

第1章 MATLAB是什么

没有MATLAB就没有乐趣。

Nachtigal, M. N., Reddy, S. C., Trefethen, L.N.(1990)。

不对称矩阵迭代有多快？

关于迭代方法的Copper Mountain会议论文集，

Copper Mountain CO, 1-5, 1990年4月。

1.1 MATLAB能做什么

MATLAB是一个可视化的计算程序，被广泛地使用于从个人计算机到超级计算机范围内的各种计算机上。

MATLAB包括命令控制、可编程，有上百个预先定义好的命令和函数。这些函数能通过用户自定义函数进一步扩展。

MATLAB有许多强有力的命令。例如，MATLAB能够用一个单一的命令求解线性系统，能完成大量的高级矩阵处理。

MATLAB有强有力的二维、三维图形工具。

MATLAB能与其他程序一起使用。例如，MATLAB的图形功能，可以在一个FORTRAN程序中完成可视化计算。

25个不同的MATLAB工具箱可应用于特殊的应用领域。

MATLAB在以下的领域里解决各种问题是一个十分有效的工具：

- 工业研究与开发。
- 数学教学，特别是线性代数。所有基本概念都能涉及。
- 在数值分析和科学计算方面的教学与研究。能够详细地研究和比较各种算法。
- 在诸如电子学、控制理论和物理学等工程和科学学科方面的教学与研究。
- 在诸如经济学、化学和生物学等有计算问题的所有其他领域中的教学与研究。
- 在MATLAB中创建的组是矩阵，MATLAB的名字取自矩阵实验室(MATrix LABoratory)。

1.2 MATLAB实例

本节中的实例恰当而简洁地展示了MATLAB能做什么。在一些实例中给出了完整的MATLAB命令；而在另一些实例中，为简化仅给出部分命令。

在本书中出现的MATLAB代码用的是一种特殊的字体以区别于书中别的文字。MATLAB的输出是斜体字，即：我们输给MATLAB的命令是正体；MATLAB给出的输出答案是斜体。

百分符号%在MATLAB中用做注释符号，在本书中全部都是这样使用。采用的其他表示方法是：数量和预定义函数用斜体字，矩阵、向量和用户自定义函数用黑体字。矩阵用大写

字母开头命名,而向量以小写字母开头。细胞矩阵是如同矩阵或向量的概念,也采用黑体字,其结构和对象也是如此。在命令表中,用斜体字表示那些可选的函数参数。例如, `command(par1, par2)`, 参数 `par1` 总是需要的, 而 `par2` 是可选的。

例1.1 二维和三维函数

MATLAB能用于计算,并以二维和三维图形显示各种函数。在 MATLAB函数中包括了所有主要的数学函数和大量的高级函数。

(a) 用简短的MATLAB命令计算并绘制在 $0 \leq x \leq 6$ 范围内的 $\sin(2x)$ 、 $\sin^2 x$ 和 $\sin^2 x$ 。

```
x=linspace(0,6); % 创建一个向量x。
y1=sin(2*x);      % 向量y1等于x坐标上某一x的sin(2x)值。
y2=sin(x.^2);     % 向量y2等于sin(x.^2),同上。
y3=(sin(x)).^2;   % 向量y3等于(sin(x)).^2,同上。
```

命令 `plot(x,y1)` 绘制向量 `y1`, `y1` 作为向量 `x` 的一个函数, `plot` 命令的定义可参见第13章。由此能够很容易地在一个图上绘制 $\sin(2x)$ 、 $\sin(x^2)$ 和 $\sin^2 x$ 的曲线并正确地标记它们(图1-1)。

