

高等院校信息技术规划教材

基于MATLAB 的计算机图形与动画技术

于万波 编著

INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY

清华大学出版社



高等院校信息技术规划教材

基于MATLAB 的计算机图形与动画技术

于万波 编著

清华大学出版社
北京

内 容 简 介

本书以 Matlab 为工具讲解计算机图形学的基本内容。

第 1 章首先介绍 Matlab 的一些常用命令以及简单的程序设计知识,然后讲解与绘图相关的一些语句及程序;第 2 章与第 3 章讲解曲线曲面绘制与几何造型;第 4 章讲解消隐与光照(第 2 章~第 4 章是经典的计算机图形学内容);第 5 章主要讲解使用一个具体的软件来制作实现虚拟现实效果;第 6 章介绍不规则物体建模与非真实感图形学的概念;第 7 章讲解动画制作的一些概念与常用方法。

本书可以作为高等院校计算机科学与技术专业、信息科学与技术专业、多媒体与动画制作专业以及机械建筑专业本科生及研究生的计算机图形学教材,也可以作为学习 Matlab、数学建模、程序设计的参考书。工程技术人员可以利用该书掌握计算机绘图与动画制作的基本原理。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

基于 MATLAB 的计算机图形与动画技术/于万波编著. —北京:清华大学出版社, 2007.6

(高等院校信息技术规划教材)

ISBN 978-7-302-14954-5

I. 基… II. 于… III. 图像处理—计算机辅助计算—软件包, MATLAB
IV. TP391.41

中国版本图书馆 CIP 数据核字(2007)第 044238 号

责任编辑:袁勤勇 李 晔

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:北京市通州大中印刷厂

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260

印 张:16.25

字 数:373 千字

版 次:2007 年 6 月第 1 版

印 次:2007 年 6 月第 1 次印刷

印 数:1~3000

定 价:20.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:023963-01

前言

foreword

计算机图形学,从广义上讲,是研究与计算机绘制图形相关的一切内容,包括:绘图设备的研发(含驱动程序设计与应用);绘图软件的开发,诸如研究如何开发 Photoshop,开发 3D MAX 等绘图建模软件;如何使用现有的一些绘图软件;研究使用语言(各种高级语言)直接绘图;通用绘图函数软件包(如 OpenGL 等)的开发与使用;一些绘图、建模以及动画制作技术难点和理论问题的研究。不同专业的学生或图形学相关的研究人员,可以根据不同的学习目标,选择某一个或几个方向来进行学习。

计算机绘图与动画技术有着广泛的应用。它的应用领域有计算机辅助设计(CAD)与计算机辅助制造(CAM)、地理信息系统、事务管理与办公自动化、系统及场景模拟、计算机辅助教学(CAI)以及广告和动漫制作等。

专门用来进行绘图的软件有很多,例如,Photoshop、Coldraw、AutoCAD 等。另外,还有一些软件,如 Flash、3D MAX、Maya 等,除了能够完成绘图与建模外,还提供了强大的动画制作功能。除了这些专用软件,诸多的语言也都提供了画图语句,提供了用来制作动画的函数。

其实,语言绘图与软件绘图之间并没有截然的区别,只是在适当的场合,软件绘图更加方便而已。在有些时候,图形动画最好用语言来实现。

本教材讲解如何用 Matlab 语言来绘制二维图形、三维图形以及动画制作;如何用 Matlab 语言来模拟实现绘图软件的功能,例如,实现 Flash 中的逐帧动画、变形动画等。

选择 Matlab 语言,是因为这种语言提供了三维绘制函数,而且是基于矩阵的;Matlab 在图形窗口上和语言中都提供了变换视点等功能,可以直观地从各个角度观察绘制出的三维物体;Matlab 也提供了四维数组函数用来存储序列图像,这些功能非常有利于处理三维图形学中的一些算法。

长期以来,在三维图形学教学中,算法实现是一个难点,选择其



他语言,由于三维算法的复杂性而不易被学生所掌握。如果选择 OpenGL 来处理三维问题,又由于其函数的集成程度过高而隐藏了三维图形的绘制原理。所以,这本教材选用了 Matlab 作为工具来实现算法。

