

# MATLAB微积分运算

薛定宇 著

Dingyü Xue

# 前言

## PREFACE

科学运算问题是每个理工科学生和科技工作者在课程学习、科学研究与工程实践中常常会遇到的问题，不容回避。对于非纯数学专业的学生和研究者而言，从底层全面学习相关数学问题的求解方法并非一件简单的事情，也不易得出复杂问题的解。所以，利用当前最先进的计算机工具，高效、准确、创造性地求解科学运算问题是一种行之有效的方法，尤其能够满足理工科人士的需求。

作者曾试图在同一部著作中叙述各个数学分支典型问题的直接求解方法，通过清华大学出版社出版了《高等应用数学问题的 MATLAB 求解》。该书从 2004 年出版之后多次重印再版，并于 2018 年出版了第 4 版，还配套发布了全新的 MOOC 课程<sup>①</sup>，一直受到广泛的关注与欢迎。首次 MOOC 开课的选课人数接近 14000 人，教材内容也被数万篇期刊文章和学位论文引用。

从作者首次使用 MATLAB 语言算起，已经有 30 年的时间了，通过相关领域的研究、思考与一线教学实践，积累了大量的实践经验与资料。这些不可能在一部著作中全部介绍，所以与清华大学出版社策划与编写了这套“薛定宇教授大讲堂”系列著作，系统、深入地介绍基于 MATLAB 语言与工具的科学运算问题的求解方法。

本系列著作不是原来版本的简单改版，而是作者通过十余年的经验和资料积累，全面贯穿“再认识”的思想写作而成的，深度融合科学运算数学知识与基于 MATLAB 的直接求解方法与技巧，力图更好地诠释计算机工具在每个数学分支的作用，帮助读者以不同的思维与视角了解工程数学问题的求解方法，创造性地得出问题的解。

本系列著作的卷 I 可以作为学习 MATLAB 入门知识的教材与参考书，也为读者深入学习与熟练掌握 MATLAB 语言编程技巧，深度理解科学运算领域 MATLAB 的应用奠定一个坚实的基础。后续每一卷试图对应一个数学专题或一门数学课程进行展开。整套系列著作的写作贯穿“计算思维”的思想，深度探讨该数学专

---

<sup>①</sup> MOOC 网址:<https://www.icourse163.org/learn/NEU-1002660001>

题的问题求解方法。本系列著作既适合学生在学完相应的数学课程之后,深入学习利用计算机工具的科学运算问题求解方法与技巧,也可作为相应数学课程同步学习的伴侣,在学习相应课程理论知识的同时,侧重学习基于计算机的数学问题求解方法,从另一个角度观察、审视数学课程所学的内容,扩大知识面,更好地学习、理解并实践相应的数学课程。

本书是系列著作的卷II,试图以一个全新的角度,按照一般微积分学教程的顺序全面介绍微积分问题的求解,侧重于利用 MATLAB 环境直接求解微积分的问题。首先介绍函数与序列的描述与图形绘制,然后介绍极限问题的求解、导数与微分问题的求解以及积分问题的求解,并介绍函数的逼近与级数求和等方面的内容,还介绍数值导数与数值积分方面的内容,并给出积分变换、分数阶微积分等的入门介绍。

值此系列著作付梓之际,衷心感谢相濡以沫的妻子杨军教授,她数十年如一日的无私关怀是我坚持研究、教学与写作工作的巨大动力。

薛定宇  
2019年5月

# 目 录

## CONTENTS

|       |                                      |    |
|-------|--------------------------------------|----|
| 第1章   | 微积分问题简介                              | 1  |
| 1.1   | 微积分学发展简史                             | 1  |
| 1.2   | 本书的主要内容                              | 4  |
| 第2章   | 函数与序列                                | 6  |
| 2.1   | 函数与映射                                | 6  |
| 2.1.1 | 函数的定义与描述                             | 6  |
| 2.1.2 | 常用超越函数的MATLAB计算                      | 7  |
| 2.1.3 | 一般函数的MATLAB表示                        | 7  |
| 2.1.4 | 函数的曲线与曲面表示                           | 8  |
| 2.2   | 不同函数的MATLAB表示                        | 9  |
| 2.2.1 | 反函数                                  | 9  |
| 2.2.2 | 复合函数                                 | 9  |
| 2.2.3 | 分段函数的描述                              | 10 |
| 2.2.4 | 隐函数                                  | 12 |
| 2.2.5 | 参数方程                                 | 13 |
| 2.2.6 | 极坐标函数                                | 16 |
| 2.3   | 奇函数与偶函数                              | 17 |
| 2.4   | 复变函数与映射                              | 18 |
| 2.4.1 | 复数矩阵及其变换                             | 18 |
| 2.4.2 | 复变函数的映射                              | 18 |
| 2.4.3 | Riemann曲面的绘制                         | 19 |
| 2.5   | 序列与函数项序列                             | 22 |
|       | 本章习题                                 | 23 |
| 第3章   | 函数与序列的极限                             | 26 |
| 3.1   | 单变量函数的极限                             | 27 |
| 3.1.1 | 单变量函数极限的 $\varepsilon$ - $\delta$ 定义 | 27 |

|       |                  |    |
|-------|------------------|----|
| 3.1.2 | 函数极限的计算机求解       | 29 |
| 3.1.3 | 复合函数的极限          | 31 |
| 3.1.4 | 序列的极限            | 31 |
| 3.1.5 | 分段函数的极限          | 32 |
| 3.1.6 | 无穷小量与无穷大量        | 33 |
| 3.2   | 单边极限与函数连续性       | 33 |
| 3.2.1 | 左极限与右极限          | 33 |
| 3.2.2 | 函数的连续性           | 35 |
| 3.2.3 | 区间极限运算           | 36 |
| 3.2.4 | 函数连续性的应用——方程解的判定 | 37 |
| 3.3   | 复函数的奇点、极点与留数     | 38 |
| 3.3.1 | 奇点与极点的计算         | 38 |
| 3.3.2 | 复变函数的留数          | 39 |
| 3.4   | 多元函数的极限          | 41 |
| 3.4.1 | 累极限              | 41 |
| 3.4.2 | 重极限及其计算          | 42 |
|       | 本章习题             | 44 |
| 第4章   | 函数的导数与微分         | 47 |
| 4.1   | 函数的导数和高阶导数       | 48 |
| 4.1.1 | 函数的导数与微分         | 48 |
| 4.1.2 | 函数导数与高阶导数        | 48 |
| 4.1.3 | 复合函数的导数          | 51 |
| 4.1.4 | 分段函数的导数          | 52 |
| 4.1.5 | 矩阵的导数            | 53 |
| 4.2   | 参数方程的导数          | 53 |
| 4.3   | 多元函数的偏导数         | 55 |
| 4.3.1 | 偏导数              | 55 |
| 4.3.2 | 全微分              | 58 |
| 4.3.3 | 多元复合函数的导数        | 58 |
| 4.4   | 场的梯度、散度与旋度       | 59 |
| 4.4.1 | 标量场与向量场          | 59 |
| 4.4.2 | 梯度、散度与旋度         | 59 |
| 4.4.3 | 向量场的势            | 61 |

|       |                     |    |
|-------|---------------------|----|
| 4.5   | 多元函数的导数矩阵           | 61 |
| 4.5.1 | Jacobi 矩阵           | 61 |
| 4.5.2 | Hesse 矩阵            | 62 |
| 4.5.3 | 标量函数的 Laplace 算子    | 63 |
| 4.6   | 隐函数的偏导数             | 63 |
| 4.6.1 | 单个隐函数的一阶导数          | 63 |
| 4.6.2 | 隐函数的高阶导数            | 64 |
| 4.6.3 | 隐函数方程组的偏导数计算        | 66 |
| 4.7   | 导数与微分的应用            | 68 |
| 4.7.1 | 极值问题                | 68 |
| 4.7.2 | Newton-Raphson 迭代方法 | 71 |
| 4.7.3 | 曲面的切面方程与法线方程        | 72 |
|       | 本章习题                | 73 |
| 第 5 章 | 函数的积分               | 75 |
| 5.1   | 单变量函数的不定积分          | 76 |
| 5.2   | 定积分与反常积分            | 79 |
| 5.2.1 | 定积分                 | 79 |
| 5.2.2 | 广义积分与反常积分           | 81 |
| 5.3   | 多重积分问题的 MATLAB 求解   | 83 |
| 5.3.1 | 多重不定积分              | 84 |
| 5.3.2 | 待定多项式的构造            | 85 |
| 5.3.3 | 多重定积分的计算            | 86 |
| 5.3.4 | 积分区域的处理与变换          | 87 |
| 5.4   | 定积分的应用              | 88 |
| 5.4.1 | 曲线弧长的计算             | 88 |
| 5.4.2 | 旋转体的体积计算            | 89 |
| 5.4.3 | 三维图形围成的体积与质量计算      | 90 |
| 5.4.4 | 概率密度与分布函数           | 91 |
| 5.4.5 | 积分变换入门              | 92 |
| 5.5   | 曲线积分                | 92 |
| 5.5.1 | 第一类曲线积分             | 93 |
| 5.5.2 | 第二类曲线积分             | 95 |
| 5.6   | 曲面积分                | 96 |
| 5.6.1 | 第一类曲面积分             | 96 |
| 5.6.2 | 第二类曲面积分             | 98 |