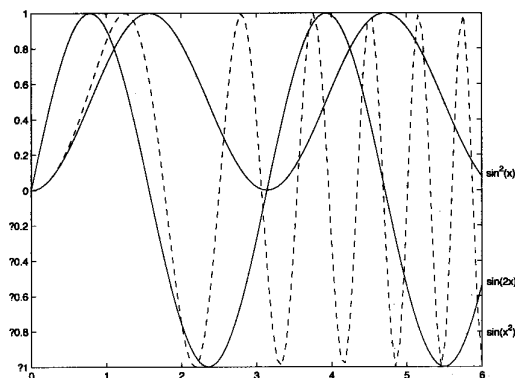


图1-1 同一图上的三条曲线

(b) 两个变量的函数需要用三维来恰当地图示, MATLAB能够给出很好的三维图。在图-2中,用四个不同的方法展示了函数 $z(x,y) = \cos(x) \cdot \sin(y)$ 的图形: 左上图用 `surf` 命令和 `shading interp`; 右上图用 `mesh`; 左下图用 `waterfall` 以及右下图用 `contour`。关于图像命令的详细信息可参见第3章。

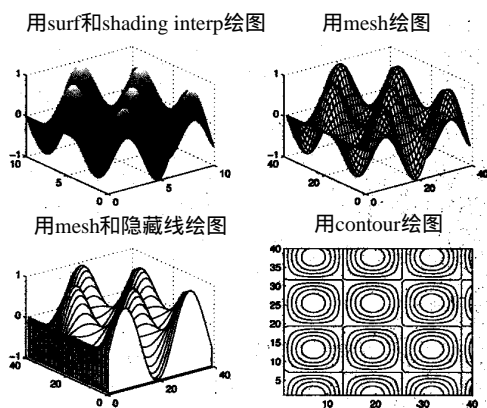


图1-2 四种方法绘制双变量的一个函数的图形

(c) MATLAB也能绘制一条参数曲线，例如：

$$\begin{cases} x = \cos t - \sin 3t \\ y = \sin t \cos t - \cos 3t \end{cases}$$

x - y 平面图如图1-3所示。

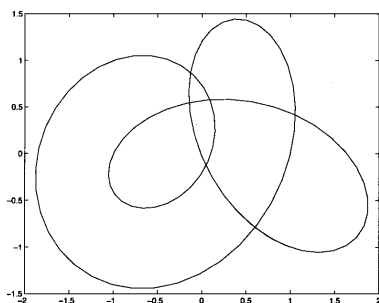


图1-3 一个参数曲线图

例1.2 函数分析

MATLAB命令fzero和fmin可以用于寻找一个函数的零点和最小值。

函数 $xe^{x^2} - e^{x^2} - \sin x^3$ 可以用名叫 **func** 的用户自定义函数 (见2.9节) 表示，并存入一个名叫 **func.m** 的M文件中。这个文件由下列行组成：

```
function y = func(x)
y = x.*exp(x.^2) - exp(x.^2) - sin(x.^3);
```

如果这个M文件被存放在当前的工作目录中，或在一个称为 **matlab** 的子目录中，函数 **func** 就可以像预定义的MATLAB函数一样调用。例如，调用 **xiszero=func(0)**，给出的答案是：

```
xiszero =
    -1
```

用这样定义的函数，MATLAB提供了一个命令来寻找方程 $xe^{x^2} - e^{x^2} - \sin x^3 = 0$ 的零点。命令 **xsolv=fzero('func', 3)** 给出：

```
xsolv =
    1.2194
```

在本例中，命令中的第2个自变量用的是3，是开始计算的一个初始近似值。

如果在 $-1 \leq x \leq 1.5$ 区间内绘制这个函数，则正确答案如图1-4所示。

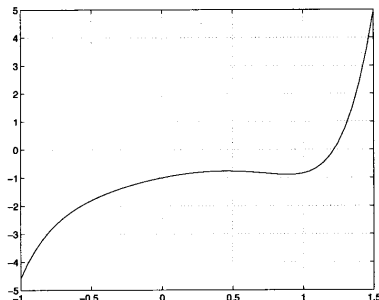


图1-4 在 $-1 \leq x \leq 1.5$ 区间内绘制函数 $xe^{x^2} - e^{x^2} - \sin x^3$ 的图形

当 x 在0.5和1之间时, 这个函数看起来有一个极小值, 为正确找出这个极小值, 用命令 `mpoint=fmin('func', 0.5, 1)`, 其结果为:

```
mpoint =
    0.8954
```

