

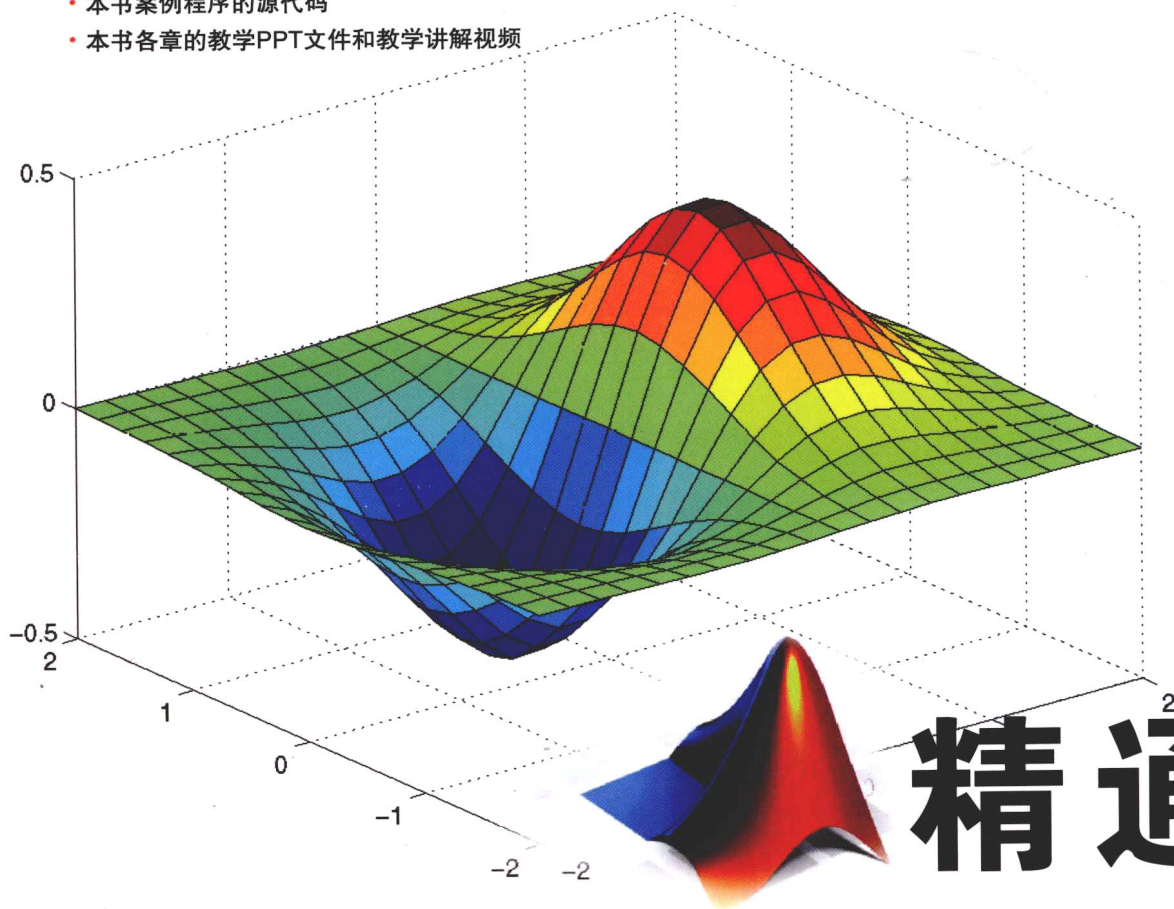
全面讲解MATLAB与C++混合编程的主流技术
提供近50个实例项目，介绍MATLAB与C++的7大混合编程应用

配2个紧密结合工程实际的项目案例：

- 工程开发实践——齿轮优化设计系统
- 汽车仿真应用——汽车ABS系统仿真

超值光盘：

- 本书案例程序的源代码
- 本书各章的教学PPT文件和教学讲解视频



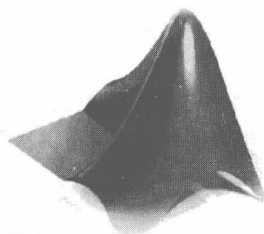
精通

MATLAB混合编程

◎ 丁毓峰 编著



10小时多媒体教学视频



精通

MATLAB混合编程

◎ 丁毓峰 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书详细讲解 Visual C++和 MATLAB 混合编程各项技术和重点应用。本书从混合编程环境的搭建、混合开发中 Visual C++和 MATLAB 必备基础知识讲解,详细讲解六种混合开发方式: Visual C++调用 MATLAB Engine 库、Visual C++调用 MATLAB 的 C/C++数学函数库、基于数据文件交换、基于 COM 技术、使用 MATCOM、使用 ActiveX 技术,还讲解了七大混合编程应用领域:科学运算、图形图像显示、图像识别、控制系统模型输入和分析、控制系统的设计仿真、信号频谱分析和数据采集和分析。最后,本书还讲解了两个案例: Visual C++和 MATLAB 开发齿轮优化设计系统, Visual C++和 MATLAB 的汽车 ABS 系统仿真等内容。

本书配套光盘提供了每章实例的源程序。本书不仅适合高等学校理工类研究生或者高年级本科生作为学习 Visual C++和 MATLAB 混合编程的教材,也可供从事 MATLAB 进行工程设计和仿真的技术人员参考使用。同时书中提供的大量实例也可供高级用户参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