编写这个教材的目的,一是使学生掌握基本的图形动画生成原理及算法,对各个层次的绘图有个贯通的理解;二是让学生掌握一些实用的图形生成技术。

在本书编写的过程中,注意引导学生积极参与,也注意分解图形学中的一些重要算法的难点。书中多以例子解释说明问题,归纳多于演绎,以便做到更加灵活、易读。

本教材可以作为高等院校计算机专业或相关专业图形学课程教材,也可以作为动漫专业学生的参考教材。如果是动漫专业的学生使用,建议除了学习这本教材外,还应学习两类课程,一类是绘图软件及语言绘图,一类是作品创意实践课程。这个教材介于语言使用与创作作品之间,是用来讲解原理与技术的。

本书各章内容安排如下:

在第 1 章的 1.1 节介绍 Matlab,用来讲解 Matlab 的基本语句和程序设计功能,介绍几个与绘图相关的工具箱。在 1.2 节中讲解 Matlab 绘图以及动画制作函数。利用 Matlab 绘制一些简单的二维、三维图形以及一些简单的典型的动画。

第 2 章利用 Matlab 绘制各种曲面,包括使用参数方程绘制,使用代数方程绘制;更多的是研究各种插值与拟合曲面。另外,还研究了 Matlab 样条函数工具箱。

第 3 章研究几何造型,重点介绍几何造型中的实体模型。

在第 4 章中,讲解隐藏面消除问题,首先使用 Matlab 计算法向量,计算面是否隐藏等,然后使用 Matlab 实现了一些简单的光照效果,进而讲解光照原理与光照算法。

在第 5 章中,讲解虚拟现实相关内容,除了了解虚拟现实概念外,还可以学习如何使用软件进行三维形体制作。

在第 6 章 6.1 节中介绍分形这一新兴的图形学分支领域,介绍绘制树、山的方法等。在 6.2 节中介绍基于图像的图形绘制。在 6.3 节中介绍近来新兴起的非真实感图形绘制技术。

第 7 章先介绍常用的三维变换,然后重点研究 Matlab 的动画制作功能,把这些功能与 Flash、3D MAX 等三维动画功能进行比较,加深对三维动画技术的理解。

在使用此教材学习时,应该认真听教师讲解,把读书、思考与上机操作结合起来,其中上机操作最重要。

书中的程序多数是作者自己设计完成的。

书中一定存在很多问题,希望读者指出,非常感谢。

作 者

2007 年 1 月

目录

Contents

第 1 章	Matlab 的绘图功能	1
1.1	Matlab 介绍	1
1.1.1	Matlab 的命令使用与程序设计	1
1.1.2	Matlab 的矩阵操作	16
1.1.3	Matlab 绘图的相关工具箱	24
1.2	Matlab 绘图与动画制作函数	28
1.2.1	二维图形绘制	28
1.2.2	三维图形绘制	38
1.2.3	动画制作方法	42
习题 1		53
第 2 章	曲线与曲面	54
2.1	曲线绘制	54
2.2	典型曲面研究	70
2.2.1	经典的代数曲面	70
2.2.2	插值拟合曲面	74
2.3	Matlab 样条函数	82
习题 2		95
第 3 章	几何造型	97
3.1	图形句柄操作与 GUI 程序设计	97
3.1.1	图形句柄操作	97
3.1.2	GUI 程序设计	108
3.2	几何造型基本单元的组织	113
3.2.1	几何体的绘制	113
3.2.2	几何造型的三种模型	118

