0.4909   -0.1091
0          0         0.5111

```

程序中“[L,U]=lu()”表示将矩阵分解为 L 矩阵和 U 矩阵的乘积。

例 3 已知

$$\text{humps}(x) = \frac{1}{(x - 0.3)^2 + 0.01} + \frac{1}{(x - 0.9)^2 + 0.04} - 6,$$

求积分 $\int_0^1 \text{humps}(x) dx$

```
>>quad('humps',0,1)
```

```
ans =
```

```
29.8583
```

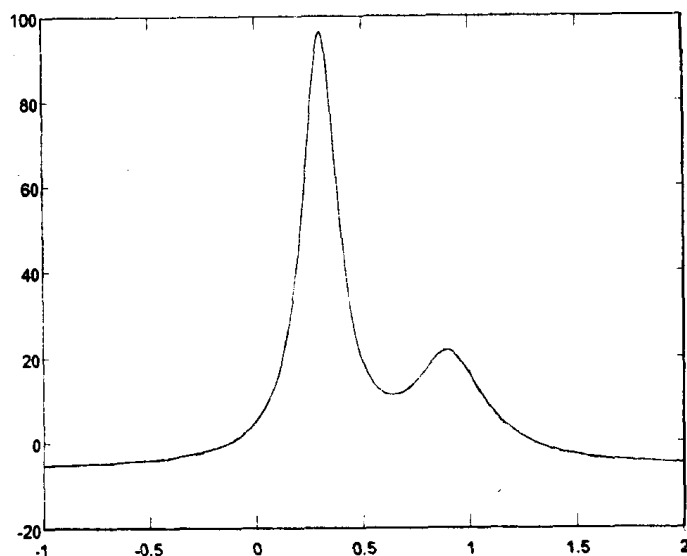


图 1.5

如图 1-5 所示。

例 5 $a=[1\ 2\ 3\ 4\ 6\ 4\ 3\ 4\ 5]$, 给出 a 的每个元素加上 2 后所得向量的图形。

可编一名为 vp1.m 的文件如下:

```

function vp1
a=[1 2 3 4 6 4 3 4 5];
b=a+2
plot(b)
grid

```

程序中“function vp1”表示将 M 程序定义成名为 vp1 的函数,“grid”表示在图形坐标上加

程序中“quad()”表示积分运算,() 中的“humps”为该被积函数名,0 和 1 为积分上下限。

在 Matlab 环境下,给出计算结果的图形与进行通常的数学运算一样方便,只需调用库函数中的绘图命令即可。

例 4 给出例 3 中函数 $\text{humps}(x)$, $x \in [-1, 2]$ 的图形。

```
>>x=-1:0.01:2;
```

```
>>plot(x,humps(x))
```

程序中“plot”表示线性坐标上的绘图命令,其中第一个变量 x 为输入变量,其值从 -1 至 2,步长为 0.01,第二个 $\text{humps}(x)$ 为输出变量。绘出的图形

上网络。

运行 vp1 即可在屏幕上显示向量 b, 如图 1-6 所示。当然, 也可以不编该文件, 直接键入命令, 即同前面的例子一样。

此外, Matlab 的很多库函数给用户多种功能。而且 Matlab 的函数命令简单, 对于不同输入变量和输出变量有着不同的含义。

例 6 最大分量函数 max

max(X) 有两种含义, 当 X 代表向量时, max(X) 代表向量中的最大元素。如:

```
>>X=[25 37 14]
```

```
>>max(X)
```

```
37
```

当 X 代表矩阵时, max(X) 代表矩阵中各列的最大元素组成的行向量。如:

```
>>X=[25 18 27; 37 29 16; 14 36 11]
```

```
X=
```

```
25 18 27
```

```
37 29 16
```

```
14 36 11
```

```
>>max(X)
```

```
ans=
```

```
37 36 27
```

当其输出变量为形式 [y, i]=max(X) 时, 其 y 代表矩阵各列的最大元素, 而 i 代表每个最大元素所对应矩阵中的行数。如:

```
>>[y,i]=max(X)
```

```
y=
```

```
37 36 27
```

```
i=
```

```
2 3 1
```

当输入变量增加为两个维数相同的矩阵 X, Y 时, max(X, Y) 返回一个与其维数相同的矩阵, 它代表比较两个矩阵相同位置的元素的大小, 取其大的一个做为返回矩阵中的相应元素。如:

```
>>X=[1 2; 13 8; 7 4; 9 6];
```

```
>>Y=[15 4; 3 9; 2 7; 1 6];
```

```
>>max(X,Y)
```

```
ans=
```

```
15 4
```

```
13 9
```

```
7 7
```

```
9 6
```