|                             |            |
|-----------------------------|------------|
| 本章习题                        | 100        |
| <b>第6章 级数展开与函数逼近</b>        | <b>103</b> |
| 6.1 级数求和                    | 103        |
| 6.1.1 数项级数的求和               | 104        |
| 6.1.2 无穷级数求和计算              | 106        |
| 6.1.3 函数项级数的求和              | 108        |
| 6.1.4 特殊的无穷项问题              | 109        |
| 6.2 无穷级数的收敛性判定              | 111        |
| 6.2.1 正项级数的一般描述             | 111        |
| 6.2.2 正项级数的收敛性判定            | 111        |
| 6.2.3 交替级数的收敛性判定            | 113        |
| 6.2.4 函数项级数的收敛区间            | 114        |
| 6.3 序列求积问题                  | 115        |
| 6.3.1 数项序列的乘积               | 115        |
| 6.3.2 函数项序列的乘积              | 116        |
| 6.3.3 正项序列求积的收敛性判定          | 116        |
| 6.4 Taylor 幂级数展开            | 117        |
| 6.4.1 单变量函数的 Taylor 幂级数展开   | 118        |
| 6.4.2 多元函数的 Taylor 幂级数展开    | 121        |
| 6.5 Fourier 级数展开            | 122        |
| 6.5.1 Fourier 级数的数学描述       | 122        |
| 6.5.2 Fourier 级数的 MATLAB 实现 | 123        |
| 6.6 单变量函数的有理函数近似            | 126        |
| 6.6.1 函数的连分式近似              | 126        |
| 6.6.2 函数的 Padé 近似           | 130        |
| 6.7 Laurent 级数展开            | 131        |
| 6.7.1 复变函数的 Laurent 级数展开    | 131        |
| 6.7.2 有理函数的 Laurent 级数      | 133        |
| 本章习题                        | 135        |
| <b>第7章 数值导数与微分</b>          | <b>139</b> |
| 7.1 数值导数算法                  | 139        |
| 7.1.1 前向差分与后向差分算法           | 140        |
| 7.1.2 $o(h^2)$ 精度中心差分算法     | 140        |
| 7.1.3 $o(h^4)$ 精度中心差分算法     | 141        |
| 7.1.4 更高精度的中心差分公式           | 141        |

|       |                    |     |
|-------|--------------------|-----|
| 7.1.5 | 一般高阶差分公式的推导与计算     | 142 |
| 7.1.6 | 高精度前向与后向差分算法       | 145 |
| 7.2   | 数值导数计算的 MATLAB 实现  | 146 |
| 7.2.1 | 二阶精度算法的实现          | 147 |
| 7.2.2 | 七点中心算法的实现          | 148 |
| 7.2.3 | 前向差分数值导数算法的实现      | 149 |
| 7.3   | 已知样本点的任意阶数值导数的求解函数 | 151 |
| 7.4   | 二元函数的偏导数计算         | 153 |
| 7.4.1 | 梯度计算               | 153 |
| 7.4.2 | 针对单变量的高精度偏导数算法     | 154 |
| 7.4.3 | 混合偏导数的数值计算         | 156 |
| 7.4.4 | 高阶混合偏导数的数值计算       | 157 |
| 7.5   | 样条插值与数值导数计算        | 158 |
| 7.5.1 | 三次样条               | 158 |
| 7.5.2 | B 样条               | 161 |
| 7.5.3 | 基于样条的数值导数计算        | 162 |
| 7.5.4 | 不等间距样本散点的数值偏导数计算   | 165 |
|       | 本章习题               | 167 |
| 第 8 章 | 数值积分               | 169 |
| 8.1   | 由给定样本点求数值积分        | 169 |
| 8.1.1 | 定积分的直接计算           | 169 |
| 8.1.2 | 积分函数的重建            | 171 |
| 8.1.3 | 等间距样本点的高精度数值积分方法   | 172 |
| 8.2   | 单变量数值积分问题求解        | 175 |
| 8.2.1 | 简单数值积分问题           | 175 |
| 8.2.2 | 数值积分问题的 MATLAB 求解  | 176 |
| 8.2.3 | 反常积分的数值计算          | 180 |
| 8.2.4 | 含参数函数的数值积分         | 181 |
| 8.2.5 | 积分函数的数值求解          | 183 |
| 8.3   | 双重积分问题的数值解         | 184 |
| 8.3.1 | 双重定积分的计算           | 184 |
| 8.3.2 | 双重积分曲面的计算          | 185 |
| 8.3.3 | 不同积分顺序的双重积分计算方法    | 185 |

|       |                          |     |
|-------|--------------------------|-----|
| 8.4   | 多重积分数值求解                 | 186 |
| 8.4.1 | 三重定积分的数值求解               | 187 |
| 8.4.2 | 含参数函数的三重积分               | 188 |
| 8.4.3 | 多重积分数值求解                 | 189 |
| 8.4.4 | 某些变边界多重积分问题的数值求解方法       | 191 |
| 8.5   | 数值积分的其他计算方法              | 191 |
| 8.5.1 | 基于 Monte Carlo 方法的数值积分近似 | 192 |
| 8.5.2 | 基于样条插值的数值积分              | 194 |
| 8.5.3 | 多重积分的数值计算                | 196 |
|       | 本章习题                     | 197 |
| 第9章   | 积分变换                     | 200 |
| 9.1   | Laplace 变换及其反变换          | 200 |
| 9.1.1 | Laplace 变换及反变换的定义与性质     | 201 |
| 9.1.2 | Laplace 变换的计算机求解         | 202 |
| 9.1.3 | 用 Laplace 变换求解微分方程       | 204 |
| 9.2   | Laplace 变换问题的数值求解        | 206 |
| 9.2.1 | 数值 Laplace 反变换           | 206 |
| 9.2.2 | 闭环系统响应的思想                | 207 |
| 9.2.3 | 数值 Laplace 变换            | 208 |
| 9.2.4 | 无理系统的响应计算                | 211 |
| 9.3   | Fourier 变换及其反变换          | 211 |
| 9.3.1 | Fourier 变换及反变换定义与性质      | 212 |
| 9.3.2 | Fourier 变换的计算机求解         | 212 |
| 9.3.3 | Fourier 正弦和余弦变换          | 213 |
| 9.3.4 | 离散 Fourier 正弦、余弦变换       | 215 |
| 9.3.5 | 快速 Fourier 变换            | 216 |
| 9.4   | 其他积分变换问题及求解              | 217 |
| 9.4.1 | Mellin 变换                | 217 |
| 9.4.2 | Hankel 变换及求解             | 219 |
| 9.5   | $z$ 变换及其反变换              | 220 |
| 9.5.1 | $z$ 变换及反变换定义与性质          | 221 |
| 9.5.2 | $z$ 变换的计算机求解             | 221 |
| 9.5.3 | 双边 $z$ 变换                | 223 |
| 9.5.4 | 有理函数 $z$ 反变换的数值求解        | 223 |
|       | 本章习题                     | 224 |

|        |                           |     |
|--------|---------------------------|-----|
| 第10章   | 分数阶微积分                    | 228 |
| 10.1   | 分数阶微积分的定义                 | 229 |
| 10.1.1 | 为什么要引入分数阶微积分的概念           | 229 |
| 10.1.2 | 分数阶微积分的定义                 | 230 |
| 10.2   | 不同分数阶微积分定义的关系与性质          | 231 |
| 10.3   | Grünwald–Letnikov 定义的数值实现 | 232 |
| 10.3.1 | Grünwald–Letnikov 定义      | 232 |
| 10.3.2 | 高精度算法与实现                  | 233 |
| 10.3.3 | 不同精度算法的定量比较               | 237 |
| 10.4   | Caputo 微积分定义的数值计算         | 239 |
| 10.5   | Oustaloup 滤波算法及其应用        | 241 |
| 10.5.1 | Oustaloup 滤波器近似           | 241 |
| 10.5.2 | Caputo 导数的滤波器近似           | 243 |
| 10.5.3 | 基于 Simulink 的 Caputo 导数计算 | 245 |
| 10.6   | 更高阶导数与积分的数值计算             | 246 |
|        | 本章习题                      | 248 |
|        | 参考文献                      | 249 |
|        | MATLAB 函数名索引              | 251 |
|        | 术语索引                      | 255 |



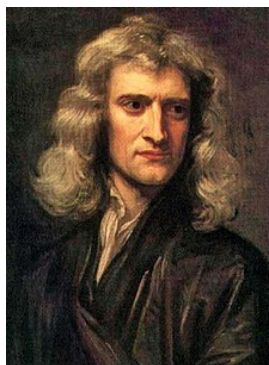
## 第1章



# 微积分问题简介

## 1.1 微积分学发展简史

人们普遍认为现代微积分学诞生于17世纪,是由英国科学家牛顿爵士(Sir Isaac Newton, 1643–1727, 图1-1(a))与德国数学家莱布尼茨(Gottfried Wilhelm Leibniz, 1646–1716, 图1-1(b))独立建立起来的。在微积分领域发展初期,牛顿与莱布尼茨使用的方法与研究对象是不同的,牛顿将时间 $t$ 看作函数的自变量,莱布尼茨认为函数的自变量为 $x$ ;牛顿将函数的导数记作 $\dot{x}, \ddot{x}$ ,莱布尼茨采用的是今天仍在使用的 $\mathrm{d}^n f(x)/\mathrm{d}x^n$ ;积分符号 $\int$ 也是莱布尼茨引入的。