用于检查MATLAB中用户自定义函数的命令可参见第10章和第11章。

例1.3 线性系统与特征值

(a) MATLAB可以用一个简单的命令行求解线性系统, 系数矩阵 A 和右侧 b 定义如下:

$$A = \begin{pmatrix} 3 & 1 & -1 \\ 1 & 2 & 4 \\ -1 & 4 & 5 \end{pmatrix} \quad b = \begin{pmatrix} 3.6 \\ 2.1 \\ -1.4 \end{pmatrix}$$

这对应于线性系统 $Ax=b$, 如下所示:

$$\begin{cases} 3x_1 + x_2 - x_3 = 3.6 \\ x_1 + 2x_2 + 4x_3 = 2.1 \\ -x_1 + 4x_2 + 5x_3 = -1.4 \end{cases}$$

这可由如下命令求解:

```
x=A\b
```

其结果为:

```
x =
    1.4818
   -0.4606
    0.3848
```

(b) 还有许多矩阵控制命令。例如, 例(a)中矩阵 A 的特征值很容易地可以由下列命令得到:

```
[EigenVectors,EigenValues] = eig(A)
```

其给出:

```
EigenVectors =
   -0.9482   -0.3129   -0.0553
   -0.2887    0.7756    0.5613
    0.1328   -0.5482    0.8258
```

```
EigenValues =
    3.4445         0         0
         0   -1.2305         0
         0         0    7.7860
```

矩阵 **EigenVectors** 的列是 A 的特征向量, **EigenValues** 中对角线元素是特征值。由于矩阵 A 是对称的, 因此, 所有的特征值都是实数, 三个特征向量是相互正交的。

MATLAB中的基本概念是矩阵。基本的矩阵命令在第3章描述, 更多的命令将在第4、7、8、9章中描述。

例1.4 曲线拟合与插值

(a) 如果有两个向量 x 和 y 表示的 x - y 平面上的一组点, 那么, 可以对它们进行插值点或者拟合一条曲线。令

```
x = (1 1.5 3 4 5 6 6.5 7 8)
```

```
y=(1.2 1 1.7 2.5 2 2.3 2.5 3 3.1)
```

对应 x - y 平面上的9个点。首先，展示以最小二乘法拟合数据的线性函数，这个可以通过MATLAB中的三个简单的命令来实现：

```
p1=polyfit(x, y, 1)      % p1=向量等于一次多项式的系数。
linc=polyval(p1, x)      % linc=向量等于x点上多项式p1的值。
plot(x, linc, x, 'x')    % 绘制多项式和由 'x' 标记的数据。
```

结果见图1-5(左图)。

能以最小二乘法对一组点拟合高次多项式。对上面的命令行进行一点小改动就可以得到 7 次多项式：

```
p7=polyfit(x, y, 7);      % p7=向量等于7次多项式的系数。
xx=1:0.25:8;             % xx=所有想要进行多项式计算的点。
polc=polyval(p7, xx);     % polc=向量等于点xx上多项式p7的值。
plot(xx, polc, x, 'x')    % 绘制多项式和由 'x' 标记的数据。
```

其结果如图1-5(右图)所示。

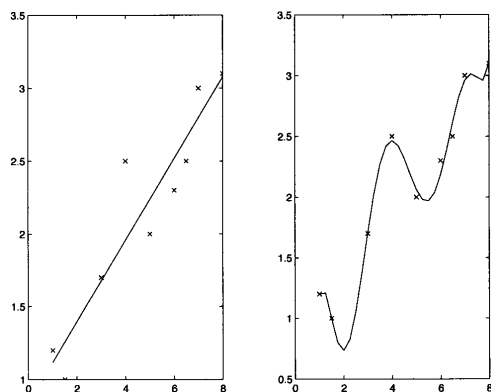


图1-5 x - y 平面上对一组具有9个点的数据拟合的1次和7次多项式

(b) MATLAB提供了二维和三维的内插函数。给定一组点 (x_i, y_i) 和一些内插点 x_i , MATLAB 能返回通过对这些数据内插的插入点的值，这可以有不同的方法实现。作为一个例子，将使用(a)中的一组点来给出在下列点中插入的值：

