

MATLAB
开发实例系列图书

MATLAB

数值计算

案例分析

刘寅立 王剑亮 陈靖 刘衍琦 王光辉 史峰 编著

登录<http://www.matlabsky.com>/注册
用户名，验证密码后即可与作者在线交流

卡号：2011000531226

密码：



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS



MATLAB 开发实例系列图书

MATLAB 数值计算案例分析

刘寅立 王剑亮 陈 靖 编著
刘衍琦 王光辉 史 峰

北京航空航天大学出版社

内 容 简 介

本书系统讲解了数值分析的方法与理论以及基于 MATLAB 软件的编程实现,全书共 12 章,内容包括 MATLAB 编程基础、数据插值、数据拟合、数值积分、常微分方程、线性方程组迭代解法、线性方程组的直接解法、非线性方程求解、偏微分方程数值解、数值优化、特征值和特征向量等。

本书以数值方法原理为主线,以 MATLAB 在数值分析中的应用为主要分析对象,在讲解数值分析算法的原理和基本思想的基础上,侧重于基于 MATLAB 软件的各种算法的实现。

本书适合高年级本科生、研究生以及相关研究人员使用。

图书在版编目(CIP)数据

MATLAB 数值计算案例分析/刘寅立等编著. --北京
: 北京航空航天大学出版社,2011. 10
ISBN 978-7-5124-0547-9

I. ①M… II. ①刘… III. ①计算机辅助计算—软件包, MATLAB IV. ①TP391.75

中国版本图书馆 CIP 数据核字(2011)第 151115 号

版权所有,侵权必究。

MATLAB 数值计算案例分析

刘寅立 王剑亮 陈 靖 刘衍琦 王光辉 史 峰 编著
责任编辑 罗晓莉

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:goodtextbook@126.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787 mm×1 092 mm 1/16 印张:14.75 字数:378 千字

2011 年 10 月第 1 版 2011 年 10 月第 1 次印刷 印数:5 000 册

ISBN 978-7-5124-0547-9 定价:30.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前 言

在科学和工程领域,科学计算是不可缺少的重要环节。然而,如高次代数方程求根,微分方程求解,复杂函数的积分,非线性最优化等,诸多问题的解析解或解析表达式要么无法给出,要么非常复杂而不便于计算,为解决这些问题,需要采用近似的计算方法——数值方法来求解这些问题。因此,数值分析是科学研究和工程计算领域的一门重要学科,研究的主要内容包括数据插值、拟合、数值积分、数值微分、微分方程求解、线性方程组、方程(组)求根、数值优化、特征值与特征向量等。

近年来,随着计算机技术的快速发展,使用计算机进行科学计算已经成为科学研究中不可缺少的环节。伴随着计算工具的进步,各种数值计算软件层出不穷,MATLAB 软件即为其中的佼佼者,该软件界面简洁,编程快捷,包含功能强大的函数和工具箱,特别适用于数学建模和科学计算,能够解决各种数值分析问题。

本书以数值分析理论为主线,以 MATLAB 在数值分析中的应用为主要内容,在介绍数值分析算法原理和基本思想的基础上,侧重于讲解基于 MATLAB 软件的各种算法的实现。全书共分为 12 章,分别讲解了 MATLAB 基础知识、数据插值、数据拟合,数值积分、常微分方程、线性方程组迭代解法、线性方程组的直接解法、非线性方程求解、偏微分方程数值解、数值优化、特征值和特征向量等。每章内容可以分为两个部分,讲解介绍数值分析的原理部分及数值分析方法的 MATLAB 实现。

本书适合高年级本科生、研究生以及相关研究人员使用。读者在阅读此书时,可以结合程序,一边运行程序,一边从书中寻找每段程序的功能以及原理,并且代入自己的数据和模型。

本书由刘寅立、王剑亮、陈靖、刘衍琦、王光辉和史峰编著,其中刘寅立完成第 7、8、10、12 章,王剑亮完成第 1、4、5、6 章,陈靖完成第 2、9 章,刘衍琦完成第 3、11 章,王光辉负责书中各例题的选取及程序的进一步调试工作,全书由史峰和刘寅立负责统稿。

本书在写作的过程中,得到了 MATLABSKY 论坛(www.matlabsky.com)的大力支持。MATLABSKY 论坛为本书开辟了读者交流版块,作者长期在线解答读者的各种疑问。我们相信,只有交流才会有进步,只有碰撞才会有火花。