(a) 牛顿



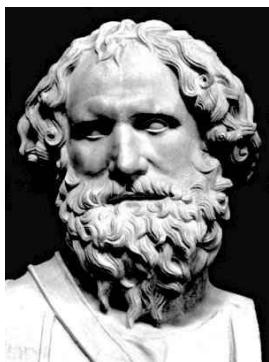
(b) 莱布尼茨

图 1-1 牛顿与莱布尼茨像

注: 图像均来源于网络

微积分学研究的起源可以追溯到古希腊哲学家、数学家、物理学家阿基米德(Archimedes, 约公元前287–公元前212年, 图1-2(a))所处的年代<sup>[1,2]</sup>, 在约公元前250年,阿基米德建立了积分学的雏形,并研究了无穷小量(infinitesimals),利用杠杆平衡原理和穷竭法(method of exhaustion)给出了将物体切成无穷小量的薄

片计算面积和体积的步骤,还提出并证明了很多定理。公元3世纪,中国数学家刘徽(约公元225–295年,图1-2(b))在公元265年出版了《九章算术注》,给出了圆面积的计算方法。公元5世纪,中国古代著名数学家祖冲之的儿子祖暅给出了球体积的计算公式,并给出了一条原理:“幂势既同,则积不容异”,亦即两个等高物体,在任何一个水平切面处,如果截面积相同,则两个物体的体积相同。



(a) 阿基米德



(b) 刘徽

图 1-2 阿基米德与刘徽像

注:图像均来源于网络

文献 [2] 列出了现代微积分学领域一些里程碑式的成果。

1666年,牛顿引入了流数(fluxion)描述即时速度。在其完成于1671年,出版于1736年的著作中,将连续变量称为流动量,将流动量的导数称为流数,提出了已知连续运动的路径,求给定时刻速度的微分法与已知运动速度求给定时间经过路径的积分法,还研究了无穷级数问题,奠定了数学分析这一全新的数学分支。

1676年,莱布尼茨使用了符号 $\int ydx$ ,积分符号是一个拉长的S字母,源于拉丁词 *summa*,表示求和,该表达式计算的是函数曲线下的面积。

1684年,莱布尼茨发表了微分学的第一篇论文。与牛顿在物理学上的出发点不同,莱布尼茨的研究领域是几何学,起源于对曲线任意点切线的研究。他明确给出了一阶微分的定义,并定义了微分符号 $dx$ 、 $dy$ 。图1-3给出了莱布尼茨的手迹<sup>[3]</sup>。

1687年,牛顿的科学巨著 *Philosophiae Naturalis Principia Mathematica* (自然哲学之数学原理)出版<sup>[4]</sup>,带有牛顿修改标记的扉页见图1-4。

1691年,法国数学家 Guillaume François Antoine L'Hôpital (1661–1704) 提出了著名的L'Hôpital法则,不过后人认为该法则是瑞士数学家 Johann Bernoulli (1667–1748) 最早发现并教给L'Hôpital的<sup>[5]</sup>。L'Hôpital的著作 *Analyse des Infin-*

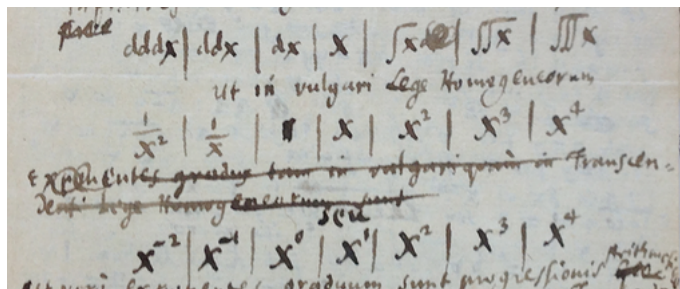


图 1-3 莱布尼茨手迹

注:图像来源于文献 [3]

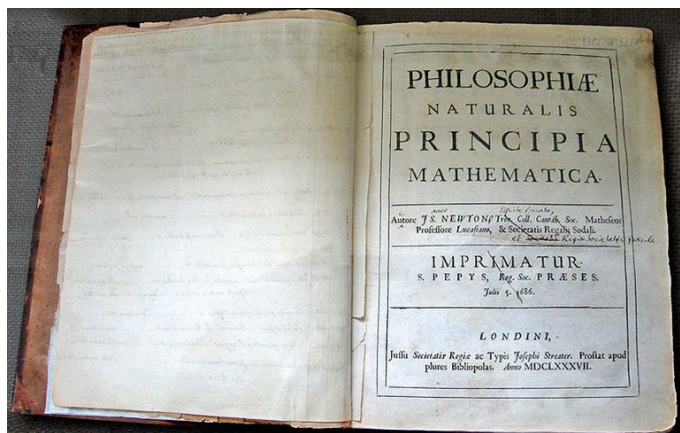


图 1-4 牛顿巨著的扉页

注:图像来源于网络

*iment Petits pour l'Intelligence des Lignes Courbes*(理解曲线的无穷小量分析)在当时有着广泛的影响。

1693年,莱布尼茨提出了微积分基本定理。

1715年,英国数学家Brook Taylor(1685–1731)提出了将函数表示成无穷级数求和的数学表达式,又称为泰勒(Taylor)级数,该级数的原始思想是苏格兰数学家、天文学家James Gregory(1638–1675)提出的。

1754年,法国数学家Jean-Baptiste le Rond d'Alembert(1717–1783)用极限的方法取代了无穷小量。

1772年,法国数学家Joseph-Louis Lagrange(1736–1813)首次使用了derived function一词,后来派生出现在用的derivative。Lagrange在1797年还提出了均值定理(mean-value theorem)。

1817年,波西米亚王国(今捷克)数学家 Bernard Bolzano (1781–1848) 提出了函数连续性的概念,并给出了极限的  $\varepsilon$ - $\delta$  定义,还提出了中值定理(intermediate-value theorem)。

1853年,德国数学家 Georg Friedrich Bernhard Riemann (1826–1866) 提出了给定区间的实函数的 Riemann 积分,该积分是 1823 年法国数学家 Baron Augustin-Louis Cauchy (1789–1857) 提出的 Cauchy 积分的拓展。

1861年,德国数学家 Karl Theodor Wilhelm Weierstrass (1815–1897) 提出了极值定理(extreme-value theorem),并于 1874 年改善了 Bolzano 的极限定义与函数连续性的概念,形成了更严格的数学理论与体系。

牛顿与莱布尼茨创立的微积分学是很多科学与工程分支的基础。单变量与多元函数微积分、函数极限、级数求和、泰勒(Taylor)级数展开、傅里叶(Fourier)级数展开、常微分方程等问题的求解是微积分学的重要内容。

## 1.2 本书的主要内容

关于 MATLAB 语言编程与基础知识的系统介绍可以参见本丛书的第 I 卷《MATLAB 程序设计基础》。本卷将侧重介绍如何利用 MATLAB 语言直接求解微积分问题,将以工程学科教授的视角介绍微积分问题的求解方法与应用方法。

由于微积分学的分析对象都是针对函数与序列的,所以第 2 章将系统介绍如何利用 MATLAB 语言描述函数与序列,以及函数与序列的绘图方法,为下一步介绍微积分学奠定一个坚实的基础。

第 3 章将系统地介绍如何利用 MATLAB 符号运算工具箱中的工具直接求解单变量函数的极限、单边极限及多元函数极限问题,数项级数与函数项级数的求解问题。配合图形显示方法,可以更好地探索极限问题的趋近与变化过程。

第 4 章介绍导数与微分问题的计算机求解方法,首先由导数定义出发,演示导数求解的方法,然后针对各种各样的函数探讨基于 MATLAB 的导数、偏导数与微分的求解方法,编写了一些参数方程、隐函数等导数解析求解的通用 MATLAB 函数,并介绍场的基本概念与导数的应用问题求解。

第 5 章介绍不定积分、定积分、多重积分、反常积分等问题的解析求解方法,介绍定积分的各种应用与计算方法,还将介绍两类曲线积分和两类曲面积分,并编写了通用的 MATLAB 求解函数。

第 6 章将介绍数项与函数项级数的求和与求积方法,对于不能求和求积的无穷级数还将探讨其收敛性与判定方法。给定单变量函数与多元函数的泰勒幂级数

展开, 给定函数的傅里叶级数逼近方法, 并利用 MATLAB 的绘图功能研究有限的拟合效果和适用范围。第6章还探讨了一般函数的连分式、Padé有理函数逼近方法与复变函数的 Laurent 级数表示方法。

这部分内容大都是解析求解和解析推导, 属于计算机代数研究的领域, 用传统的数值分析方法是不能求解的。对不熟悉计算机代数系统开发的读者来说, 用 C 这样的底层语言直接进行解析解推导有极大难度, 必须使用计算机数学语言完成这类问题的分析与求解。