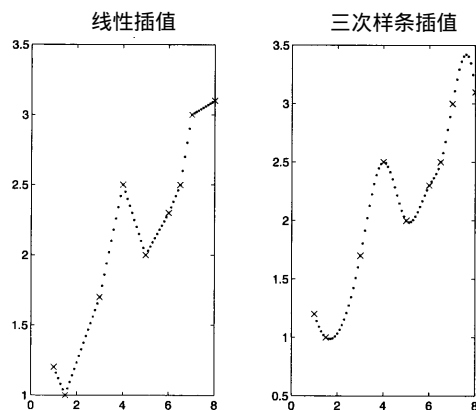


图1-6 piecewise 线性函数插值(左)和三次样条插值(右)

$\tilde{x}=(1 \ 1.1 \ 1.2 \ 1.3 \ \dots \ 7.9 \ 8)$

在图1-6中, 分别展示了分段线性插值和三次样条插值。‘x’ 标记表示原始数据, 点线是中间点的被插函数。

有关插值和曲线拟合的详细信息可参见第10章。

例1.5 统计

MATLAB包含了统计命令。例如, 很容易地求得实验数据的平均值和中位值以及绘制统计频数直方图或直方图。

图1-7显示了某小镇上每个人的年龄。上图是统计频数直方图, 显示了每个年龄的人数。例如, 看到两个年龄最大的人是92岁。统计频数矩形图也表示了小镇上没有人是11岁或12岁, 7岁的儿童7人。

也能看出32岁以下的人与32岁以上的人一样多, 因为这是一个中间年龄。此外, 平均年龄是35岁。这些也都在图中标出。

另外, 下图展出了小镇上居民的年龄。例如, 如果知道要列出的是第11人, 那么也就知道这个人是一个小孩。因为第11个直方块接触x轴。这种绘图称为阶梯图, 即没有内部线的直方图。统计命令在第6章中描述。

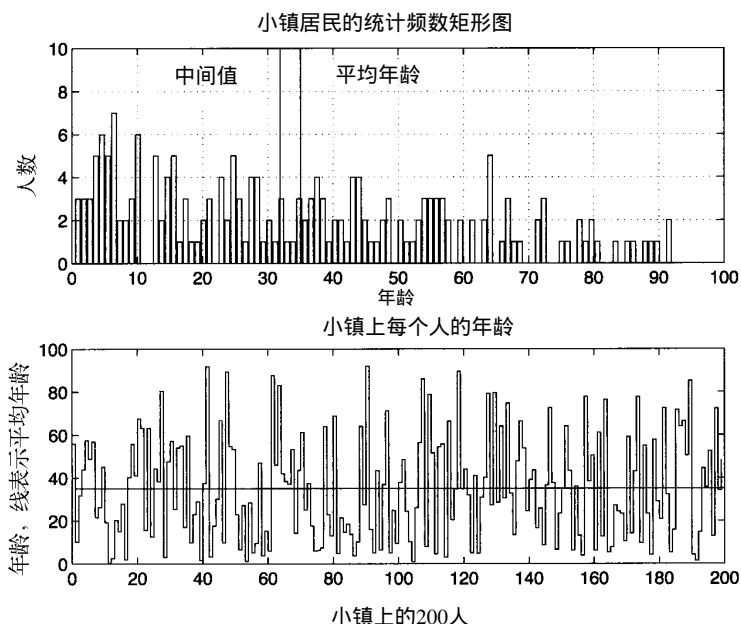


图1-7 来自小镇数据的一个统计频数矩形图

例1.6 傅立叶变换与信号分析

MATLAB能利用快速傅立叶变换FFT计算离散傅立叶变换, 这用于信号分析和解微分方程。

为了证明MATLAB的傅立叶变换, 用0和1之间的随机数干扰 $5\sin(x)+2\sin(5x)$ 函数:

```
x=linspace(0, *pi, 64);
signal=5*sin(x)+2*sin(5*x)+randn(x);
```