3.3 实体模型构造方法	122
3.3.1 边界表示法	122
3.3.2 分解表示法	124
3.3.3 扫描造型法	124
3.3.4 数据文件的存取	127
习题 3	128
第 4 章 隐藏面处理与光照效果	129
4.1 三维空间的观察	129
4.1.1 三维图形显示与视点	129
4.1.2 裁剪	132
4.2 隐藏面处理	133
4.2.1 隐藏面	133
4.2.2 隐藏面计算方法	135
4.3 光照效果	139
4.3.1 简单光照模型	139
4.3.2 Matlab 中的光照函数	148
习题 4	151
第 5 章 虚拟现实	153
5.1 虚拟现实概述	153
5.1.1 VRML-虚拟现实建模语言	153
5.1.2 Matlab 虚拟现实工具箱	155
5.2 V-Realm Builder	157
5.2.1 V-Realm Builder2 常用节点的使用	157
5.2.2 复杂几何体造型节点	171
5.2.3 V-Realm Builder2 的库操作	178
5.2.4 控制方式	181
5.2.5 交互功能的实现	186
5.2.6 V-Realm Builder2 动画制作	188
5.3 Matlab 虚拟现实函数	189
5.3.1 接口函数	189
5.3.2 虚拟世界操作函数	190
习题 5	192
第 6 章 不规则物体建模	194
6.1 分形技术	194
6.1.1 线性迭代函数系统	194

6.1.2 二元二次迭代绘制图形	198
6.1.3 粒子系统	201
6.2 基于图像的图形绘制	202
6.2.1 图像的存取与显示	202
6.2.2 图像的几何操作	207
6.2.3 图像处理与分析	212
6.2.4 图像作为纹理、材质与贴图	215
6.2.5 图像的图形化	216
6.3 非真实感图形绘制技术	217
6.3.1 物体的不规则变换	218
6.3.2 图像像素操作	221
6.3.3 模拟自然介质和艺术手法	223
习题 6	224
第 7 章 动画制作原理	225
7.1 常用的三维变换	225
7.1.1 平移变换与旋转变换	225
7.1.2 比例变换与错切变换	228
7.1.3 Matlab 中与变换相关的函数	230
7.2 基于图像的动画制作	231
7.2.1 利用图像几何操作制作动画	231
7.2.2 基于图像颜色与亮度的动画制作	232
7.2.3 图像处理与分析用于动画制作	235
7.3 动画的分类制作方法	235
7.3.1 逐帧动画、形变动画与路径动画	236
7.3.2 使用语言与软件制作动画	241
7.3.3 动画播放与导出	247
习题 7	249
参考文献	250

Matlab 的绘图功能

1.1 Matlab 介绍

1.1.1 Matlab 的命令使用与程序设计

首先,在机器上安装 Matlab,安装时按照提示选择安装自己需要的工具箱。
安装完 Matlab 后,就可以使用它进行工作了。

1. 命令窗口的使用

许多简单的计算工作都可以在命令窗口中完成,例如:

1) 数的运算

在命令窗口中输入:

```
25 * 4
```

按 Enter 键,则得到结果:

```
ans =
```

```
100
```

输入 $(1+2) * 3^{0.5}$ 按 Enter 键,得到结果 $ans=5.1962$ 。

输入 13^4 ; `sqrt(ans)` 按 Enter 键,得到结果 $ans=169$ 。

输入 $2 * \pi$ 按 Enter 键,得到结果 $ans=6.2832$ 。

输入 $(1+2 * i)(1-i)$ 按 Enter 键,得到结果 $ans=3.0000+1.0000i$ 。

输入 $1/\inf$ 按 Enter 键,得到结果 $ans=0$ 。

上面程序中的 π 是圆周率, i 是复数单位, $\inf$ 是无穷大。在 Matlab 中可以用这些字母来代替有些无理数与复数单位。

2) 向量与矩阵计算

可以在命令窗口中完成矩阵的赋值,输入 $a1=[1\ 2\ 3\ 4;-1\ 0\ 5-2]; a2=[2\ 3;-2\ 2;3\ 0;1-1]$,按 Enter 键,有:

```
a1 =
```



```

1 2 3 4
-1 0 5 -2

a2=
2 3
-2 2
3 0
1 -1

```

输入命令 `a1 * a2`, 按 Enter 键, 则进行矩阵乘法运算, 得到结果:

```

ans=
11 3
11 -1

```

数学上关于矩阵的计算基本上都可以在 Matlab(命令窗口)中进行。例如:

`norm` 计算矩阵范数, `rand` 计算矩阵的秩, `trace` 计算矩阵的迹, `det` 计算矩阵行列式的值等。

向量是矩阵的特例, 矩阵的所有运算函数都可以用在向量上。除了矩阵的通用函数外, 也提供了计算向量的特殊函数, 例如点乘、叉乘函数等。

在命令窗口中输入 `a=[1,2,3]; b=[3,4,5]; dot(a,b)`, 计算向量 a 与 b 的点乘, 输出 26。

如果不清除 workspace 中的变量, 在命令窗口中继续输入命令 `cross(a,b)` (也是函数, 用来计算两个向量的叉乘), 则有 `ans=-2 4 -2`, 得到的是积向量。

【注】 命令 `clear` 用来清除 workspace 中已有的变量, `clc` 用来擦除命令窗口中的内容。

3) 符号运算

使用 Matlab 可以进行多项式乘除运算, 也可以进行因式分解。例如:

要计算 $(x+1)^3$, 在命令窗口中输入 `syms x; expand((x+1)^3)`, 按 Enter 键, 得到结果:

```
ans=x^3+3*x^2+3*x+1
```

因式分解 x^9-1 :

输入 `syms x; factor(x^9-1)`, 按 Enter 键, 得到结果:

```
ans=(x-1)*(x^2+x+1)*(x^6+x^3+1)
```

也可以计算极限导数等, 例如计算极限:

`syms x t; limit(sin(x)/x)` 能计算出极限值为 1, 默认 x 趋近于 0。再输入 `limit((1+2*t/x)^(3*x), x, inf)`, 得到的结果是 `exp(6*t)`, 此时 x 趋近于无穷大。

上面通过数的运算、向量运算与符号运算介绍了 Matlab 的命令窗口的使用。

在命令窗口中执行命令语句可以一句一句执行, 清晰方便。但是, 有很多复杂的工作还是需要进行程序设计。

2. M-File 程序设计

使用菜单项中的文件选项,选择 File→New→M-File 菜单,在打开的窗口中编辑与运行程序。

一般情况下,程序的运行结果显示在命令窗口中(如果绘制图形,那么图形输出在图形窗口中;如果在程序中加入写文件操作,那么可以把运行结果输出到外部文件中)。

程序中的变量以及变量的维数可以在命令窗口左上部的 workspace 中找到。

【例 1-1】 随机生成一个 6 行 10 列矩阵,矩阵的每个元素在 0~1 之间,如果元素大于 0.5,则把这个元素减去 1。

编写程序如下:

```
Ra=rand(6,10);
for i=1:6
    for j=1:10
        if Ra(i,j)>0.5
            Ra(i,j)=Ra(i,j)-1
        end
    end
end
Ra
```

Matlab 的 for 语句与 if 语句的基本规则与其他语言是一样的,只是书写方式稍有差异。

【例 1-2】 使用下面程序绘图,然后修改参数 r 的值,观察图形的变化。

```
y=0.2; x=0.2; z=0.2;
for i=1:10000
    r=-0.2;
    s=0.2;
    q=0.1;
    x1=r-x+x^2-x*y+y+y^2;
    y1=s-x+x*y;
    z1=q-x*y-y*z-z*x;
    plot3(x1,y1,z1)
    x=x1; y=y1; z=z1;
    hold on
end
```

plot3(x1,y1,z1)是在三维空间(x1,y1,z1)位置绘制出一个点。程序一共绘制出了 10 000 个点,有些点距离很近,在机器绘制时重叠在一起。

改变 r 、 s 与 q 的值,能够发现绘制的图形也发生变化,有时变化比较小,有时比较大。这其中变化的规律还有待于研究。

图 1-1 绘制在 Matlab 的图形窗口上。



其实, Matlab 的图形(Figure)窗口不仅可以显示图形,图形窗口还提供了很多操作图形的功能,这些功能通过该窗口中的菜单选项或工具条来实现。

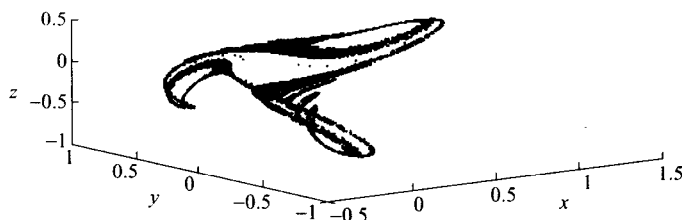


图 1-1 迭代绘制出的图形

3. Figure 窗口的图形操作功能

Figure 窗口主要功能是输出图形。例 1-2 绘制出的图形就是显示在 Figure 窗口中。使用 Figure 窗口的 Edit→Copy Figure 把图形放到剪切板上,再粘贴到 Word 文档中。