在实际科学与工程研究中, 微积分问题解析求解有时也面临困难。例如, 若函数本身未知, 只由科学实验测出的一些实验数据, 则无法用推导的方式通过数据对其代表的函数求导或求积分, 而需要通过数值方式进行数值微积分运算。第7章中将探讨单变量与多元函数的数值微分计算问题。在实际应用中还有很多函数积分的解析解不存在, 所以需要通过数值积分的算法进行近似。第8章将介绍用数值算法求取函数积分及重积分问题的求解方法, 以及基于样条插值的数值微积分方法。第9章将介绍积分变换的基本知识, 侧重于拉普拉斯 (Laplace) 变换、傅里叶变换、梅林 (Mellin) 变换、汉克尔 (Hankel) 变换、 $z$  变换的求解方法, 如果解析解并存在, 则将探讨积分变换问题的数值求解方法。

如果微积分的阶次可以选择为非整数, 还可以引入一个新的学科——分数阶微积分学。第10章将系统介绍分数阶微积分学问题及其 MATLAB 求解方法, 给出分数阶微积分运算的高精度数值计算方法。

本书不是严格意义下的数学教程, 其目标是尽可能地用计算机可以接受的指令, 将微积分学的问题直接推给计算机, 利用计算机得出问题的解。可以充分利用计算机与计算机数学语言的强大功能, 直接求出问题的解, 并对以往不易求解的问题得出探索性的解。另外, 如果某类微积分学问题没有解析解, 也可以通过数值微分与数值积分技术得出问题的数值解。

本书推荐使用的工具是 MATLAB 的符号运算工具箱, 对某些不能直接求解的问题, 我们将编写出相应的 MATLAB 通用函数, 直接求解相关的问题, 对一些较复杂的问题, 也推荐使用其他学者开发 MATLAB 工具箱或求解函数。本书的总体目标是帮助读者更快、更简洁、更准确、更可靠地得出研究问题的解。

通过本书的初步学习, 读者可能会发现借助计算机去求解曾令很多学生望而生畏的吉米多维奇《数学分析习题集》<sup>[6]</sup> 中的绝大部分计算问题变得轻而易举。利用本书介绍的新视角, 读者甚至可以重新审视微积分学领域学习过的问题, 并利用强大的计算机技术创造性地解决相关的数学问题。

## 第2章



# 函数与序列

函数与序列是整个微积分学的重要研究对象,学习函数与序列的表示方法与处理方法是学习整个微积分学的基础。

2.1 节先给出函数的定义,并介绍单变量函数、多元函数的 MATLAB 表示方法与绘制方法。2.2 节将介绍反函数、复合函数、隐函数、参数方程等不同的函数类型及 MATLAB 处理。2.3 节将探讨函数的奇偶性及其判定方法。2.4 节介绍复变函数的定义与表示方法,并介绍复平面映射等方面的内容。2.5 节将介绍序列的 MATLAB 表示,并介绍利用计算机工具绘制序列演变趋势的方法。

## 2.1 函数与映射

### 2.1.1 函数的定义与描述

本节首先给出函数的数学定义,然后介绍各种各样函数的 MATLAB 表示方法,并介绍函数的绘图方法。

**定义 2-1** 设在某变化过程中有两个变量  $x$ 、 $y$ ,如果对于  $x$  在某一范围内的每个确定的值, $y$  都有唯一确定的值与它对应,那么就称  $y$  是  $x$  的函数,记作  $y = f(x)$ , $x$  称为自变量, $y$  称为因变量。函数  $y = f(x)$  还可以描述成变量  $x$  到变量  $y$  的映射,在数学上记作  $f: x \rightarrow y$ 。

这种函数更严格地应该称为单变量函数(univariate function)。

**定义 2-2** 如果自变量为  $x_1, x_2, \dots, x_n$ ,且它们之间相互独立,则函数称为多元函数(multivariate function),记作  $y = f(x_1, x_2, \dots, x_n)$ 。

**定义 2-3** 自变量  $x$  取值的集合称为函数的定义域,和自变量  $x$  对应的  $y$  的值称为函数值,函数值的集合称为函数的值域。

函数是微积分学中重要的研究对象。本节先介绍常用超越函数(transcendental functions)的概念与计算方法,再介绍函数的 MATLAB 描述与绘图方法。

### 2.1.2 常用超越函数的MATLAB计算

变量的加、减、乘、除运算在MATLAB下可以用“+”“-”“\*”“/”符号直接实现,如 $a+b*c/(d+e)$ ;乘方和开方运算可以使用“^”符号直接实现,如 $a^{(1/3)}$ 。

**定义 2-4** 超越函数通常指变量之间的关系不能用有限次加、减、乘、除、乘方、开方运算表示的函数,例如指数函数、对数函数、三角函数等。

MATLAB 超越函数的一般调用格式为  $y=\text{fun}(x)$ , 其中  $x$  可以是标量,也可以是向量、矩阵或多维数组,可以是符号变量,也可以是实数、复数。得出的  $y$  的数据结构和维数与  $x$  相同。下面列出常用的超越函数及相应的MATLAB函数。

(1) **三角函数**: 正弦、余弦、正切、余切的MATLAB函数分别为  $\sin()$ 、 $\cos()$ 、 $\tan()$ 、 $\cot()$ ; 正割(余弦的倒数)、余割(正弦的倒数)函数可以由  $\sec()$ 、 $\csc()$  函数计算; 双曲正弦  $\sinh x = (e^x - e^{-x})/2$ 、双曲余弦  $\cosh x = (e^x + e^{-x})/2$  可以由  $\sinh()$ 、 $\cosh()$  函数直接计算。三角函数默认的单位是弧度,若想使用角度单位,则可以用单位变换公式  $x=180x_1/\pi$  进行转换,还可以使用  $\text{find}()$  这类函数。

(2) **反三角函数**: 如果三角函数名前有一个  $a$ , 如  $\text{asin}()$ , 则计算反三角函数。

(3) **指数函数**:  $e$  的指数函数  $e^x$  可以由  $\text{exp}()$  函数直接计算。

(4) **对数函数**: 自然对数  $\ln x$  可以由  $\log()$  计算; 常用对数  $\lg x$  可以由  $\log_{10}()$  计算; 一般的对数函数  $\log_a x$  可以由对数的换底公式  $\log(x)/\log(a)$  直接计算。

### 2.1.3 一般函数的MATLAB表示

对于一般的函数  $y = f(x)$ , 在MATLAB下有两种直接的表示方法,一种方法是申明  $x$  为符号变量,然后由一般的符号表达式直接将函数表示出来,另一种是在符号型数据结构下直接定义出函数,即使用符号函数的方法。

**例 2-1** 试用MATLAB符号变量形式表示出下面函数

$$f(x) = ax^2 + bx + c, \quad g(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$$

**解** 可以使用下面两种格式来定义这两个函数

```
>> syms x y a b c; f1=a*x^2+b*x+c; f2(x)=a*x^2+b*x+c;
    g1=(x^2-2*x)*exp(-x^2-y^2-x*y); %符号表达式格式
    g2(x,y)=(x^2-2*x)*exp(-x^2-y^2-x*y); %符号函数格式
```

如果想求出  $f(5)$  与  $g(a^2, a+b)$  的值, 用两种方法需要使用不同的方式调用函数, 从调用语句的方便程度看, 显然第二种函数定义的格式更方便, 也更接近数学原型。另外, 两类定义的数据结构是不同的,  $f_1$  为符号变量  $\text{sym}$ ,  $f_2$  为符号函数  $\text{symfun}$ 。

```
>> subs(f1,x,5), f2(5), class(f1), class(f2)
    subs(g1,{x,y},{a^2,a+b}), g2(a^2,a+b)
```

如果  $x$  为实数,还可以用下面的格式直接定义实函数,并对其求值

```
>> syms x real; f(x)=a*x^2+b*x+c;
```

这里介绍的两种将函数输入 MATLAB 环境的方法对问题的描述没有本质上区别,在实际应用中用哪种方法方便就使用哪种方法。不过,由于函数描述的方式可以直接指出自变量,更接近数学原型,所以在可能的情况下建议使用函数格式直接定义数学中的函数。

使用符号函数时应当注意,如果定义了  $f(x)$ ,再定义  $f(x,y)$  会导致错误,所以定义  $f(x,y)$  之前应该给出 `clear f` 命令先清除以前的定义。

### 2.1.4 函数的曲线与曲面表示

一般情况下,一元函数可以由曲线表示,二元函数可以用曲面表示。这里将介绍如何在 MATLAB 环境下绘制函数的曲线与曲面。

(1) **一元函数:** 如果已知一元函数  $y = f(x)$ ,则可以用 `fplot()` 函数直接绘制其曲线,调用格式为

```
fplot(f), %默认绘图区间[-5,5],或 fplot(f,[x_m,x_M])
```

其中  $f$  可以为匿名函数,也可以为符号表达式或符号函数。不过在使用匿名函数时,应该采用点运算的形式。

在早期版本 MATLAB 中,还可以使用 `ezplot()` 函数直接绘制一元显函数曲线,该函数的调用格式为

```
ezplot(f), %默认绘图区间[-2π,2π],或 ezplot(f,[x_m,x_M])
```