感谢天津科技大学理学院,黑龙江科技学院计算机与信息工程学院,东方电子有限公司,华中科技大学机械学院的同事、同学们及笔者的家人们对编者工作的支持,尤其感谢中华老师对本书的关心与指导,在成书过程中,谢老师倾注了极大的热情并提出了宝贵的意见和建议。

由于作者水平有限,书中尚存缺点和遗漏之处,恳请读者提出宝贵的意见和建议,以便于我们完善和提高。

编 者

2011 年 3 月

目 录

第 1 章 MATLAB 编程基础	1
1.1 矩阵的基本操作与基本运算	1
1.1.1 矩阵的基本操作	1
1.1.2 矩阵的基本运算	2
1.1.3 * 与 .* 和 / 与 ./ 的区别	3
1.1.4 使用 find 函数索引符合某些特定条件的矩阵元素	3
1.1.5 eps 函数与避免除以 0 的方法	4
1.2 MATLAB 的数据结构	4
1.3 变量、脚本与函数	8
1.3.1 变 量	8
1.3.2 全局变量使用例子	9
1.3.3 局部变量不会被替代的例子	10
1.3.4 函数与脚本	10
1.3.5 函数的构成	11
1.3.6 函数的类型	12
1.3.7 函数调用与函数句柄	14
1.3.8 可变参数函数调用	14
1.4 MATLAB 技巧	15
1.4.1 MATLAB 的函数重载	15
1.4.2 冒号(:)操作符	17
1.4.3 Tab 键自动补全	17
1.4.4 上下箭头回调	17
1.4.5 可变参数个数的函数的占位符	17
1.4.6 whos 查看	18
1.4.7 whos 通配符的例子	18
1.4.8 程序调试	18
1.5 MATLAB 工具箱函数 ode23 剖析	18
1.6 MATLAB 的帮助文档导航	22
1.7 MATLAB 常见错误	23
1.7.1 常见写法错误	23
1.7.2 字符串连接出错	24
1.7.3 矩阵维数不同的例子	25
1.7.4 赋值出错	26

第 2 章 数值分析的基本概念	27
2.1 数值分析的研究对象	27
2.2 误差与有效数字	30
2.2.1 误差的产生及分类	30
2.2.2 误差的相关概念	30
2.3 近似计算中的注意事项	31
2.4 数值算法的稳定性	34
2.5 机器精度	35
第 3 章 数据插值	37
3.1 插值与多项式插值	37
3.2 Lagrange 插值	37
3.2.1 Lagrange 插值的定义	37
3.2.2 Lagrange 插值的 MATLAB 实现	38
3.3 Newton 插值	40
3.3.1 Newton 插值定义	40
3.3.2 有限差商	40
3.3.3 Newton 插值的 MATLAB 实现	41
3.4 Hermite 插值	42
3.4.1 Hermite 插值定义	42
3.4.2 Hermite 插值的 MATLAB 实现	43
3.5 分段低次插值	45
3.5.1 高次插值的 Runge 现象	45
3.5.2 分段低次 Lagrange 插值	45
3.5.3 interp1 函数	46
3.6 三次样条插值	47
3.6.1 三次样条插值	47
3.6.2 三次样条函数	48
第 4 章 数据拟合	50
4.1 数据的曲线拟合	50
4.1.1 曲线拟合的误差	50
4.1.2 曲线拟合的最小二乘法	51
4.2 多项式拟合	52
4.2.1 多项式曲线拟合	52
4.2.2 多项式曲线拟合的 MATLAB 实现	52
4.2.3 MATLAB 多项式曲线拟合应用的扩展	54
4.3 圆拟合的例子讲解	57
4.3.1 圆拟合问题描述(使用最小二乘方法)	57
4.3.2 圆拟合的 MATLAB 实现	58