精通 MATLAB 混合编程 / 丁毓峰编著. —北京: 电子工业出版社, 2012.6
ISBN 978-7-121-16895-6

I. ①精… II. ①丁… III. ①Matlab 软件②C 语言—程序设计 IV. ①TP317②TP312

中国版本图书馆 CIP 数据核字(2012)第 084897 号

策划编辑: 胡辛征

责任编辑: 徐津平

文字编辑: 张丹阳

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 26.25 字数: 672 千字

印 次: 2012 年 6 月第 1 次印刷

印 数: 4000 册 定价: 59.00 元(含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlls@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

Visual C++作为 Windows 系统应用程序的重要开发工具，自从发布后一直得到广大程序开发人员的青睐，程序员使用该工具在 Windows 平台上开发了众多的应用程序。MATLAB 作为一个科学计算可视化工具，经过 20 多年的发展，已经成为应用最广泛的科学处理工具软件包，为科学研究、工程设计等众多科学领域提供了一种全面的解决方案，代表了当今国际科学计算软件的先进水平。

MATLAB 虽然在科学运算领域应用广泛，但是在图形用户界面处理方面功能不够强大，缺少独立的应用程序框架，所以开发的计算程序必须在安装 MATLAB 的计算机上才能够运行。而 Visual C++集成环境则是以 C/C++语言为编程语言，将可视化程序设计和算法融合，成为多领域程序开发利器。所以如果能够将 MATLAB 与 Visual C++进行有效结合，将发挥 Visual C++和 MATLAB 各自的科学优势，势必能更好地解决科学和工程领域的相关计算和可视化的相关问题。

笔者结合自己多年使用 Visual C++和 MATLAB 混合编程进行科学研究和项目开发的经验撰写了本书。希望引领各位读者能够通过混合编程的方式解决自己的领域问题。本书结合大量不同领域的实际案例，全面、系统地介绍了 Visual C++和 MATLAB 混合编程的基础知识，不同类型的混合编程开发方式以及在不同领域的应用，全书以大量实例贯穿于讲解过程中。学习完本书后，读者应该可以具备使用 Visual C++和 MATLAB 通过编程实现领域问题求解的能力。

本书读者对象包括：

- Visual C++程序开发人员；
- 利用 MATLAB 做产品开发的工程技术人员；
- MATLAB 技术爱好者；
- 大中专院校的工程类学生。

本书由丁毓峰编著，参加编写的还有潘迁、龚力、李建峰、李龙海、房伟、曹胜平、张云群、李伟等硕士研究生。编写过程中参考了多种同类教材、专著，在此向参考文献与资料的编、著者表示衷心感谢！由于时间仓促，作者的水平所限，本书难免有一些错误和不足之处，欢迎广大读者给予批评和指正。读者在阅读本书的过程中如遇到问题或者有相关建议，请发邮件到 dyfjd@126.com，我们将及时进行反馈。

编者

2012 年 4 月

目 录

第 1 章 混合编程环境的搭建	1	2.2.3 Visual C++异常处理	26
1.1 MATLAB 与 C/C++混合编程的优点	1	2.3 Windows 程序内部运行机制	28
1.1.1 MATLAB 编程的优缺点	1	2.3.1 API 与 SDK	28
1.1.2 C/C++编程的优缺点	1	2.3.2 窗口与句柄	29
1.1.3 混合编程的优缺点	2	2.3.3 消息与消息队列	29
1.2 混合编程主要方法概述	3	2.4 动态链接库基础	30
1.2.1 Visual C++调用 MATLAB 引擎	3	2.4.1 DLL 与进程的地址空间	30
1.2.2 基于数据文件交换的方法	3	2.4.2 DLL 分类	31
1.2.3 基于 COM 技术的方法	4	2.4.3 创建 DLL 模块	32
1.2.4 使用 MATCOM 方法	4	2.5 Visual C++程序编译链接的原理与过程	34
1.2.5 基于 ActiveX 控件的方法	4	2.5.1 程序设计编译原理	34
1.2.6 使用 MATLAB Add-in 方法	5	2.5.2 Visual C++程序编译链接过程	34
1.3 Visual C++和 MATLAB 混合编程环境要求	5	2.6 MFC 框架程序	36
1.4 Visual C++的安装和配置	5	2.6.1 MFC AppWizard	36
1.4.1 Visual C++ 6.0 的安装	5	2.6.2 基于 MFC 的程序框架剖析	37
1.4.2 Visual C++的配置	8	2.7 ActiveX 控件	42
1.5 MATLAB 的安装和配置	11	2.7.1 概述	42
1.5.1 MATLAB 的安装	11	2.7.2 ActiveX 控件分类	43
1.5.2 MATLAB 的配置	13	2.7.3 ActiveX 控件应用	43
1.6 安装和配置的常见问题	15	2.8 Visual C++程序的调试和优化	43
1.6.1 Visual C++的安装和配置问题	15	2.8.1 Visual C++程序调试方法和过程	43
1.6.2 MATLAB 的安装和配置问题	16	2.8.2 Visual C++程序优化	46
1.7 小结	18	2.9 小结	48
第 2 章 Visual C++开发基础	19	第 3 章 MATLAB 编程基础	49
2.1 C++面向对象程序设计	19	3.1 MATLAB 程序流程控制	49
2.1.1 面向对象语言和方法	19	3.1.1 顺序结构	49
2.1.2 类、对象和消息	20	3.1.2 循环结构	50
2.2 C++异常处理机制	21	3.1.3 选择结构	51
2.2.1 错误和异常	22	3.1.4 分支语句	52
2.2.2 异常处理的机制和实现	22	3.1.5 其他控制语句	52
		3.2 函数句柄	53
		3.2.1 创建和查看函数句柄	53
		3.2.2 使用函数句柄	54