其中  $f$  可以是符号表达式或符号函数,还可以是数值计算中常用的字符串,相比之下, `ezplot()` 函数可以比较好地自动选择坐标轴,但不能直接绘制分段函数,而 `fplot()` 函数可以。

(2) **二元函数:** 若已知二元函数  $z = f(x,y)$ ,则可以用 `fsurf()` 函数绘制其曲面图,其调用格式为

```
fsurf(f) %默认区间[-5,5],或
```

```
fsurf(f,[x_m,x_M]), fsurf(f,[x_m,x_M,y_m,y_M])
```

也可以用早期版本函数 `ezsurf()` 绘制,该函数的调用格式为

```
ezsurf(f), %默认区间[-2π,2π],或
```

```
ezsurf(f,[x_m,x_M]), ezsurf(f,[x_m,x_M,y_m,y_M])
```

其中  $f$  可以是符号表达式、符号函数或字符串。

还有一些特殊形式的函数,其曲线、曲面的绘制方法将在 2.2 节进一步介绍。

## 2.2 不同函数的MATLAB表示

除了前面介绍的常规函数之外,在实际应用中还有各种特殊形式的函数,如反函数、复合函数、隐函数与参数方程等,本节将分别介绍这些函数的定义、MATLAB输入方法与图形绘制方法。

### 2.2.1 反函数

**定义 2-5** 如果已知函数  $y = f(x)$ , 并且对值域内的每个  $y$ , 都有一个函数  $x = g(y)$  与之对应, 且这种对应关系是一一对应, 则称  $x = g(y)$  为  $y = f(x)$  的反函数, 记作  $f^{-1}(y)$ 。

`finverse()` 函数可以求出一些给定函数的反函数。值得指出的是, 即使一个函数是一一对应的函数, 有时也不一定能得出其反函数的解析解。

**例 2-2** 试求出函数  $f(x) = 1 + \ln(x+1)$  的反函数。

**解** 先输入原函数, 然后求出其反函数, 再将反函数的自变量替换成  $y$ , 则可以得出原函数的反函数为  $x = g(y) = f^{-1}(y) = e^{y-1} - 1$ 。

```
>> syms x y; f(x)=1+log(x+1); g(x)=finverse(f,x); g(y)
```

### 2.2.2 复合函数

**定义 2-6** 如果函数  $y = f(u)$  为  $u$  的函数, 而  $u = g(x)$  为  $x$  的函数, 则  $y$  也是  $x$  的函数,  $y = f(g(x))$ , 这种函数称为复合函数。

在实际应用中, 还可以将一个函数嵌入另一个函数来构造复合函数。例如, 已知两个函数  $f(x)$  和  $g(x)$ , 则可以构造出两个复合函数  $f(g(x))$  和  $g(f(x))$ 。复合函数  $f(g(x))$  又记作  $(f \circ g)(x)$ 。

**例 2-3** 已知数学函数  $f(x) = \frac{x \sin x}{\sqrt{x^2+2}(x+5)}$ ,  $g(x) = \tan x$ , 试求出复合函数  $f(g(x))$  和  $g(f(x))$ 。

**解** 可以有三种方法在 MATLAB 下表示复合函数, 第一种是通过变量替换的模式计算, 第二种是调用符号运算工具箱的 `compose()` 函数顺序计算, 第三种是按照函数的方式直接计算。下面将给出这三种方法的 MATLAB 语句。

```
>> syms x; f=x*sin(x)/sqrt(x^2+2)/(x+5); g=tan(x);
    F1=subs(f,x,g), F2=subs(g,x,f)           % 第一种方法: 变量替换的方法
    F3=compose(f,g,x), F4=compose(g,f,x)      % 第二种方法: 直接合成的方法
    f(x)=x*sin(x)/sqrt(x^2+2)/(x+5); g(x)=tan(x);
    F5=f(g), F6=g(f)                         % 第三种方法: 函数求值的方法
```

这三种方法得出的结果是一致的。从使用的方便程度看,第三种方法值得推荐。

$$F_1 = F_3 = F_5 = \frac{\sin \tan x \tan x}{(\tan x + 5) \sqrt{\tan^2 x + 2}}, \quad F_2 = F_4 = F_6 = \tan \left[ \frac{x \sin x}{\sqrt{x^2 + 2} (x + 5)} \right]$$

例2-4 Dawson函数是一个特殊函数,其定义为

$$\text{daw}(z) = e^{-z^2} \int_0^z e^{\tau^2} d\tau \quad (2-2-1)$$

已知 Dawson 函数的一阶导数可以由下式计算

$$\frac{d}{dz} \text{daw}(z) = 1 - 2z \text{daw}(z) \quad (2-2-2)$$

试用复合函数的方法递推求出该函数的二阶、三阶和四阶导数。

**解** 可以先在 MATLAB 下申明符号函数  $f(x)$ , 然后采用变量替换的方式求取 Dawson 函数的高阶导数。

```
>> syms x daw(x); f=1-2*x*daw; f1=f;
    for i=1:3
        f1=diff(f1,x);
        f1=collect(expand(subs(f1,diff(daw,x),f)),daw)
    end
```

这样可以得出一系列 Dawson 函数的导数为

$$\begin{aligned} \text{daw}''(x) &= (4x^2 - 2)\text{daw}(x) - 2x \\ \text{daw}'''(x) &= (-8x^3 + 12x)\text{daw}(x) + 4x^2 - 4 \\ \text{daw}^{(4)}(x) &= (16x^4 - 48x^2 + 12)\text{daw}(x) - 8x^3 + 20x \end{aligned}$$

### 2.2.3 分段函数的描述

**定义 2-7** 如果某函数在不同的自变量取值范围内函数的表达式也是不同的, 则这类函数称为分段函数。

例2-5 分段函数的一个实例 [7]

$$p(x_1, x_2) = \begin{cases} 0.5457 \exp(-0.75x_2^2 - 3.75x_1^2 - 1.5x_1), & x_1 + x_2 > 1 \\ 0.7575 \exp(-x_2^2 - 6x_1^2), & -1 < x_1 + x_2 \leq 1 \\ 0.5457 \exp(-0.75x_2^2 - 3.75x_1^2 + 1.5x_1), & x_1 + x_2 \leq -1 \end{cases}$$

在符号型数据结构下, 一般不适合采用 `if` 类语句描述函数。如何来输入分段函数呢? MATLAB 符号运算工具箱的新引擎——MuPAD 支持分段函数的使用, 提供了底层的 `piecewise()` 函数描述分段函数, 不过该函数在 MATLAB 下是不能直接调用的, 必须为其编写一个接口函数, 例如编写接口函数 `piecewise1()`。

```
function f=piecewise1(varargin), str=[];
try %试探结构:如果输入变元成对出现则处理,否则报错
    for i=1:2:length(varargin), %处理分段函数对
```

```

    str=[str,['',varargin{i},'','varargin{i+1}',''],'];
end
catch, error('Input arguments should be given in pairs.'), end
f=feval(symengine,'piecewise',str(1:end-1)); %构造总的分段函数

```

该函数的调用格式为  $f=\text{piecewise1}(\text{var}_1, \text{var}_2, \dots)$ , 其中输入变元  $\text{var}_j$  必须成对出现, 都应该由字符串给出, 前面一个是条件, 后面一个是该条件下的函数表达式, 条件中的逻辑运算可以由关键词 `and`、`or` 或 `not` 表示。

该函数使用了 `try`, `catch` 结构, 确保输入变元成对出现, 否则将给出错误信息。由于生成的字符串中最后一个字符是多余的逗号, 所以使用 `end-1` 下标舍弃它。

从 MATLAB R2016b 版开始, 提供了 `piecewise()` 接口函数, 用符号表达式表示条件和函数, 如果有该函数建议尽量采用 `piecewise()` 函数。

**例 2-6** 考虑饱和非线性函数  $y = \begin{cases} 1.1 \operatorname{sign}(x), & |x| > 1.1 \\ x, & |x| \leq 1.1 \end{cases}$ , 试绘制其曲线。

**解** 可以首先描述分段函数, 然后绘制该函数曲线。值得指出的是, 由于符号运算本身的局限性, 分段函数定义的符号表达式不能用 `ezplot()` 函数直接绘制。

```

>> syms x
f(x)=piecewise(abs(x)>1.1,1.1*sign(x), abs(x)<=1.1,x);
fplot(f) %直接绘制饱和函数

```

如果  $|x| \leq 1.1$  在数学上表示成  $-1.1 \leq x \leq 1.1$ , 也可以将其理解成  $x \geq -1.1$  且  $x \leq 1.1$ , 这时相应的符号表达式表示应该为  $x \geq -1.1 \ \& \ x \leq 1.1$ 。

**例 2-7** 试用符号表达式表示例 2-5 中给出的分段函数, 并绘制其三维曲面。

**解** 可以先申明符号变量, 并用分段函数的形式定义出二元函数  $p(x_1, x_2)$ , 如果用常规方法, 可以生成一组网格数据, 将网格数据代入函数就可以直接进行函数求值, 得出的函数值再由符号型数据结构转换成双精度数据, 并直接绘制出表面图。这里采用直接绘制的方法得出表面图, 如图 2-1 所示。

```

>> syms x1 x2;
p(x1,x2)=piecewise(x1+x2>1,...
    0.5457*exp(-0.75*x2^2-3.75*x1^2-1.5*x1),...
    -1<x1+x2 & x1+x2<=1,0.7575*exp(-x2^2-6*x1^2),...
    x1+x2<=-1,0.5457*exp(-0.75*x2^2-3.75*x1^2+1.5*x1))
fsurf(p,[-2,2,-2,2])