4.4	cftool 自定义拟合	60
4.5	cftool 代码自动生成与修改	62
第 5 章	数值积分	66
5.1	数值积分的基本思想	66
5.1.1	数值求积的基本思想	66
5.1.2	几种常见的数值积分公式	66
5.2	数值求积公式的构造	67
5.2.1	代数精度	68
5.2.2	插值型求积公式	68
5.2.3	Newton - Cotes 求积公式	69
5.3	复化积分公式	70
5.3.1	复化 Simpson 公式	70
5.3.2	复化求积公式及其 MATLAB 实现	70
5.3.3	MATLAB 的 trapz 函数	72
5.4	Romberg 求积公式	73
5.4.1	数值积分公式误差分析	73
5.4.2	Romberg 算法	74
5.4.3	Romberg 求积公式的 MATLAB 实现	76
5.5	Gauss 求积公式	77
5.5.1	Gauss 积分公式	77
5.5.2	Gauss - Legendre 求积公式的 MATLAB 实现及应用实例	78
5.6	积分的运算选讲	79
5.6.1	二重积分	79
5.6.2	三重积分	79
5.6.3	变上限积分	79
5.6.4	符号积分	81
5.6.5	MATLAB 常见积分函数列表	82
第 6 章	常微分方程	83
6.1	常微分方程分类及其表示形式	83
6.1.1	MATLAB 关于 ODE 的函数帮助简介	83
6.1.2	MATLAB ODE suite 中关于 ODE 的分类	83
6.2	典型常微分方程举例	84
6.2.1	一阶常微分方程	84
6.2.2	二阶常微分方程	84
6.2.3	高阶常微分方程	85
6.2.4	边值问题	85
6.2.5	延迟微分方程	85
6.3	解的存在性、唯一性和适定性	86

6.3.1	初值问题的存在性与唯一性	86
6.3.2	MATLAB 中常微分方程的通用形式及其向量表示	87
6.3.3	刚性常微分方程	87
6.4	常微分方程的时域频域表示以及状态方程表示	89
6.4.1	时域与频域表示形式	89
6.4.2	状态空间表示形式	90
6.5	单步多步和显式隐式概念	91
6.6	常微分方程数值求解方法构造思想举例	92
6.7	常微分方程数值解的基本原理	93
6.7.1	一阶常微分方程与一阶微分方程组	93
6.7.2	求解区间[a,b]的离散	93
6.7.3	微分方程的离散	93
6.7.4	Taylor 展开法	94
6.7.5	常微分方程数值求解的欧拉方法	97
6.7.6	欧拉方法的 MATLAB 实现	97
6.7.7	改进的欧拉方法	98
6.7.8	改进的欧拉方法的 MATLAB 实现	99
6.7.9	四阶龙格-库塔公式的 MATLAB 实现	99
6.7.10	Adams 预测-校正公式	100
6.8	常微分方程工具箱	102
6.8.1	总体介绍	102
6.8.2	各个求解器的特点与比较	103
6.8.3	使用 odefile. m 模板求解常微分方程	103
6.8.4	odefile. m 模板使用	105
6.9	单自由度振动系统例子	106
6.9.1	单自由度二阶系统基于传递函数与状态空间的 simulink 模型求解	106
6.9.2	总结	110
6.10	三自由度振动系统例子	110
6.10.1	三自由度振动系统 simulink 模型求解以及状态方程的 ode45 求解器求解	110
6.10.2	总结	114
第 7 章	线性方程组的迭代解法	115
7.1	线性方程组的迭代法概述	115
7.1.1	迭代法概述及压缩原理	115
7.1.2	迭代法基本概念	115
7.1.3	MATLAB 的相关命令	117
7.2	常见的线性方程组的迭代法	118
7.2.1	Jacobi 迭代法	118
7.2.2	Gauss-Seidel 迭代法	120

7.2.3	SOR 迭代法	123
7.3	迭代法的收敛性	125
7.3.1	迭代法的收敛性定理	125
7.3.2	主对角优势	125
7.3.3	SOR 迭代法的收敛性	126
第 7 章	线性方程组的直接解法	127
8.1	线性方程组的消元法	127
8.1.1	线性方程组的直接求解方法	127
8.1.2	Gauss 消去法	127
8.1.3	Gauss 主元素法	130
8.1.4	Jordan 消去法	133
8.2	矩阵的三角分解	135
8.2.1	LU 分解	136
8.2.2	LU 分解的 MATLAB 实现	136
8.2.3	对称正定矩阵的 Cholesky 分解	138
8.2.4	Cholesky 分解法的 MATLAB 实现	139
8.2.5	改进平方根法	141
8.2.6	改进平方根法的 MATLAB 实现	142
8.3	MATLAB 的相关命令	144
8.3.1	逆矩阵	144
8.3.2	矩阵的左除及最小二乘解	145
8.3.3	欠定方程的解	145
第 9 章	非线性方程求解	147
9.1	求解非线性方程的 MATLAB 符号法	147
9.2	二分法	149
9.2.1	二分法原理	149
9.2.2	二分法的 MATLAB 程序	149
9.3	迭代法	151
9.3.1	迭代法原理	151
9.3.2	迭代法的几何意义	151
9.3.3	迭代法的 MATLAB 程序	152
9.4	切线法	153
9.4.1	切线法的几何意义	154
9.4.2	切线法的收敛性	154
9.5	割线法(弦截法)	155
9.5.1	割线法的几何意义	155
9.5.2	割线法的 MATLAB 程序	155
9.6	常见非线性方程数值方法的优缺点	156