图 1-1 在 Copy Figure 之前,先使用 Figure 窗口的 Insert→X Label, Y Label, Z Label 添加了坐标轴标示字 x 、 y 与 z 。

Figure 窗口还有很多操作图形的功能。

1) File 菜单

Save 选项可以把窗口中的图形存储为 .Fig 文件。

Open 选项可以打开已有的 .Fig 文件,图形显示在另一个 Figure 窗口中。

Export 选项可以把图形导出变成其他图像格式,例如,变成 .jpg 格式。

其他的选项还可以对图形页面进行设置,如打印等。

2) Edit 菜单

Figure Properties 选项可以对图形颜色等属性进行修改。

Copy Options 选项可以对复制图形的属性进行设置,例如,把背景色设置为透明色或白色等。图 1-1 是把背景色设置成了白色。

Copy Figure 选项是执行复制功能的。单击这个选项便把设置好的图形放到剪切板上。

3) Insert 菜单

使用这个菜单,可以给图形添加标示,也可以在图形上添加箭头、直线、文字以及颜色条等。

选择 Arrow 选项可以在图形上用鼠标添加箭头。

选择 Line 选项可以在图形上用鼠标绘制直线。

选择 Text 选项可以在图形上添加文字。

4) Tools 菜单

Edit Plot 选项可以对图形进行编辑。

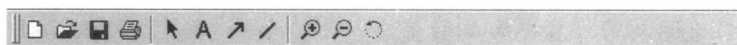
Zoom Out 选项与 Zoom In 选项是用来缩放图形的。

Rotate 3D 选项用来旋转三维图形。在旋转的过程中图形转动,以便从各个角度观察图形。

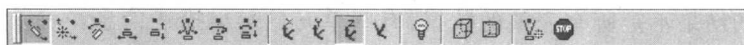
Move Camera 选项用来移动相机。

5) View 菜单与工具条

View 菜单有两个选项,一个是 Figure Toolbar,一个是 Camera Toolbar。默认时(Figure Toolbar 被选中),Figure 窗口中出现如图 1-2(a)所示图形工具条。如果选中 View 菜单中的 Camera Toolbar 选项,那么如图 1-2(b)所示的摄像机工具条就会弹出。



(a) Figure Toolbar(图形工具条)



(b) Camera Toolbar(摄像机工具条)

图 1-2 Matlab 图形窗口工具条

下面对两个工具条上的按钮进行简单解释。

从左到右分别表示 File 菜单中的 New、Open、Save 与 Print 选项。

从左到右分别表示 Tool 菜单中的 Edit Plot 选项,Insert 菜单中的 Text、Arrow 与 Line 选项。

从左到右分别表示 Tool 菜单中的 Zoom In、Zoom Out 与 Rotate 3D 选项。

这些工具用于平移、旋转与缩放等操作。

是主轴选择器。

是用来设置场景灯光的。

左边一个是设置为正交投影,右边一个是设置为透视投影。

左边是重新设置相机与场景灯光,右边是停止相机与场景灯光的运动。

4. 单元数组与结构体

Matlab 是以矩阵为基本单位进行运算的,矩阵与其他语言的数组在广义上是相同的。

Matlab 除了具有数组操作功能之外,还定义了单元数组与结构体等数据类型。

单元数组(cell array)和结构体(structure)这两种数据类型都是将类型不同的相关数据集成到一个变量中来,这使得数据的处理与引用变得简单方便,有利于数据的统一管理。单元数组与结构体的区别是,单元数组中的数据成员是用数字来标识的,结构体中的数据成员是用名称标识的。