```

**例 2-8** 使用分段函数的一个优点是可以同时给出函数的定义域。试在 MATLAB 下表示函数  $f(x) = 2 \sin 3x$ , 其中  $-\pi/6 \leq x \leq \pi/6$ 。

**解** 该函数及其定义域可以由分段函数直接描述出来, 如果自变量  $x$  在定义域范围内, 就可以正常求值; 不在定义域内, 则返回 NaN。

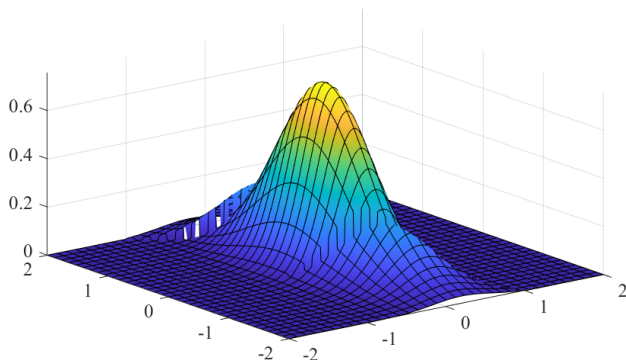


图 2-1 二元函数的表面图

```
>> syms x; f(x)=piecewise(-pi/6<=x & x<=pi/6,2*sin(3*x))
    f(1/7), f(sym(1/7)), f(5)
```

#### 2.2.4 隐函数

在实际应用中经常会遇到一类函数, 满足  $f(x, y) = 0$ , 但没有办法将其写出  $y = g(x)$  的显式形式, 这类函数称为隐函数。本节将讨论二元隐函数和三元隐函数的数学形式及其图形表示方法。

**定义 2-8** 隐函数的一般数学形式为  $f(x_1, x_2, \dots, x_n) = 0$ 。

##### 1) 二元隐函数

以二元隐函数  $f(x, y) = 0$  为例, 在实际应用中可以有两种方法描述这样的隐函数, 一种是符号表达式, 另一种是字符串方式。描述了隐函数的数学模型, 则可以通过 `ezplot()` 函数直接绘制隐函数曲线。

还可以用符号表达式或匿名函数  $f$  描述隐函数, 然后调用 `fimplicit(f)` 函数绘制隐函数曲线, 这时匿名函数中应该采用点运算。默认的自变量区间为  $[-5, 5]$ 。如果人为地指定自变量范围, 则可以由下面格式绘制隐函数曲线:

```
fimplicit(f, [x_m, x_M]) 或 fimplicit(f, [x_m, x_M, y_m, y_M])
```

**例 2-9** 已知隐函数  $x^2 \sin(x + y^2) + y^2 e^x + y + 5 \cos(x^2 + y) = 0$ , 试绘制出  $x$  与  $y$  之间的关系曲线。

**解** 用两种方法描述这样的隐函数, 再调用 `ezplot()` 函数, 就可以绘制出隐函数的曲线, 如图 2-2 所示。这两种描述原函数的方法会得出完全一致的二维曲线。

```
>> f1='x^2*sin(x+y^2)+y^2*exp(x+y)+5*cos(x^2+y)';
    syms x y; f2(x,y)=x^2*sin(x+y^2)+y^2*exp(x+y)+5*cos(x^2+y);
    ezplot(f1), figure, ezplot(f2)
```

如果使用新函数 `fimplicit()`, 则可以给出下面语句, 得出完全一致的结果。

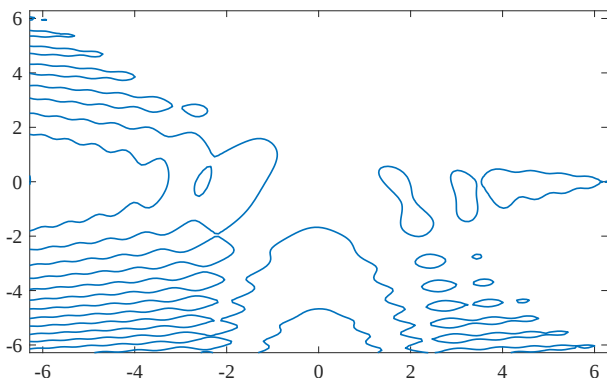


图 2-2 隐函数的相平面图

```
>> fimplicit(f2,[-2*pi,2*pi])
```

## 2) 三元隐函数

对三元隐函数  $f(x, y, z) = 0$  而言, MATLAB 提供了 `fimplicit3()` 函数, 可以直接绘制隐函数曲面。可以用符号表达式或匿名函数直接描述三元隐函数  $f$ , 然后由 `fimplicit3(f)` 直接绘制隐函数曲面。当然, 用户也可以自行指定坐标轴范围。

```
fimplicit3(f,[xm,xM]), % x 轴范围
fimplicit3(f,[xm,xM,ym,yM]), % x、y 轴范围
fimplicit3(f,[xm,xM,ym,yM,zm,zM]), % x、y、z 轴范围
```

例 2-10 试绘制下面的三元隐函数曲面

$$(x^2 + y^2 + z^2 + 2y - 1)((x^2 + y^2 + z^2 - 2y - 1)^2 - 8z^2) + 16xz(x^2 + y^2 + z^2 - 2y - 1) = 0$$

解 可以由符号函数直接描述三元隐函数, 然后调用 `fimplicit3()` 函数直接绘制曲面的表面图, 如图 2-3 所示。

```
>> syms x y z
f(x,y,z)=(x^2+y^2+z^2+2*y-1)*((x^2+y^2+z^2-2*y-1)^2-8*z^2) ...
+16*x*z*(x^2+y^2+z^2-2*y-1);
fimplicit3(f,[-4 4,-4,4,-4,4])
```

## 2.2.5 参数方程

**定义 2-9** 如果一组因变量可以定义成一个或多个自变量的函数, 则这组因变量称为参数方程。

这里将分别考虑两种参数方程——单自变量的参数方程与双自变量的参数方程的 MATLAB 表示与图形绘制方法。

**定义 2-10** 如果有一个二维平面内的质点, 已知该质点的坐标是时间的函数,

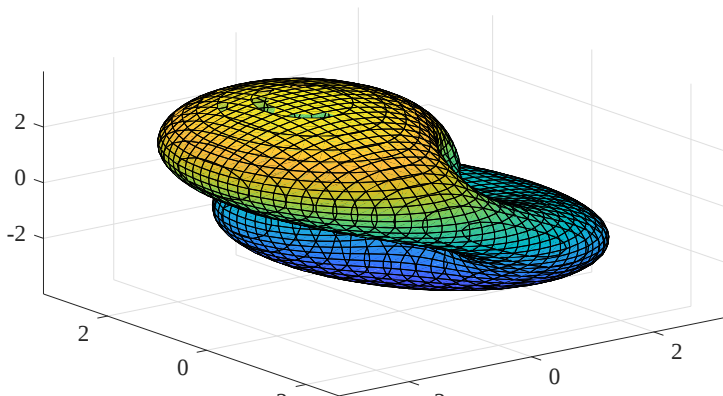


图 2-3 三元隐函数曲面

$x = f(t), y = g(t)$ , 则称这类函数为单自变量的参数方程,  $t$  称为参数。

参数方程的轨迹有两种绘制方法, 一种是生成一个时间向量  $t$ , 计算出  $x$  和  $y$  向量, 则通过 `plot(x,y)` 直接绘制出轨迹的相平面曲线, 类似的方法还可以处理质点的三维相空间轨迹绘制问题; 另一种方法是用符号表达式分别表示  $f(t)$  和  $g(t)$  函数, 然后由 `fplot(f,g,[tm,tM])` 函数即可绘制相平面曲线, 其中  $t \in [t_m, t_M]$ 。函数 `ezplot()` 也可以用同样的格式绘制参数方程曲线。

例 2-11 若  $a = 8, b = 5$ , 且  $t \in (0, 10\pi)$ , 试绘制下面参数方程的曲线

$$x = (a+b)\cos t - b\cos[(a/b+1)t], \quad y = (a+b)\sin t - b\sin[(a/b+1)t]$$

解 可以由符号表达式直接描述参数方程的数学模型, 这样即可绘制出该函数的曲线, 如图 2-4 所示。

```
>> a=8; b=5; syms t;
    x=(a+b)*cos(t)-b*cos((a/b+1)*t);
    y=(a+b)*sin(t)-b*sin((a/b+1)*t); fplot(x,y,[0,10*pi])
```

例 2-12 已知参数方程

$$x(t) = (0.5\sqrt{t} + 2)e^{-0.1t}\cos t^2, \quad y(t) = 2\sqrt[3]{|\sin t|}\sin 0.1\sqrt[3]{t^2}$$