3.3	变量的检测和限权使用函数	54	4.4	应用实例	82
3.3.1	输入/输出变量检测指令	54	4.5	小结	88
3.3.2	跨空间变量传递	55			
3.3.3	子函数和私有函数	56	第 5 章 Visual C++调用 MATLAB		
3.4	串(表达式)演算函数	56	的 C/C++数学函数库	89	
3.4.1	eval	56	5.1	MATLAB C++数学库概述	89
3.4.2	feval	57	5.2	在 Visual C++环境下调用	
3.5	MATLAB 面向对象编程	57	MATLAB C++数学库	90	
3.5.1	MATLAB 中的类	57	5.2.1	设置静态链接库	90
3.5.2	具有类属性的数据	58	5.2.2	设置 C++选项卡中的选项	91
3.5.3	实现带类方法的操作	59	5.2.3	设置头文件	92
3.6	MATLAB 的数据类型	60	5.3	mwArray 阵列及系统函数	
3.6.1	变量与常量	60	的调用	93	
3.6.2	数字变量的运算及显示格式	62	5.3.1	操作矩阵	93
3.6.3	字符串	63	5.3.2	操作 MATLAB mwArray	
3.7	M 脚本文件和 M 函数文件	63	阵列概述	94	
3.7.1	M 文件的一般结构	64	5.3.3	创建 MATLAB mwArray	
3.7.2	M 脚本文件	64	阵列的操作	94	
3.7.3	M 函数文件	65	5.3.4	数据阵列的操作	95
3.7.4	局部变量和全局变量	66	5.3.5	稀疏矩阵阵列的操作	96
3.8	MATLAB 程序的调试和优化	66	5.3.6	字符型阵列的操作	97
3.8.1	MATLAB 程序调试方法		5.3.7	单元阵列的操作	99
和过程	67	5.3.8	结构体阵列的操作	100	
3.8.2	MATLAB 程序优化	72	5.3.9	调用系统函数	101
3.9	小结	73	5.4	应用实例	102
			5.5	小结	107
第 4 章 Visual C++调用			第 6 章 基于数据文件交换的混合		
MATLAB Engine 库	74		编程方法	108	
4.1	MATLAB Engine 概述	74	6.1	MAT 文件概述	108
4.2	Visual C++使用		6.1.1	MAT 文件格式	108
MATLABEngine 库	75		6.1.2	读取 MAT 文件 load	110
4.2.1	设置 Visual C++编译环境	75	6.1.3	写 MAT 文件	111
4.2.2	启动/关闭引擎	76	6.2	Visual C++操作 MAT 时	
4.2.3	向 MATLAB 发送命令	77	的环境设置	112	
4.2.4	显示或隐藏 MATLAB 窗口	79	6.3	用 C/C++语言操作 MAT	
4.3	MATLAB 数据类型 mxArray	80	文件的 API 函数	114	
4.3.1	创建 mxArray 类型数据	80	6.3.1	读写 MAT 文件的 API	
4.3.2	删除 mxArray 类型数据	81	函数简介	114	
4.3.3	获取 mxArray 数据大小	81	6.3.2	打开 MAT 文件	115
4.3.4	判断 mxArray 数组类型	81	6.3.3	关闭 MAT 文件	115
4.3.5	操作 mxArray 数组数据	82			

6.3.4	获得 MAT 文件中所有 阵列的目录	116	7.4	在 Visual C++ 中使用 MATLAB 的 COM 组件	142
6.3.5	获得 MAT 文件的 C 语言 文件句柄	116	7.4.1	以早期绑定方式调用 COM 组件	143
6.3.6	从 MAT 文件中获取一个 阵列变量	116	7.4.2	#import 指令的使用	143
6.3.7	将阵列变量内容写入 MAT 文件	117	7.5	应用实例	144
6.3.8	获得 MAT 文件中下一个 阵列的数据	117	7.5.1	创建 MATLAB 组件	144
6.3.9	从 MAT 文件中删除一个 阵列	118	7.5.2	创建 Visual C++ 工程	145
6.3.10	将阵列内容写入到 MAT 文件中	118	7.5.3	完善代码	147
6.3.11	从 MAT 文件中读取 MATLAB 阵列头信息	118	7.6	小结	151
6.3.12	从 MAT 文件中读取下一个 MATLAB 阵列头信息	118	第 8 章	使用 MATCOM 工具的 混合编程	152
6.4	应用实例	119	8.1	安装 MATCOM	152
6.4.1	环境设置	119	8.2	MATCOM 的基础及应用	154
6.4.2	在 MATLAB 中定义 两个变量	120	8.2.1	使用 MATCOM C++ 矩阵库的矩阵类 Mm	154
6.4.3	建立 Visual C++ 工程	120	8.2.2	在 Visual C++ 中使用 MATCOM C++ 矩阵库	157
6.5	小结	126	8.2.3	MATCOM C++ 矩阵库 的图形和图像显示	161
第 7 章	基于 COM 技术的方法 实现混合编程	127	8.2.4	MATCOM 用于图形显示 的函数	163
7.1	COM 技术概述	127	8.2.5	MATCOM 用于图像显示 的函数	164
7.1.1	COM 结构	127	8.3	MIDEVA 概述	164
7.1.2	COM 组件的有关概念	129	8.4	Visual C++ 使用 MIDEVA 的环境设置	165
7.1.3	COM 特性	131	8.4.1	添加头文件和添加库文件	165
7.1.4	COM 发展前景	132	8.4.2	添加 MIDEVA 提供的插件	166
7.2	COM 技术接口	132	8.5	应用实例	166
7.2.1	从 API 到 COM 接口	132	8.6	小结	169
7.2.2	接口定义和标示	133	第 9 章	使用 ActiveX 技术的 混合编程	170
7.2.3	用 C++ 语言定义接口	135	9.1	ActiveX 技术基础	170
7.2.4	接口描述语言 IDL	135	9.1.1	ActiveX 的定义	170
7.3	使用 MATLAB COM 编译器生成 COM 组件	136	9.1.2	ActiveX 的内容	171
7.3.1	MATLAB COM 编译器用法	136	9.1.3	MATLAB 支持的 ActiveX 技术	171
7.3.2	MATLAB COM 编译器 产生的 COM 组件	141	9.2	利用 ActiveX 自动控制器 实现混合编程	172