干扰信号和原始信号如图 1-8(上)所示。

然后，变换这个信号，并且删除变换后信号中的所有高频，即系数向量中心部分设为零。

```
transf=fft(signal);
filttransf(1:9)=transf(1:9);
filttransf(56:64)=transf(56:64);
```

傅立叶变换的实际部分如图 1-8(下)所示。图 1-9(上)中删除了高频傅立叶变换，仅低频向量通过逆傅立叶变换而变换：

```
filtsig=ifft(filttransf);
```

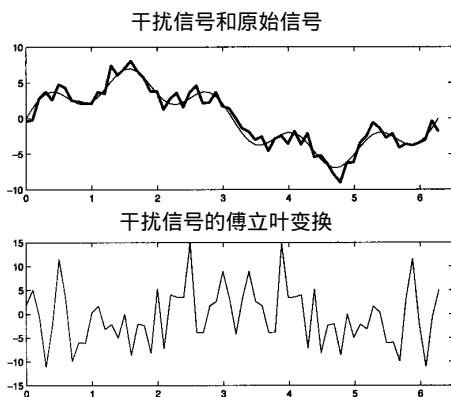


图1-8 干扰信号和它的傅立叶变换

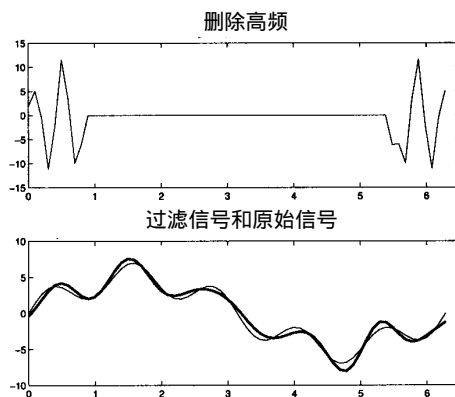


图1-9 过滤傅立叶变换

这个过滤的信号与原始信号一起在图 1-9(下)中展示。这个过滤的信号如期望一样是光滑的。因为这种干扰也影响部分低频信号，所以它与原始信号不等。

在10.5节中，作为MATLAB的一部分描述了傅立叶变换，更详细的有关信号处理工具箱的信息参见附录C。

例1.7 常微分方程

MATLAB可以用数字求解常微分方程。作为一个实例，展示在一个轨道上一只猫追赶一只机器玩具老鼠。猫和老鼠的位置分别用 $(x(t), y(t))$ 和 $(X(t), Y(t))$ 表示。猫的速度向量是和猫与老鼠之间的差向量对应的。猫从 $(50, 40)$ 开始，这就给出了如下的微分方程系统：

$$\begin{cases} \frac{dx}{dt} = -(X(t) - x(t)) \\ \frac{dy}{dt} = -(Y(t) - y(t)) \\ x(0) = 40 \\ y(0) = 50 \end{cases}$$