其中  $t \in (0, 10)$ , 试绘制  $x(t)$  与  $y(t)$  间的关系曲线。

解 如果仿照前面的方法绘制曲线, 则可以直接给出下面语句。不过就这个具体的例子而言, 由 `ezplot()` 很难控制步距选择与坐标系尺度调整, 所以得出的结果往往不能令人满意, 可以用 `fplot()` 函数直接求解。

```
>> syms t; x=(0.5*sqrt(t)+2)*exp(-0.1*t)*cos(t^2);
    y=2*abs(sin(t))^(1/3)*sin(0.1*t^(2/3)); fplot(x,y,[0,10])
```

可以考虑采用另一种方法绘制相平面曲线, 得出的结果如图 2-5 所示。可以看出, 这样的曲线变化是杂乱无章的, 如果不借助计算机很难绘制其曲线。

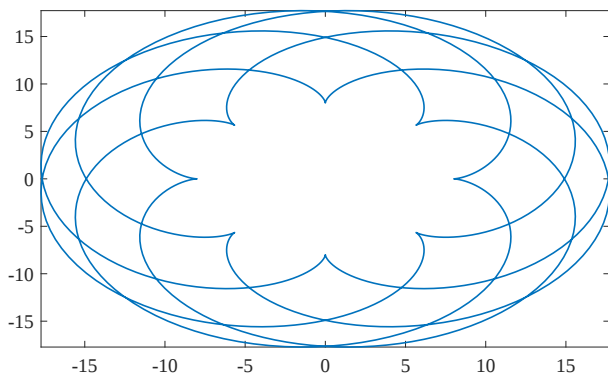


图 2-4 参数方程的曲线图

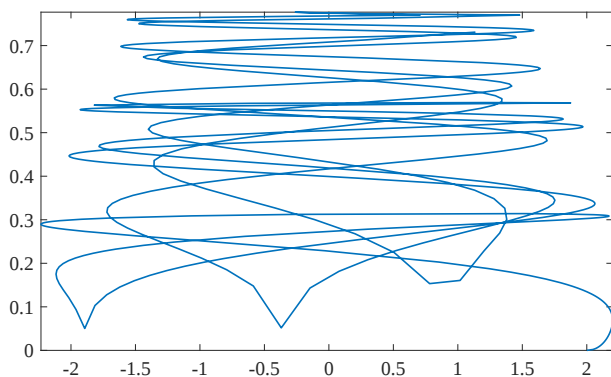


图 2-5 参数方程的相平面图

```
>> t=0:0.001:10; x=(0.5*sqrt(t)+2).*exp(-0.1*t).*cos(t.^2);
    y=2*abs(sin(t)).^(1/3).*sin(0.1*t.^(2/3)); plot(x,y)
```

定义 2-11 双参数的参数方程由下式给出：

$$x = x(u, v), \quad y = y(u, v), \quad z = z(u, v) \quad (2-2-3)$$

若  $u_m \leq u \leq u_M, v_m \leq v \leq v_M$ , 则由 `fsurf(x, y, z, [um, uM, vm, vM])` 函数可以直接绘制三维表面图,  $u, v$  变量的默认区间为  $[-5, 5]$ 。`ezsurf()` 函数也可以用于这类曲面的绘制, 调用格式是一致的, 默认区间为  $(-2\pi, 2\pi)$ 。

例 2-13 著名的 Möbius 带可以由数学模型

$$x = \cos u + v \cos u \cos u/2, \quad y = \sin u + v \sin u \cos u/2, \quad z = v \sin u/2$$

描述。如果  $0 \leq u \leq 2\pi, -0.5 \leq v \leq 0.5$ , 试绘制 Möbius 带的三维表面图。

解 首先需要申明两个符号变量  $u, v$ , 并将参数方程输入到 MATLAB 环境中, 这样就可以由下面的语句直接绘制 Möbius 带, 得出如图 2-6 所示的表面图。

```
>> syms u v;
x=cos(u)+v*cos(u)*cos(u/2); y=sin(u)+v*sin(u)*cos(u/2);
z=v*sin(u/2); fsurf(x,y,z,[0,2*pi,-0.5,0.5]) % Möbius 带的绘制
```

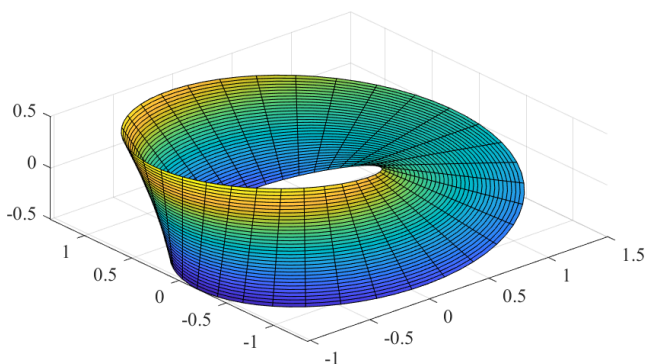


图 2-6 Möbius 带的表面图(图形旋转后结果)

## 2.2.6 极坐标函数

通常使用的坐标系是直角坐标系, 又称为 Cartesian 坐标系, 而实际应用中有可能会用到其他坐标系, 例如极坐标系(polar coordinates)。

**定义 2-12** 考虑平面内由原点出发的一条射线, 由射线角度  $\theta$  与射线长度  $\rho$  可以构造一个坐标系, 这样的坐标系称为极坐标系。通常极坐标可以表示为显式表达式  $\rho = \psi(\theta)$ 。

**定理 2-1** 极坐标系到直角坐标系的转换公式为  $x = \rho \cos \theta, y = \rho \sin \theta$ 。

**定理 2-2** 直角坐标系到极坐标系的转换公式为  $\rho = \sqrt{x^2 + y^2}, \theta = \arctan y/x$ 。

在 MATLAB 中可以使用 `polarplot()` 函数直接绘制出显式极坐标函数  $\rho = \psi(\theta)$  的曲线, 该函数的调用格式为 `polarplot( $\theta, \rho$ )`, 其中  $\theta$  与  $\rho$  为数据向量,  $\theta$  的单位为弧度(rad)。早期版本 `polar()` 函数的作用是一致的, 但不建议使用。

**例 2-14** 试绘制极坐标函数  $\rho = e^{-0.1\theta} \sin 4\theta$  的曲线。

**解** 选择自变量  $\theta$  的变化范围  $\theta \in (0, 10\pi)$ , 则可以生成函数的极坐标数据, 并绘制出极坐标图形, 如图 2-7 所示。

```
>> theta=0:0.001:10*pi;
rho=exp(-0.1*theta).*sin(4*theta); polarplot(theta,rho)
```

很多极坐标函数是周期性函数, 选择一个周期内的  $\theta$  值就可以把极坐标曲线绘制出来。这里的函数不是周期性的, 所以不论取多大范围都不能绘制完整的极坐标曲线。

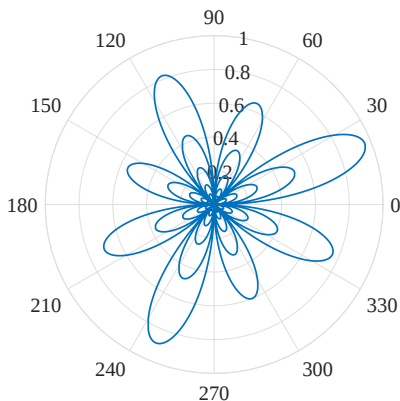


图 2-7 极坐标曲线

## 2.3 奇函数与偶函数

**定义 2-13** 假设函数  $f(x)$  定义于对称区间  $-L \leq x \leq L$ 。如果  $f(x) = f(-x)$ ，则  $f(x)$  为偶函数；如果  $f(x) = -f(-x)$ ，则函数  $f(x)$  为奇函数。

其实，利用 MATLAB 的符号运算功能可以很容易判定出一个给定的函数是奇函数还是偶函数，只需化简  $f(x) + f(-x)$  与  $f(x) - f(-x)$ ，观察哪个为零就可以了。如果前者为零，则  $f(x)$  是奇函数；后者为零，则  $f(x)$  为偶函数；如果都非零，则  $f(x)$  既不是奇函数也不是偶函数。下面将通过一个简单例子判定函数的奇偶性。

**例 2-15** 试判定  $f(x) = \sqrt{1+x+x^2} - \sqrt{1-x+x^2}$  函数的奇偶性。

**解** 申明给出的函数，并化简两个式子，可以发现前者为零，说明  $f(x)$  为奇函数。由 `fplot()` 函数还可以绘制出函数的曲线，如图 2-8 所示。从图中可以看出，该函数确实是奇函数。

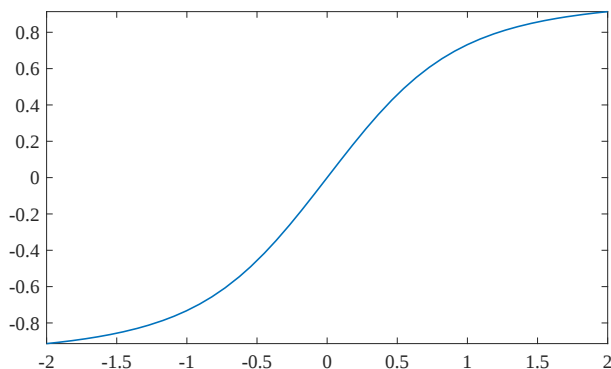


图 2-8 奇函数曲线

```
>> syms x; f(x)=sqrt(1+x+x^2)-sqrt(1-x+x^2);
    simplify(f(x)+f(-x)), simplify(f(x)-f(-x)), fplot(f,[-2,2])
```