9.2.1	ActiveX 自动控制器	172	10.5	编程方式不同的对比	207
9.2.2	ActiveX 相关函数	173	10.6	小结	207
9.2.3	ActiveX 对象的创建、事件 处理与对象释放	174	第 11 章 图形图像显示 208		
9.2.4	查询和设置 ActiveX 对象 的属性	176	11.1	Visual C++和 MATLAB 图形图像处理混合编程原则	208
9.2.5	查询及调用 ActiveX 组件 的方法、事件	177	11.2	Visual C++的图形图像处理 及接口设计	208
9.3	ActiveX 自动化服务器	177	11.2.1	Visual C++的图像 处理方法	208
9.3.1	在客户程序中执行 MATLAB 命令	178	11.2.2	Visual C++图形处理方法	212
9.3.2	与客户程序进行数据交换	178	11.2.3	Visual C++和 MATLAB 图形 图像处理的接口设计	212
9.4	应用实例	179	11.3	MATLAB 图像处理基础	213
9.4.1	利用 ActiveX 自动控制器 实现混合编程	179	11.3.1	显示图像	213
9.4.2	利用 ActiveX 的自动化 服务器进行混合编程	181	11.3.2	查看内存中的图像	214
9.4.3	利用 MATLAB ActiveX 引擎进行混合编程	183	11.3.3	图像灰度分布直方图 均衡化	214
9.5	小结	185	11.3.4	图像文件的保存	215
第 10 章 科学运算 186			11.3.5	查看新生成文件的内容	215
10.1	科学运算概述	186	11.4	图像格式与 MATLAB 图像类型	216
10.2	混合编程在科学运算的 开发原则	186	11.4.1	常用图像格式	216
10.2.1	Visual C++处理科学 运算问题的优缺点	187	11.4.2	MATLAB 图像类型	218
10.2.2	MATLAB 处理科学 运算的优缺点	187	11.4.3	MATLAB 图像类型转换	219
10.2.3	混合编程在科学 运算的开发原则	187	11.5	MATLAB 图像显示命令	222
10.3	Visual C++和 MATLAB 在科学运算中的衔接方式	188	11.5.1	MATLAB 图像的读/写 和显示	222
10.3.1	MATLAB 与 Visual C++ 混合编程实现方法	188	11.5.2	二进制图像的显示方法	223
10.3.2	MATCOM 编译器	188	11.5.3	灰度图像的显示方法	223
10.4	线性方程组求解	189	11.5.4	索引图像的显示方法	224
10.4.1	MATLAB 求解	190	11.5.5	RGB 图像的显示方法	224
10.4.2	Visual C++求解	193	11.5.6	磁盘图像的直接显示	224
10.4.3	Visual C++和 MATLAB 混合 编程对线性方程组求解	204	11.6	MATLAB 图形显示命令	225
			11.7	Visual C++和 MATLAB 图形图像处理应用实例	225
			11.7.1	Visual C++中调用 MATLAB 函数画图	225
			11.7.2	利用 MATCOM 绘制 动态曲线	228
			11.7.3	二维和三维曲线绘制 综合应用	231

11.8 小结	244	13.3.4 系统不同模型之间的转换	279
第 12 章 图像识别	245	13.4 控制系统的分析	283
12.1 图像识别的混合编程规则 和接口设计	245	13.4.1 系统的时域分析	283
12.1.1 Visual C++和 MATLAB 图像识别的混合编程原则	245	13.4.2 系统的根轨迹分析	285
12.1.2 Visual C++和 MATLAB 图像识别的接口设计	245	13.4.3 系统的频域分析	287
12.2 图像识别概述	246	13.4.4 系统的稳定性分析	290
12.2.1 图像识别的发展阶段	246	13.5 基于 Visual C++和 MATLAB 的控制系统的模型输入和分析	291
12.2.2 图像识别的基础	247	13.5.1 Visual C++界面设计	292
12.2.3 图像识别原理	247	13.5.2 Visual C++调用 MATLAB 引擎实现的关键技术	292
12.3 主要图像识别方法	248	13.5.3 运行结果	296
12.3.1 统计模式的识别方法	248	13.6 小结	297
12.3.2 结构语句的识别方法	248	第 14 章 控制系统的设计仿真	298
12.3.3 模糊集识别法	249	14.1 控制系统校正器原理	298
12.3.4 神经网络识别法	249	14.1.1 串联校正	298
12.4 图像识别的应用	249	14.1.2 反馈校正	300
12.4.1 图像预处理	250	14.1.3 复合校正	301
12.4.2 图像分割	253	14.2 控制系统校正器设计和仿真	302
12.4.3 图像识别	254	14.2.1 超前校正设计	302
12.5 图像识别应用综合实例	260	14.2.2 滞后校正设计	305
12.5.1 新建图像识别 M 文件	260	14.2.3 滞后-超前校正设计	307
12.5.2 在 MATLAB 下创建 COM 组件	264	14.2.4 PID 校正器	308
12.5.3 Visual C++单击工程中 调用 COM 组件	267	14.3 基于 Visual C++和 MATLAB 的控制系统校正器设计	311
12.5 小结	269	14.3.1 控制系统校正器的 实现方式	311
第 13 章 控制系统模型输入和分析	270	14.3.2 控制系统分析界面设计	312
13.1 MATLAB 控制系统工具箱	270	14.3.3 代码实现	313
13.1.1 Simulink	270	14.3.4 运行结果	315
13.1.2 其他解决控制领域问题 的工具箱	271	14.4 小结	317
13.2 控制系统模型输入和分析 的混合编程原则及接口设计	271	第 15 章 信号频谱分析	318
13.3 控制系统模型的输入	273	15.1 信号频谱分析的混合编程 规则和接口设计	318
13.3.1 传递函数模型的输入	273	15.1.1 Visual C++和 MATLAB 信号 频谱分析的混合编程原则	318
13.3.2 零极点增益模型的输入	276	15.1.2 Visual C++和 MATLAB 信号 频谱分析的接口设计	318
13.3.3 状态空间模型的输入	278	15.2 信号频谱分析的基本概念	319