式中 w 是老鼠的速度， a 是猫和老鼠在 t 时刻相距的距离，由下式给出：

$$a(t) = \sqrt{(X(t) - x(t))^2 + (Y(t) - y(t))^2}$$

这个系统的解如图1-10所示，从图中可以看到由于老鼠跑得太快，猫并没有成功地抓到老鼠。

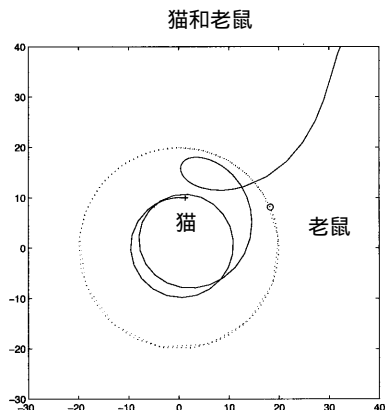


图1-10 猫捉机器玩具老鼠。小圆是老鼠的终点，十字是猫的终点

在11.2节中描述了如何使用MATLAB求解一般的常微分方程问题。

例1.8 偏微分方程

在MATLAB中有许多用于不同应用领域的工具箱。这里，作为一个例子给出 PDE工具箱，这个工具箱能通过使用有限元方法 (the Finite Element Method)解椭圆方程、抛物线方程和双曲线方程。区域要分成大量的三角形子区域，对每个三角形，其解由一个简单函数估计。所用的三角形越多，其偏差就越小。

求解这个区域里的椭圆问题 - $\Delta u = \sin(2\pi y + \pi/2)\cos(2\pi x + \pi/2)$ ，如图1-11所示。

拉普拉斯算子 Δ 是 $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ ，边界条件 $u=0$ ，也可由PDE工具箱所做的三角形子区域给出。这个问题的解可以用可视化等高线以二维绘制，如图1-12所示。

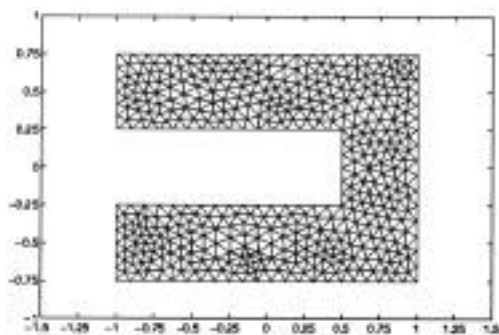


图1-11 PDE工具箱采用FEM的三角形子区域

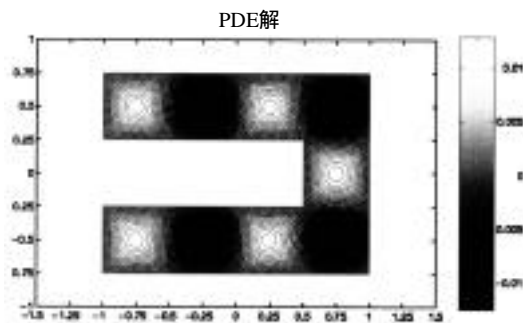


图1-12 PDE工具箱计算的等高线解的图示

例1.9 MATLAB中的编程

MATLAB是可编程的。命令序列可以在文本编辑器上写入，然后可以调用 MATLAB命令窗口上的用户定义函数或程序。文件的名字必须要有扩展名 .m，这个文件就称为 M文件。这些M文件可以用相同的方法象标准 MATLAB函数一样使用。

(a) 阶乘 $n! = 1 \times 2 \times 3 \times \dots \times n$ 可以用不同的方法计算。这里，展示一个递归的用户定义函数：


```
function p =factorial(nn)
```

```
%计算nn的阶乘。
```

```
if nn == 0
```

```
    p = 1;
```

```
else
```

```
    p = nn*factorial(nn-1);
```

```
end
```

这个M文件名叫**factorial.m**，对这个函数的调用如下所示：

```
fourfactorial=factorial(4)
```

其结果为：

```
fourfactorial=
```

```
    24
```

(b) 在MATLAB中可以使用面向对象的程序设计。有一个类 **world**，用来模拟一个小世界内的生与死。对于一个要在这个世界出生的个人来说，他需要一些邻居，但太多的邻居会构成生存危机。类 **world** 的命令是在目录 @world 中的文件 **world.m** 中。

```
Function w=world(Size, Density, nrCreate, nrSurvive)
```

```
% WORLD 生成一个生命世界。
```

```
% 这是命令方式。
```

```
% Size 给出这个世界的边，它是方的。
```

```
% Density 应该是在0与1之间，定义了实与空的近似比率。
```

```
% nrCreate 个需要生长的空单元的最近邻居的号码。
```

```
% nrSurvive 个需要生存和长大的满单元的最近邻居的号码。
```

```
colormap('gray')
```

```
w.Size = Size;
```

```
w.nrCreate = nrCreate;
```

```
w.nrSurvive = nrSurvive;
```

```
w.Map = round( (-0.5+Density)+rand(Size,Size) );
```

```
w.NextY = zeros(Size,Size);
```

```
w = class(w,'world');
```