9.7	方程 $f(x)=0$ 数值解的 MATLAB 实现	157
9.7.1	求函数零点指令 fzero	157
9.7.2	fzero 的使用举例	157
9.8	求解非线性方程组 MATLAB 命令	160
9.8.1	符号方程组求解	160
9.8.2	求解非线性方程组的基本方法	161
9.8.3	求方程组的数值解	162
第 10 章	偏微分方程数值解	166
10.1	基本概念	166
10.2	有限差分法	167
10.2.1	椭圆方程的差分形式	167
10.2.2	抛物方程的差分形式	168
10.2.3	双曲方程的差分形式	170
10.3	MATLAB 的 pdepe 函数	171
10.3.1	pdepe 函数的说明	171
10.3.2	pdepe 函数的实例	172
10.4	MATLAB 的 PDEtool 工具箱	173
10.4.1	PDEtool 的界面	174
10.4.2	PDEtool 的使用	174
第 11 章	数值优化	177
11.1	单变量函数优化	177
11.1.1	基本数学原理	177
11.1.2	黄金分割法	178
11.1.3	牛顿法	181
11.1.4	最速下降法	185
11.1.5	共轭梯度法	188
11.2	多变量函数优化	191
11.2.1	Nelder-mead 方法	191
11.2.2	Nelder-mead 方法的 MATLAB 实现	192
11.2.3	Powell 方法	193
11.2.4	Powell 方法的 MATLAB 实现	194
11.3	MATLAB 最优化函数	197
11.3.1	MATLAB 最优化工具箱介绍	197
11.3.2	MATLAB 最优化函数介绍	198
11.3.3	MATLAB 最优化工具介绍	201
11.3.4	MATLAB 最优化函数应用实例	204
第 12 章	特征值和特征向量	208
12.1	特征值与特征向量	208

12.1.1	特征值与特征向量的定义	208
12.1.2	特征值与特征向量的计算	208
12.1.3	MATLAB 的 eig 命令	209
12.2	幂法与反幂法	209
12.2.1	幂法的原理	210
12.2.2	幂法的 MATLAB 实现	210
12.2.3	反幂法	212
12.2.4	反幂法的 MATLAB 实现	213
12.3	对称矩阵的特征值——Jacobi 方法	214
12.3.1	Jacobi 方法的原理	214
12.3.2	Jacobi 方法的 MATLAB 实现	215
12.4	Householder 方法	217
12.4.1	初等反射矩阵	218
12.4.2	用正交相似变换约化矩阵	218
12.4.3	算法的 MATLAB 实现	220
12.5	QR 分解与 QR 方法	221
12.5.1	矩阵的 QR 分解	221
12.5.2	计算矩阵特征值的 QR 方法	222
12.5.3	QR 方法的 MATLAB 实现	222
	参考文献	224

第 1 章

MATLAB 编程基础

本章写作目的:①提供最核心的 MATLAB 基础知识;②结合实例对 MATLAB 用于数值计算中的难点和易错易混淆的问题进行讲解,力求清除读者阅读和编写 MATLAB 数值计算程序中的障碍;③指出 MATLAB 帮助文档中与数值计算关系最为紧密的内容。当读者对某些问题想进一步了解时,可以寻找帮助中相关章节的内容。作者鼓励读者查阅 MATLAB 自带的帮助文档,这是第一手的资料,是其他文档所不能替代的。本章主要内容分为以下四个部分。

第一部分 对 MATLAB 基本的矩阵操作函数,变量、脚本、函数等概念以及 MATLAB 的数据结构等内容进行了讲解。这部分是 MATLAB 的核心基础知识,包括 1.1、1.2 和 1.3 节内容。