15.2.1	周期信号与离散光谱的定义	319	16.6	混合编程实例	359
15.2.2	信号频谱分析和显示原理	320	16.6.1	使用 Visual C++和 MATLAB 混合编程的方式绘制 sinc 函数波形	359
15.3	信号 FFT 频谱分析	321	16.6.2	数据传输和小波分析示例程序	362
15.3.1	数据采样	321	16.7	小结	365
15.3.2	采样数据导入 MATLAB	322			
15.3.3	对采样数据进行频谱分析	323			
15.4	MATLAB 频谱分析函数	324	第 17 章 Visual C++和 MATLAB		
15.5	Visual C++开发信号 FFT 频谱分析与显示结果	326		开发齿轮优化设计系统	366
15.5.1	新建 M 文件	326	17.1	优化设计理论	366
15.5.2	创建 COM 组件	327	17.2	MATLAB 优化设计工具箱	367
15.5.3	Visual C++单击工程中调用 COM 组件	330	17.2.1	优化工具箱简介	367
15.5.4	信号 FFT 实践及频谱分析	332	17.2.2	优化工具箱 4.x 的新特色	368
15.6	小结	340	17.2.3	优化函数	368
			17.2.4	无约束优化问题	370
			17.2.5	约束优化问题	373
第 16 章 数据采集和分析		341	17.3	Visual C++与绘图软件 SolidWorks 的接口	376
16.1	信号和信号处理	341	17.3.1	SolidWorks 的 API 接口概述	376
16.2	数据采集和分析处理的混合编程原则	341	17.3.2	对象分类	376
16.3	Visual C++的数据采集和分析及接口设计	343	17.3.3	几个重要的 SolidWorks API 对象	378
16.3.1	监控程序中的数据实时采集	343	17.3.4	使用 Visual C++对 SolidWorks 的二次开发	379
16.3.2	基于 MSCComm 控件的串口通信	344	17.4	齿轮优化设计系统概述	379
16.4	信号的采集原理	346	17.4.1	齿轮优化设计系统需求	379
16.4.1	采样定理	346	17.4.2	齿轮优化设计系统设计	380
16.4.2	信号的产生	347	17.5	齿轮优化设计系统开发	380
16.4.3	连续时间信号在 MATLAB 中的运算	353	17.5.1	齿轮优化设计的 MATLAB 实现	380
16.4.4	连续时间 LTI 系统的时域分析	354	17.5.2	SolidWorks 环境内的 Visual C++和 MATLAB 混合编程	385
16.5	小波理论在信号分析中的应用	356	17.5.3	齿轮优化设计系统运行	397
16.5.1	小波分析原理	356	17.6	小结	398
16.5.2	小波在语音信号增强中的应用	357			
16.5.3	小波在语音信号压缩中的应用	358	第 18 章 基于 Visual C++和 MATLAB 的汽车 ABS 系统仿真		399
			18.1	汽车 ABS 系统工作原理	399

18.2	MATLAB 中建立仿真模型	399	18.3	建立汽车 ABS 仿真模型	401
18.2.1	Simulink 建立模型	399	18.3.1	汽车 ABS 系统数学模型	402
18.2.2	Simulink 模型运行及 分析	400	18.3.2	建立和运行汽车 ABS 系统仿真模型	403
18.2.3	仿真数据的输入输出 设置	400	18.4	混合仿真过程	406
18.2.4	解算器 (Solver) 的设置	400	18.5	小结	409
18.2.5	Simulink 模块的合成 与封装	401	参考文献		410

第 1 章 混合编程环境的搭建

在实际程序开发中，为了提高开发效率并使开发的程序功能强大，性能优异，通常将 MATLAB 与 C/C++ 进行混合编程。混合编程可以发挥它们各自的优势，更好地解决实际问题。本章将介绍 MATLAB 与 C/C++ 混合编程的优缺点、MATLAB 与 C/C++ 混合编程的主要方法、Visual C++ 6.0 和 MATLAB 的安装过程，以及安装和配置的常见问题。

1.1 MATLAB 与 C/C++ 混合编程的优点

MATLAB 目前已经成为科学运算领域的标准编程语言和开发工具。但是，在图形用户界面处理方面功能不够强大。而 C/C++ 由于其适用性广、效率高，在很多领域得到了广泛应用。尤其是 Visual C++ 集成环境将可视化程序设计和算法相融合，成为多领域程序开发利器。所以，如果能够将 MATLAB 与 Visual C++ 进行有效结合，发挥各自的优势，则将达到事半功倍的效果。