属于类的这种方法在目录 @world 中定义，但并不在这里说明。现在，人们可以通过下列命令产生一个对象 **w**：

```
w=world(20, 0.6, [2 4 5], [3 6 7]);
```

现在重复

```
w=year(w)
```

与人类所经历的10年一样，反复10次，得到

图1-13。

关于MATLAB中的有关编程信息可参见第2章。

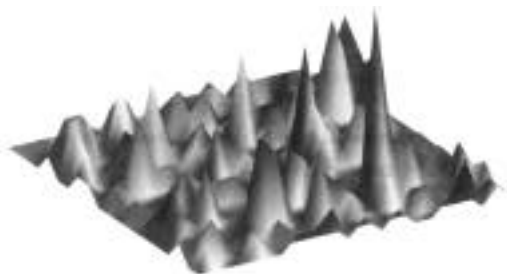


图1-13 10年后世界 **w** 上的人口

例1.10 图形用户界面

MATLAB提供了一个设计易于使用的程序的机会。程序能够在控制图上由按钮、弹出菜单、可编辑文本等控制，这部分在 14.3 节中描述。在 14.4 节中，演示了一个由瑞典科学家

设计的图形用户界面的实例。这个程序用于解决非等距网格的一个模型方程。

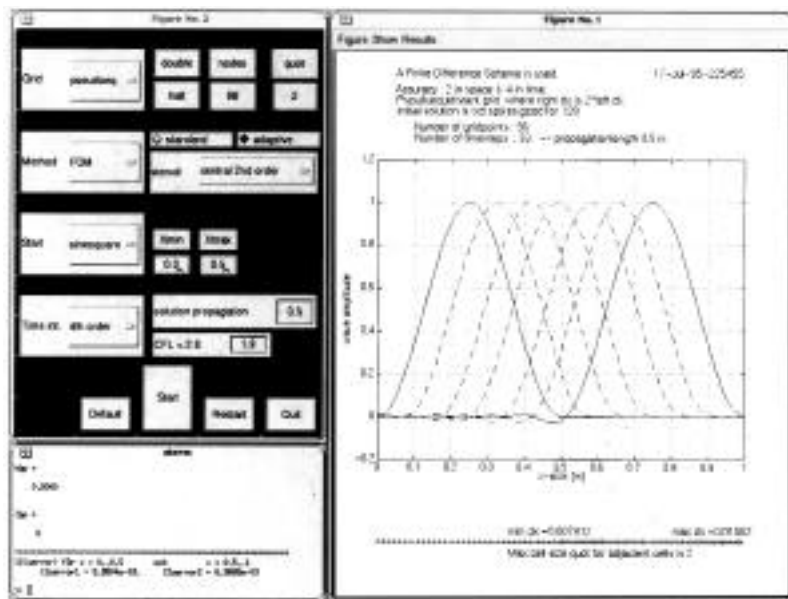


图1-14 带图形用户界面应用的一个 MATLAB 程序

1.3 MATLAB 帮助

本书打算展示 MATLAB 中可用的命令，并解释如何使用 MATLAB。对于读者，多数时候这已足够，但对想学习更多内容的读者来说，在 MATLAB 中可以使用帮助。在系统提示符下，键入 help 命令，MATLAB 给出这个命令的解释。

在 MATLAB 5 中也包含了一个强有力的 HTML 格式的帮助用户数据库。这使得寻找一个命令更容易，依参考而得到帮助。当给出 helpdesk 命令时，一个浏览器如 Netscape Navigator 或 Microsoft Internet Explorer 就运行，即载入一个索引页。

如果这还不能满足，可向 MathWorks 公司订购 MATLAB 的完整手册，具体方法参见前言。