第二部分 对 MATLAB 的一些操作技巧进行讲解,同时还给出了 ode23 函数的详细剖析,包括 1.4 和 1.5 节内容。

第三部分 在 1.6 节中作者给出了帮助文档的导航,指出了数值分析中所涉及的内容。

第四部分 在 1.7 节中,作者通过实例给出了一些编程中常见的错误分析。

1.1 矩阵的基本操作与基本运算

1.1.1 矩阵的基本操作

矩阵的基本操作包括创建矩阵,矩阵赋值以及矩阵数据索引等,如表 1-1 所列。

表 1-1 矩阵的基本操作

分类	功能	函数	示例
矩阵创建	直接指定矩阵元素	[,],[;],[]	A=[1 2 3];A=[1;2;3]; A=[1,2,3;2,3,4];
	冒号操作符	:	A=1:0.1:10;
	等差数列	linspace	A=linspace(1,3,10);
矩阵赋值	整体赋值	B=A	A=[1,3,4],B=A;
	单个元素赋值	A(i,j)=x;	A(3,4)=5;
	多个元素赋值	A(i1:i2;j1:j2)=B;	A(1:2,[3,4,7])=zeros(2,3);
创建特殊矩阵	生成全 1 的矩阵	ones	A=ones(3);A=ones(1,3);
	生成单位矩阵	eye	A=eye(3);
	生成全 0 的矩阵	zeros	A=zeros(3);
	生成随机矩阵	rand	A=rand(3,4);
	给定对角元素生成矩阵	diag	diag(1:3);
	生成 3-D 图所需要数据以及多维数据	meshgrid,ndgrid	[xx,yy]=meshgrid(1:10,1:10); [xx,yy,zz]=ndgrid(1:10,1:10,1:10);

分类	功能	函数	示例
删除矩阵元素	删除单个或者多个矩阵元素	[]	A=[],A(:,3)=[];
矩阵的连接	水平方向连接	horzcat,[A,B]	A=magic(3);B=zeros(3); C=horzcat(A,B)或C=[A,B];
	垂直方向连接	vertcat,[A;B]	A=magic(3);B=zeros(3); C=vertcat(A,B)或C=[A;B];
矩阵变形操作	转置	'或 transpose	B=A';
	旋转 90°	rot90	B=rot90(A,3);
	左右位置互换	fliplr	B=fliplr(A);
	上下位置互换	flipud	B=flipud(A);
	赋值矩阵	repmat	B=repmat(A,1,3);
	变为一维列向量	(:)	B=A(:);
	重新形成矩阵维数	reshape	A=reshape(B,3,3);
索引操作	单元素索引	A(i,j)	a=A(1,3);
	行索引	A(i,:)	b=A(1,:);
	列索引	A(:,j)	c=A(:,1);
	当成列向量索引单个元素	A(m)	d=A(7);
	逻辑矩阵索引	A(ind)	B=A(find(A)>3);
	列向量索引变为下标索引	ind2sub	[i,j]=ind2sub([3,3],7);
	下标索引变为列向量索引	sub2ind	ind1=sub2ind([3,3],1,3);

1.1.2 矩阵的基本运算

矩阵的基本运算包括常见的算符,初等函数以及线性代数运算等,如表 1-2 所列。

表 1-2 矩阵的基本运算

分类	功能	函数	实例
三角函数	正弦与反正弦	sin,asin	sin(pi),asin(0.5);
	余弦与反余弦	cos,acos	cos(pi),acos(0.5);
	正切与反正切	tan,atan	tan(pi),atan(0.5);
	余切与反余切	cot,acot	cot(pi),acot(0.5);
指数函数	以 e 为底的指数和对数	exp,log	exp(3),log(3);
	以 10 为底的对数	log10	log10(1);