1.1.1 MATLAB 编程的优缺点

MATLAB 作为第 4 代计算机语言，利用其丰富的函数资源，可以使编程人员从烦琐的程序代码中解放出来，将精力集中在领域问题上而不是编程上。MATLAB 语言相对于传统的编程语言有很多优点，主要包括以下几点。

- 易用性：MATLAB 提供的集成环境、工作台和扩展范例等编程工具便于用户快速入门。
- 可移植性：MATLAB 可以在 Windows 和众多版本的 UNIX 系统上安装运行。
- 用户图形界面向导：利用 MATLAB 提供的用户图形界面向导，可以轻松地建立一个交互式的用户图形界面，MATLAB 还具有较强的编辑图形界面的能力。
- 专业领域工具箱：MATLAB 中提供了多种专业工具箱，可以帮助用户解决领域内复杂的问题。

MATLAB 虽然有很多优点，但是也存在一些缺点，主要包括以下几点。

- 由于 MATLAB 是解释型语言，而不是编译型语言，产生的程序执行速度慢。
- 若将其可执行文件编译后，则需要附带一个库函数包，编译独立运行的程序比较麻烦，而且要做较多的配置。

1.1.2 C/C++ 编程的优缺点

C 语言作为一种简单、高效的编程语言，易于编写简单的程序。C 语言使用面向过程

的编程模式，从而减少对计算机的依赖。作为一种面向过程的编程语言，由于其十分接近汇编语言。所以，可以说 C 语言既是高级语言也是低级语言。

C 语言的优点包括以下几点。

- 面向机器，让用户可以完全地操纵计算机，效率较高。
- 面向过程开发，以函数为中心。通过函数的参数传递实现了接口连接。
- 运行效率高，可以广泛应用于各种科学和工程计算领域。

C 语言也存在一些缺点，主要包括以下几点。

- 由于过多地强调效率，使用 C 语言编程时，更多地需要考虑计算机的处理能力。所以抽象能力不足，容易出现小错误。对于编写大型的程序时调试比较麻烦。
- 函数无法有效控制相关数据，不能保证外部状态不变，容易出现程序错误。
- 数据常常作为全局的变量保存起来。所以，很难保证其他代码不破坏函数依赖的数据状态。

C++语言是作为 C 语言的扩展语言出现的，开始叫做“带类的 C”。后来，C++逐渐演化成一门独立的语言。但还是和 C 语言兼容，并且可以组织在一起解决问题。C++语言虽然以支持面向对象的开发而闻名，但是 C++支持多范型的开发方式，包括面向过程、基于对象、面向对象、模版等，并且 C++和 C 语言是完全兼容的。因此，可以使用 C++编译系统编写 C 语言的程序。

C++语言的主要优点包括以下几点。

- 同时支持面向对象程序设计和面向过程设计。
- 通过继承、重载和多态性等特征实现了软件重用和自动生成程序，提高了编程效率，完全兼容 C 语言，多数 C 语言编写的库函数和应用程序都可为 C++所用。
- 在继承了 C 语言优点的基础上，扩充了新的数据类型，使 C++的编译系统可检查出更多类型错误；加强了一致性（Consistency）检查机制，提高了软件开发的效率和质量。

C++语言虽然具有较多优点，但是也存在一些缺点，主要包括以下几点。

- 编写的程序比 C 语言的程序占用更多的内存。
- 编写的程序可移植性不强。
- 编程过程中使用的动态对象必须及时销毁，否则会造成内存泄漏，可能会导致程序崩溃。

1.1.3 混合编程的优缺点

MATLAB 是当今应用最为广泛的数学软件，它具有相当强大的数值计算、数据处理、系统分析、图形显示以及符号运算、各专业领域工具箱等功能。在这个平台上，简单地编程就可以完成十分复杂的功能，大大提高了工程分析计算的效率。

在科学计算和仿真方面，虽然 MATLAB 具有强大的功能，但是如果只是使用 MATLAB 进行问题求解，那么编写的程序只在该平台上才能使用。也就是说，必须在安装了 MATLAB 系统的计算机上才能编译和运行 .m 文件，这样就给用户进行科学和工程计算带来了很大的

不便。另外，由于 MATLAB 使用解释方式执行代码，导致代码的执行速度很慢。

和 MATLAB 的编程语言相比，C/C++ 语言和相关开发工具如 Visual C++、C++Builder 等，在图形界面开发、进程控制、程序打包分发、执行速度等方面具有明显优势。

由此，开发者萌生了混合编程的思想。采用混合编程的方式，可以集成两者的优点进行程序开发。使用混合编程的方法，可以充分发挥 MATLAB 在矩阵（数组）上的强大运算能力，同时可以利用涵盖了各个领域的功能强大的算法函数库，不再需要浪费时间，使用 C++ 单独开发相关算法函数。通过开发 MATLAB 与 C/C++ 的接口，可以把 MATLAB 的强大功能融入使用 C/C++/Visual C++ 开发的应用程序中，并且通过 Visual C++ 的高级语言编译器将代码编译为二进制代码，从而可以提高执行速度。

使用 MATLAB 和 C/C++ 语言进行混合编程，可以集成它们各自的优点进行程序开发。但是，同时也为程序开发人员混合提出了更多要求，带来了新的挑战。程序开发人员必须同时熟练掌握 MATLAB 和 C/C++ 语言的特点、编程方法和开发环境，这样才能够在规定时间内开发出高效率的程序。