1) 单元数组

单元数组中的每个元素称为单元(cell),单元可以包含任何类型的数据。

可以使用两种方法创建一个单元数组,一个是通过赋值语句直接创建;一个是利用 cell 函数先为单元数组开辟一个内存空间,然后再给各个单元赋值。

下面例题只使用第一种方法对单元数组进行赋值。关于 cell 函数以及其他一些复



杂内容可以参考其他 Matlab 教材。

【例 1-3】 单元数组赋值、显示与调用。

```
>> A(1,1)={'这是一个单元数组'};
>> A(1,2)=[1 3 5;2 4 6];
>> A(2,1)=[2+5i 2-5i];
>> A(2,2)={10:-4:0}

A=

    '这是一个单元数组'    [2x3 double]
           [1x2 double]    [1x3 double]
```

上面这些命令完成了一个单元数组的赋值。

显示出的结果不是所有完整的数据,有些单元显示出描述信息。可以使用 `celldisp` 函数显示所有单元内容。

```
>> celldisp(A)
```

按 Enter 键后,显示出:

```
A{1,1}=
    这是一个单元数组
A{2,1}=
    2.0000 + 5.0000i    2.0000 - 5.0000i
A{1,2}=
     1     3     5
     2     4     6
A{2,2}=
    10     6     2
```

可以使用 `A{i,j}` 这种形式对单元数组中的单元内容进行调用。例如:

```
>> b=A{2,1};
>> c=b(1) * b(2)

c=

    29
```

`A{2,1}` 单元中两个复数的乘积是 29。

2) 结构体

结构体与单元数组非常相似,与单元数组不同的是,结构体的组成成员称为字段(fields),结构体采用点号来调用字段中的数据。例 1-4 演示了结构体的创建、显示与调用方法。

【例 1-4】 结构体的创建、显示与调用。

```
>> circle1.r=[0:0.1:1];
>> circle1.center=rand(2,11);
>> circle1.color={'red','green','yellow'}

circle1=
```

```
r: [0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000 0.9000 1]
center: [2x11 double]
color: {'red' 'green' 'yellow'}
```

上述命令创建了一个结构体并且赋值。下面是对结构体成员的调用。

```
>> circle1.center
ans=
0.7919 0.7382 0.4057 0.9169 0.8936 0.3529 0.0099 0.2028 0.6038 0.1988
0.7468
0.9218 0.1763 0.9355 0.4103 0.0579 0.8132 0.1389 0.1987 0.2722 0.0153
0.4451
>> circle1.color(1)
ans=
'red'
>> circle1.center(5,8)
??? Index exceeds matrix dimensions.
>> circle1.center(2,8)
ans=
0.1987
```

5. 自定义函数

Matlab 拥有大量的库函数,也允许用户自定义函数。

把下面两行语句写入 M-File 编辑窗口,保存,自动命名为 f.m。

```
function w=f(x,y,z)
w=x.^3-2*y.^2-2*z+5
```

这两个语句就构造了一个简单的自定义函数。在命令窗口中输入 $f(1,2,3)$, 按 Enter 键,则输出结果为 $w=-8$ 。

也可以在其他程序(或自定义函数)中调用函数 $f(x,y,z)$,例如,下面定义了函数 $fp()$,在该函数中又调用了函数 $f()$ 。

```
function wp=fp(x,y,z)
wp=f(1,2,3)+f(x,y,z)
```

在命令窗口中输入 $fp(1,2,3)$, 按 Enter 键,则输出结果为 $wp=-16$ 。

有些自定义函数没有输出值,只是为了完成一定的功能。例如:

```
function plot2D
x=-2:0.1:2;
y1=sin(x);
y2=cos(x);
y=y1-y2;
plot(x,y)
```



这个函数就是在 $[-2, 2]$ 上绘制函数 $y = \sin(x) - \cos(x)$ 的曲线。

与其他语言一样, Matlab 中的函数具有通用性, 给定参数就能输出函数值, 或者执行一定的工作; 函数具有重用价值, 有些经常使用的程序段都可以作成函数, 以备调用。与有些语言不同, Matlab 中函数的制作与调用更加简单。