续表 1-2

分类	功能	函数	实例
复数函数	复数幅值	abs	abs(3+4i);
	复数实部	real	real(3+4i);
	复数虚部	imag	imag(3+4i);
常见算符	加与减	+, -	A+3, B-A;
	矩阵乘法	*	A*B;
	矩阵对应元素相乘	.*	A.*B;
	矩阵左除与右除	\与/	A\B, A/B;
	矩阵对应元素相除	.\与./	A.\B;
	矩阵乘方	^	A^3;
	矩阵对应元素的幂	.^	A.^3;
矩阵基本函数	矩阵对应元素的幂	exp	exp(i*pi);
	矩阵的指数函数	expm	expm(A);
	矩阵对应元素的对数	log	log([1 2; 1 2]);
	矩阵的对数函数	logm	logm([3, 3; 3 4]);
	矩阵的行列式	det	det(eye(3));
	矩阵的秩	rank	rank(eye(3));
	矩阵的特征值与特征向量	eig	[x, l]=eig(eye(3));
	矩阵的范数	norm	norm(eye(3));
	矩阵的条件数	cond	cond(hilb(5));
	矩阵的奇异值分解	svd	[u, v, w]=svd(rand(3,5));

值得读者注意的是,表 1-1 和表 1-2 只是对 MATLAB 常用基本操作和基本函数的讲解。更为全面的介绍,请读者查阅 help matlab\ops(算符及特殊符号),help matlab\elmat(基本的矩阵操作),help matlab\elfun(初等函数),help matlab\matfun(线性代数),help matlab\specfun(特殊函数),help matlab\funfun(泛函以及常微分函数)等内容。

1.1.3 * 与 .* 和 /与 ./ 的区别

“.*”的英文解释是 elementwise times,即按对应元素相乘,而“*”的英文解释是 matrix times,是矩阵乘法。输入命令 doc mtimes,可以打开帮助文档中对矩阵乘法的定义。

$$C=A * B \quad (1.1)$$

$$C(i, j) = \sum_{k=1}^p A(i, k) B(k, j) \quad (1.2)$$

式中, A 是大小为 $m \times p$ 的矩阵; B 是大小为 $p \times n$ 的矩阵; C 是大小为 $m \times n$ 的矩阵。矩阵 C 中每个元素等于 $A(i, :)*B(:, j)$,显然矩阵乘法要求 A 的列数和 B 的行数相等,对于式(1.2)即要求 p 相等。

1.1.4 使用 find 函数索引符合某些特定条件的矩阵元素

以 sinc 函数为例,使用 open sinc 命令可以打开 sinc 函数的代码,摘录如下。

```
function y = sinc(x)
% SINC Sin(pi * x)/(pi * x) function.
i = find(x == 0);
x(i) = 1;          % From LS: don't need this is /0 warning is off
y = sin(pi * x)./(pi * x);
y(i) = 1;
```

其中“ $i = \text{find}(x == 0)$ ”中 $x == 0$ 是逻辑判断, find 从逻辑判断矩阵 ($x == 0$ 的返回值) 中找到“1”所在的位置索引, 并赋给变量 i , 然后令 x 中 i 位置的元素 $x(i) = 1$ 。从该例子中可以看到, 结合逻辑判断和 find 函数可以很方便地找到满足某些条件(例如逻辑判断条件)的元素的位置, 并对这些位置的元素进行相应地操作。

1.1.5 eps 函数与避免除以 0 的方法

eps 函数可以得到浮点数的相对精度, 也就是相邻两数之间的最小距离。例如 $1e20$ 与相邻数的距离是 16384。

```
>> eps(1e20)
ans =
    16384
```

用 eps 函数代替 0, 可以避免除以 0, 如以下的 mysinc 函数:

```
function y = mysinc(x)
% SINC Sin(pi * x)/(pi * x) function.
i = find(x == 0);
x(i) = eps(0) + x(i);    % 使用 eps(0)代替 0
y = sin(pi * x)./(pi * x);
```

作者自编的 mysinc 函数与系统函数 sinc 函数实现同样的功能。系统自带的 sinc 函数采用了另外一种策略, 从以下的结果可以看出, 两者结果一致。从而验证了使用 eps 函数策略的有效性。

```
>> x = [0; 0.1; 0.4; 0]
x =
     0     0.1000     0.2000     0.3000     0.4000     0
>> mysinc(x)
ans =
    1.0000    0.9836    0.9355    0.8584    0.7568    1.0000
>> sinc(x)
ans =
    1.0000    0.9836    0.9355    0.8584    0.7568    1.0000
```

1.2 MATLAB 的数据结构

算法+数据结构=程序(《Algorithms+Data Structures=Programs》——Niklaus Emil Wirth)
数据结构在编程中具有基础地位, 任何操作都是基于数据的操作。数据是操作的对象和