1.2 混合编程主要方法概述

MATLAB 和 Visual C++ 混合编程有很多种。目前常用的几种方法包括 Visual C++ 调用 MATLAB 引擎、基于数据文件交换的方法、基于 COM 技术的方法、使用 MATCOM 方法、基于 ActiveX 控件的方法和使用 MATLAB Add-in 方法等混合编程方法。

1.2.1 Visual C++ 调用 MATLAB 引擎

MATLAB Engine（引擎）是指一组由 MATLAB 软件提供的接口函数。它支持 C/C++、Fortran 等编程语言。通过调用这些接口函数，用户可以在其他编程环境中实现对 MATLAB 的控制。

Visual C++ 调用 MATLAB 引擎，就是使用 Visual C++ 在应用程序中对 MATLAB 进行调用，将 MATLAB 作为一个操作引擎使用，使其在后台运行。MATLAB 引擎采用客户端/服务器（Client/Sever）的方式，是指把应用处理程序负载分布到客户机和服务器上的模式。

在这种模式下，MATLAB 作为后台服务器，而 Visual C++ 程序作为前台客户端。通过 Windows 的动态控件与服务器通信，向 MATLAB 引擎传递命令和数据信息，从 MATLAB 引擎接受数据信息。用户可以在 Visual C++ 应用程序中调用这些接口函数，实现对 MATLAB 引擎的控制。采用该方法几乎能利用 MATLAB 提供的全部功能，但是前提条件是必须在计算机上安装了 MATLAB 软件，而且执行效率低，主要用于个人使用或做演示，在实际大型应用程序开发中使用得不多。

1.2.2 基于数据文件交换的方法

MATLAB 中的数据文件 MAT 是一种数据文件，用来保存 MATLAB 工作空间中的变

量。使用它可以解决不同编程语言之间传递数据的问题，这也是混合编程的主要方法之一。MAT 文件是 MATLAB 软件使用的一种特殊文件格式的数据文件，该数据文件采用二进制格式进行描述。开发人员可以将一个或者多个特定的 MATLAB 变量定义并保存在 MAT 文件中，供需要时候使用。MATLAB 通常采用 MAT 文件把工作空间的变量存储在磁盘里。

当需要重新载入先前保存的变量到工作空间时，只需要重新将对应的 MAT 加载即可。在 MATLAB 中载入对应的 MAT 文件后，在当前 MATLAB 工作空间，可以完全再现当初保存该 MAT 文件时的那些变量，这是其他文件格式无法做到的，体现了 MAT 文件的优势。

1.2.3 基于 COM 技术的方法

基于 COM 技术的混合编程方法是 MathWorks 公司推荐的使用方法，该方法主要通过使用 MATLAB 提供的 Combuilder 工具实现。COM (Component Object Module) 是一种通用的对象接口，编程语言只要按照接口标准实现功能，就可以使用其他开发语言调用它。MATLAB 推出的 Combuilder 可以将 MATLAB 编写的程序制作成 COM 组件，供其他编程语言调用。该方法实现简单，通用性强，而且几乎可以使用 MATLAB 的任何函数，但是不支持脚本文件。因此，在应用程序规模较大、调用工具箱函数或调用函数较多时使用。

1.2.4 使用 MATCOM 方法

MATCOM 是第三方控件，原本属于 MathTool 公司，后来由 MathWorks 公司收购。使用该工具可以将 MATLAB 的 m 脚本文件和 m 函数转化成为功能相同的 C/C++ 文件。通常在 m 文件不大或者不需要进行三维绘图的情况下使用。相比使用其他混合编程方法具有以下优点。

- 转换过程简单 (由 MATCOM 工具自动实现)，容易实现。
- 可以转换独立的脚本文件，也可以转换嵌套脚本文件。
- 可以方便地生成动态链接库文件和可执行文件。
- 设置环境后，可以使用 MATLAB 的工具箱函数。

同时，使用 MATCOM 进行混合编程也存在以下一些缺点。

- 对 C/C++ 语言中的 struct 等类的支持有缺陷。
- 程序中涉及的复杂函数调用时嵌套的文件数量会较多。
- 部分绘图语句无法实现或得不到准确图形图像，尤其是对于三维图像。

1.2.5 基于 ActiveX 控件的方法

ActiveX 组件是指符合特定规范的一些可执行的代码或一个程序。例如，可执行文件 (.EXE)、动态链接库文件 (.DLL) 或 OCX 控件 (.OCX) 等都可以作为 ActiveX 组件存在。其最大的优势就是可重用性好。通过使用 ActiveX 技术，开发人员就能够将可复用的软件组装到应用程序或者服务程序中，然后嵌入到程序或者网页中，随网页传送到客户的浏览器上，并在客户端执行。利用 MATLAB 软件提供的 ActiveX 技术，可以在 Visual C++ 开发

的应用程序中完全控制 MATLAB。该方案利用一个专门的类对 ActiveX 的接口代码进行封装，使用户可以专注于 MATLAB 的功能应用。

通过 ActiveX 组件，Visual C++ 可以与 MATLAB 实现无缝集成，用户可以轻松地开发出高效、功能强大的应用程序。

1.2.6 使用 MATLAB Add-in 方法

对于 C/C++ 和 MATLAB 混合编程而言，该方法是最方便的方法之一。MATLAB 6.0 以后的版本对其编译器 (Compiler) 做了较大的改进，可以支持更多的数据类型，提供更强的优化功能，更重要的是提供了一个 MATLAB Add-in，可以用它实现一个 MATLAB 和 Visual C++ 直接集成的途径。