【例 1-5】 设计鼠标移动与鼠标按下函数, 然后用这些鼠标函数实现交互操作。

```
function WindowButtonMotion(hObject,eventdata,handles)
global istep
global xdata
global ydata
global h
p=get(gca,'currentpoint');
if(istep==1)
    xdata(2)=p(1);
    ydata(2)=p(3);
    set(h,'XData',xdata,'YData',ydata)
end

function WindowButtonDown(hObject,eventdata,handles)
global istep
global xdata
global ydata
global h
istep=istep+1;
p=get(gca,'currentpoint');
if(istep==1)
    xdata(1)=p(1);
    ydata(1)=p(3);
    xdata(2)=p(1);
    ydata(2)=p(3);
    h=line(xdata,ydata,'EraseMode','xor');
elseif(istep==2)
    xdata(2)=p(1);
    ydata(2)=p(3);
    set(h,'XData',xdata,'YData',ydata,'EraseMode','normal')
    istep=0;
end

function interplot
global istep
global xdata
global ydata
istep=0;
set(gcf,'pointer','cross');
```



```
set(gcf, 'WindowButtonDownFcn', @ WindowButtonDown);
set(gcf, 'WindowButtonMotionFcn', @ WindowButtonMotion);
```

这个程序中有 3 个自定义函数, 分别是 WindowButtonMotion()、WindowButtonDown() 与 interplot。该程序运行后出现空坐标系, 按下鼠标左键选择始点, 将鼠标移动到线段终点, 再单击左键, 就完成了—条直线段的绘制。结果如图 1-3 所示。

在下面交互控制中对程序进行详细讲解。

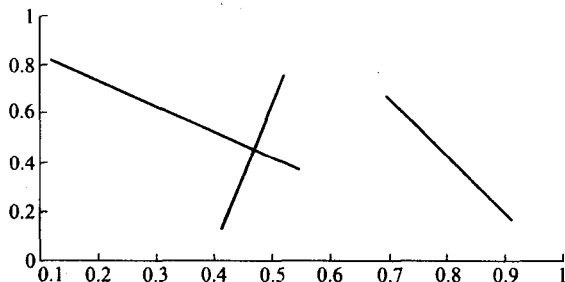


图 1-3 鼠标交互绘制直线段

6. 交互控制

交互控制是计算机语言的一个重要组成部分。最常用的交互控制是鼠标控制与键盘控制。

1) 鼠标控制

鼠标控制可以用 get 函数与 set 函数来完成。

在例 1-5 中, 第一个函数(鼠标移动函数) WindowButtonMotion() 中, 关键语句是 `p=get(gca, 'currentpoint')`, 该语句返回鼠标当前位置的坐标, 坐标值赋给变量 `p`。 `gca` 也是一个命令, 用来返回当前图形窗口中当前坐标轴的句柄; `currentpoint` 是命令参数(关键字), 表示鼠标当前位置。

在函数 WindowButtonDown() 中, 语句 `istep=istep+1` 通过后面程序确定是鼠标移动还是按下鼠标, 该函数也有返回当前鼠标的语句, 用来确定线段的起点与终点。

在主函数 interplot 中, 首先使用语句 `set(gcf, 'pointer', 'cross')` 把鼠标设置成为十字形, 所以, 运行主程序后, 鼠标呈十字形状; 然后在语句 `set(gcf, 'WindowButtonDownFcn', @WindowButtonDown)` 中调用函数 WindowButtonMotion, 让该函数在鼠标按下时执行; 最后, 调用函数 WindowButtonMotion, 让该函数在鼠标移动时执行。

程序中 `istep`、`xdata`、`ydata` 与 `h` 是定义的 4 个全局变量, 程序中其他函数也可以调用这些全局变量。

鼠标移动与按下鼠标左键是常用的两个功能, 其他功能(如按下鼠标右键等)也是用上面类似的方法实现。

2) 键盘控制

命令 `T=waitforbuttonpress` 能使程序停止下来, 等待键盘按键或者鼠标动作。按键盘的任意键, 程序继续执行, 返回的 `T` 值为 1; 按下鼠标任意键, 程序继续执行, 返回的 `T`