核心。MATLAB 的操作主要是矩阵操作，而矩阵运算对矩阵的维数有很多要求。因此在 MATLAB 中进行数据操作时，对数据的类型和数据的维数必须非常清楚，尤其是矩阵的维数。例如 `det` 函数，就要求操作数是方阵。

```
a = [1 2
     3 4
     5 6]
a =
     1     2
     3     4
     5     6
>> det(a)
??? Error using ==> det
Matrix must be square.
```

提示出错，矩阵必须是方阵(square array)。

矩阵与向量

MATLAB 中的数据是以矩阵为基础的，标量(scalar)和向量(vector)是特殊情形。标量是 1×1 的矩阵，而向量分列向量(大小为 $n \times 1$)和行向量(大小为 $1 \times n$)。

MATLAB 中几乎所有的数据都以矩阵(array 或 matrix, 在 MATLAB 中这两个术语经常可以替换使用)的形式出现，这区别于其他的高级语言，例如 C++。这也是它被称为第四代语言(the 4th language)的原因之一。

字符串

字符串在编程中具有基础地位，文件的操作以及其他命令的构造等，都依赖于字符串，因此字符串在各种编程语言中都属于基础核心类。MATLAB 中字符串以单引号或者 `string()` 函数定义。注意是单引号不是双引号，有别于 C 语言。字符串在许多地方都有应用，如文件操作，符号计算，属性设置等。这里仅仅给出构造字符串用于保存文件一例。

```
>> A = 0;0.01;10;
>> datFileName = 'dataSin'
>> datFileName = [datFileName,num2str(1),'.mat']
datFileName =
dataSin
datFileName =
dataSin1.mat
>> saveFileStr = [' save(''',datFileName,'','',',',' A ''',')']
saveFileStr =
save('dataSin1.mat','A')
>> eval(saveFileStr)
>> clear
>> load dataSin1
>> plot(A,sin(A),'--k')
```

以上代码没有直接使用 `save` 函数保存数据，而是构造字符串并使用 `eval` 执行字符串来保存数据，采用这种方式就可以任意构造文件名，或者根据程序需要更改文件名，例如上面代码中的 `num2str(1)`，若 1 用变量 `i` 代替，则文件名就可以因 `i` 的不同而不同。这就是构造字符串

的好处。代码中的 `clear` 清除了所有的变量,使用 `load dataSin1` 读取存储过的数据,然后使用 `plot` 作图,保存的数据如图 1-1 所示。

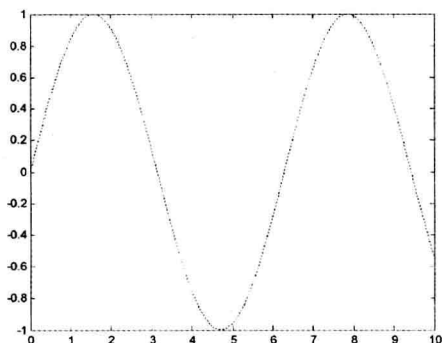


图 1-1 使用字符串读入文件画的正弦函数图

元胞数组

元胞数组因为存储内容长度任意,很适合处理复杂的字符串。向 MATLAB 导入数据以及做一些面向对象的操作,或者混合处理多种数据时,经常使用元胞数组。元胞数组的创建可以使用 `cell` 函数或者“{}”(大括号),实际上“{}”是对 `cell()` 函数的重载。

元胞数组的元素索引是比较特殊的,要结合“()”“{}”使用。“()”返回元胞元素,而“{}”返回元胞元素的内容。有趣的是可以使用 `cellplot` 显示元胞数组的结构,详见以下程序注释部分。

```
>> A1 = diag([1,2])
A1 =
    1     0
    0     2
>> A2 = diag([3,4,5])
A2 =
    3     0     0
    0     4     0
    0     0     5
>> B1 = {A1,A2} % 使用“{}”创建元胞数组
B1 =
    [2x2 double]    [3x3 double]
>> B1(1) % 使用“()”索引元胞元素
ans =
    [2x2 double]
>> B1{1} % 使用“{}”索引元胞元素
ans =
    1     0
    0     2
>> cellplot(B1) % 画出元胞的结构示意图
```

元胞的结构示意图如图 1-2 所示。

建议使用 `doc cell` 命令查看具体的帮助。

结构数组

与元胞数组类似,结构数组也可以存储各种类型的数据。不同的是它使用域名来访问元