该方法具有如下特征：快速集成 M 文件到 C++ 工程中，创建独立的 C/C++ 应用程序或 C MEX DLL。通过 M 文件创建共享库或 MEX 文件；直接修改 M 源文件而不是修改生成的 C/C++ 文件；内含 Visual Matrix Viewer，调试过程中可以查看矩阵变量的值；可以方便、快捷地打包应用程序等。

不过，需要注意的是，MATLAB 7.0 之后就不再支持这种方法了，如果使用该方法，可以考虑使用 MATLAB 6.5 结合 Visual C++ 进行开发。

1.3 Visual C++ 和 MATLAB 混合编程环境要求

由于目前 MATLAB 应用广泛的版本是运行在 Windows 系统平台上。Visual C++ 也是 Windows 操作系统上的主要开发工具之一，所以本书讨论的 Visual C++ 和 MATLAB 混合编程是指在 Windows 操作系统上的混合编程。

另外，由于 Windows 操作系统由多个版本，包括 Windows NT、Windows 2000、Windows 2003、Windows XP、Windows 7 和 Windows Vista 等。在某个操作系统下开发的应用程序有可能不能在其他操作系统上运行。为了能够正常运行，需要在新的操作系统上进行调试，调试通过才能保证程序的正常稳定运行。考虑到 Windows XP 是目前使用最为广泛的操作系统，所以使用 Windows XP 操作系统作为系统平台。另外，使用 Visual C++ 和 MATLAB 2009/2010 版本作为开发工具。

1.4 Visual C++ 的安装和配置

为了使用 Visual C++ 进行程序开发，首先必须正确安装 Visual C++，并进行正确的配置。本节主要介绍 Visual C++ 的安装过程和需要正确配置的问题。


1.4.1 Visual C++ 6.0 的安装

Visual C++ 6.0 是 Visual C++ 开发工具中最经典的版本。目前，很多涉及 Visual C++ 开

发的教程使用该版本。从 Visual C++ 6.0 以后，微软没有再出单独版本的 Visual C++ 版本，而是直接集成在 Visual Studio 中，主要版本包括 Visual C++ 2005、Visual C++ 2008 和 Visual C++ 2010 等。

随着版本的不断升级，标准也有适当的修改，功能越来越强大。但是，对计算机的内存和硬盘要求很高。例如，Visual Studio 2008 安装后需要 4GB 以上的空间，内存 2GB 以上，并且运行速度很慢，其中安装包中的很多功能开发人员几乎没有机会用到。为了使读者更方便进行安装、调试、学习，本书使用 Visual C++ 6.0 作为开发环境。

运行 Visual Studio 6.0 软件中的 setup.exe 程序，选择安装 Visual C++ 6.0，然后按照安装程序的指导一步一步完成安装过程。

(1) 使用鼠标双击  SETUP 安装文件，启动安装程序，会出现如图 1.1 所示的对话框，单击“下一步 (N)”按钮，弹出“最终用户许可协议”界面，如图 1.2 所示。

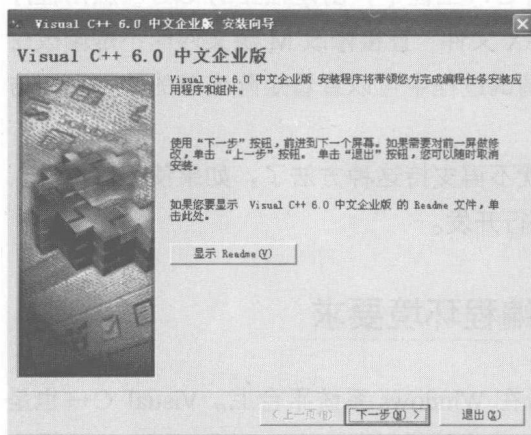


图 1.1 安装向导

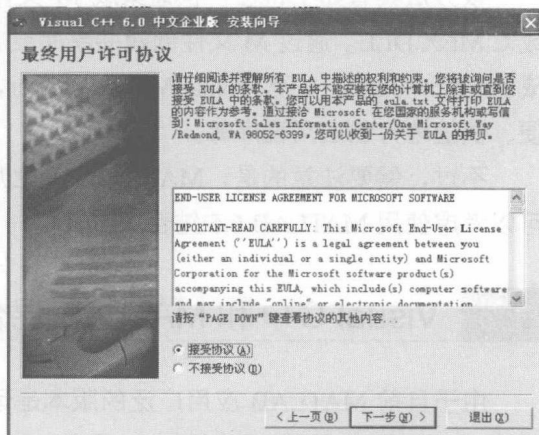


图 1.2 “最终用户许可协议”界面

(2) 选择“接受协议”单选项，单击“下一步 (N)”按钮，进入“产品号和用户 ID”界面，如图 1.3 所示。

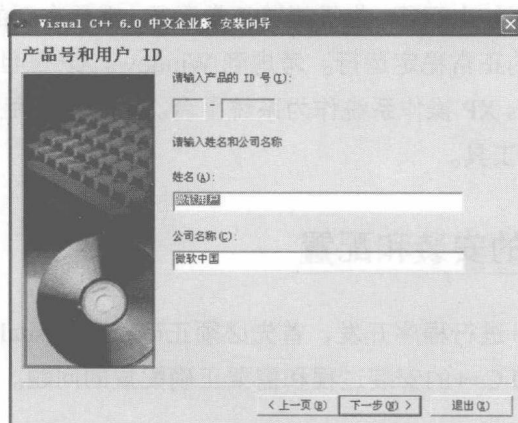


图 1.3 “产品号和用户 ID”界面

(3) 输入产品的 ID 号、姓名和公司名称等信息，单击“下一步 (N)”按钮，选择服