## 2.4 复变函数与映射

自变量为实数的函数称为实函数, 自变量为复数的函数称为复变函数。本节将介绍复变函数的基本定义与简单运算, 并介绍复变函数的映射及复变函数 Riemann 面的绘制方法。

### 2.4.1 复数矩阵及其变换

MATLAB 可以直接表示复数矩阵。假设已知一个复数矩阵  $Z$ , 则可以使用简单函数对该矩阵进行如下变换:

- (1) 共轭复数矩阵  $Z_1 = \text{conj}(Z)$ 。
- (2) 实部、虚部提取  $R = \text{real}(Z)$ ,  $I = \text{imag}(Z)$ 。
- (3) 幅值、相位表示  $A = \text{abs}(Z)$ ,  $P = \text{angle}(Z)$ , 其中相位的单位为弧度。

其实, 这里的  $Z$  并不局限于矩阵, 还可以是多维数组或符号表达式。

### 2.4.2 复变函数的映射

**定义 2-14** 若某函数  $f(z)$  的自变量  $z$  为复数, 则该函数称为复变函数。

由于复数是 MATLAB 最基本的数据结构, 在大多数算法中均未特别区分实数和复数, 所以前面介绍的绝大多数内容均可以直接用于复变函数的分析。

**例 2-16** 已知某复变函数为  $f(z) = \frac{z^2 + 3z + 4}{(z - 1)^5}$ , 其中  $z$  为复数变量, 试求  $f(-j\sqrt{5})$ 。

**解** 由下面语句即可得出所需的结果为  $f_1 = -7/243 + 3\sqrt{5}j/972$ 。

```
>> syms z; f(z)=(z^2+3*z+4)/(z-1)^5; f1=simplify(f(-sqrt(-5)))
```

在复变函数中, 一种很重要的数学变换形式是映射, 即函数可以从一个变量  $z$  变换成另一个变量  $w$  的函数, 其中  $z = g(w)$  为给定的函数。经常使用的映射是平移映射  $z = w + \gamma$ , 反演映射  $z = 1/w$  和双线性映射  $z = (aw + b)/(cw + d)$ 。这里  $\gamma$  为给定复数,  $a, b, c, d$  为给定实数。平移映射将函数的原点平移到  $\gamma$  点; 反演映射可以将单位圆内外的点相互映射; 双线性映射实现直线与圆的相互映射。

**例 2-17** 考虑例 2-16 中给出的复变函数  $f(z)$ , 试得出  $z = (s - 1)/(s + 1)$  下的映射函数  $F(s)$ 。

**解** 映射问题可以由下面命令直接得出, 得出的结果需要化简。

```
>> syms z s; f(z)=(z^2+3*z+4)/(z-1)^5;
    F(s)=simplify(f((s-1)/(s+1)))
```

该语句可以直接得出映射函数为

$$F(s) = -\frac{1}{16}(s+1)^3(4s^2+3s+1)$$

**例 2-18** 试将左半平面内的数据  $s$  通过双线性变换  $z = (s+1)/(s-1)$  映射成  $z$  平面内的点,并观察其分布。

**解** 假设实轴范围选择为  $(-a, 0)$ , 纵轴范围选择为  $(-b, b)$ , 并令  $a = 1, b = 3$ , 则可以用网格方法生成一些  $s$  左半平面的点。另外,还可以在区域内绘制一条垂直直线  $x = -0.55$  和一条水平直线  $y = 1.65$ , 如图 2-9 所示。

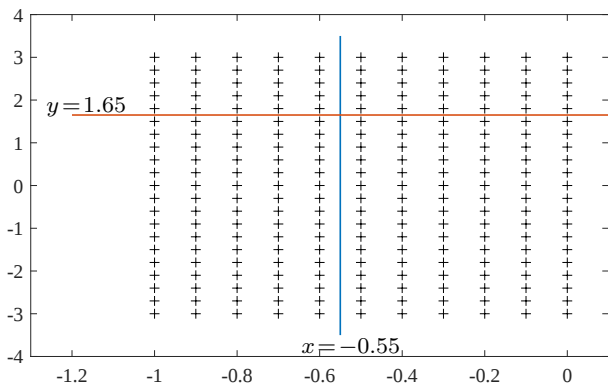


图 2-9  $s$  域的样本点分布示意图

```
>> [x,y]=meshgrid(-1:0.1:0,-3:0.3:3); %生成均匀分布的样本点
s=x+sqrt(-1)*y; plot(s,'+'); axis([-1.3 0.1 -4 4]) %原始数据分布
s1=-0.55+(-3.5:0.1:3.5)*sqrt(-1); hold on; plot(s1),
s2=(-1.2:0.1:0.2)+1.65*sqrt(-1); plot(s2), hold off
```

通过双线性变换的直接映射,可以得出  $z$  平面内的映射点,如图 2-10 所示,该图还给出了  $s$  平面内水平、垂直线的映射结果。

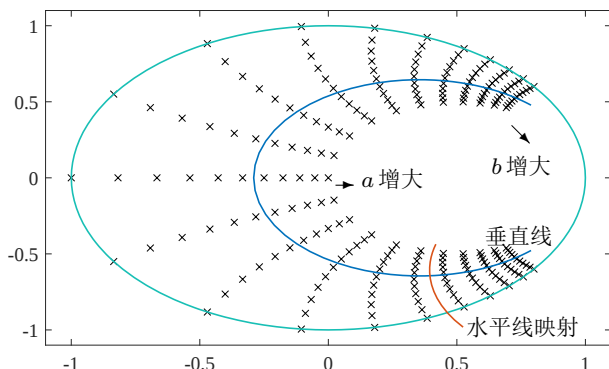
```
>> z=(s+1)./(s-1); plot(z,'x'), hold on
ezplot('x^2+y^2=1'), z1=(s1+1)./(s1-1); %映射点
plot(z1), z2=(s2+1)./(s2-1); plot(z2), hold off
```

可见,通过双线性变换,可以将左半平面内的点映射成  $z$  平面单位圆中的点。不过从图 2-10 中给出的结果看,映射不是很完全,因为  $a, b$  取值不是足够大,增加  $a$  和  $b$  的值则可能得出覆盖全单位圆的映射结果。

一般情况下,人们只知道通过这种变换,  $s$  左半平面上的点可以映射到单位圆内,但不知道具体会如何映射,如果借助计算机工具就可以更好地观察具体的映射方式。

### 2.4.3 Riemann 曲面的绘制

复变函数映射的三维图形表示和实函数是不一致的,复变函数映射应该首先采用函数 `cplxgrid()` 生成极坐标网格,用户可以根据给定的单值复变函数公式计

图 2-10  $z$  域的映射效果

算出变量  $f$ , 然后由 `cplxmap()` 函数绘制该复变函数的映射曲面, 这类曲面又称为 Riemann 面。这些函数具体调用格式为

`z=cplxgrid(n); %给出语句计算  $f$ ; cplxmap(z,f)`

例 2-19 试绘制出复变函数  $f(z) = z^3 \sin z^2$  的映射曲面。

解 由下面的语句可以立即绘制出相应的复变函数的映射曲面, 如图 2-11 所示。

`>> z=cplxgrid(50); f=z.^3.*sin(z.^2); cplxmap(z,f)`

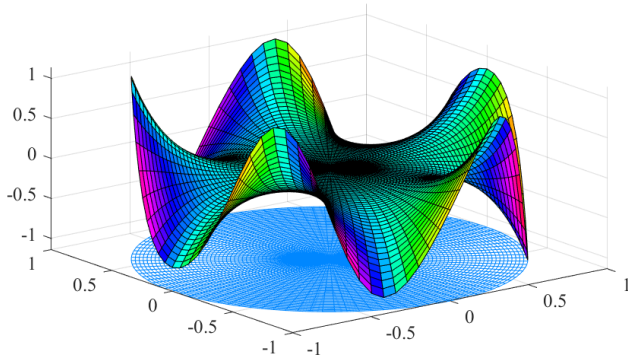


图 2-11 复变函数的 Riemann 面

对于复数变量  $z$ , 多值复变函数的 Riemann 面可能有多个分支, 这些分支又称为 Riemann 叶 (Riemann sheet), 例如  $f(z) = \sqrt[n]{z}$  就有  $n$  个分支。MATLAB 提供了绘制其各次方根 Riemann 曲面的函数 `cplxroot(n)`, 可以直接绘制出  $\sqrt[n]{z}$  的 Riemann 曲面。

例 2-20 试绘制  $\sqrt[3]{z}$  和  $\sqrt[4]{z}$  的 Riemann 曲面。

解 绘制这些方根函数的 Riemann 前面甚至不用事先生成网格, 由 `cplxroot()` 函数可以直接绘制出  $\sqrt[3]{z}$  和  $\sqrt[4]{z}$  的 Riemann 面, 如图 2-12、图 2-13 所示。

`>> cplxroot(3), figure, cplxroot(4) %分别绘制  $\sqrt[3]{z}$  和  $\sqrt[4]{z}$  的 Riemann 面`

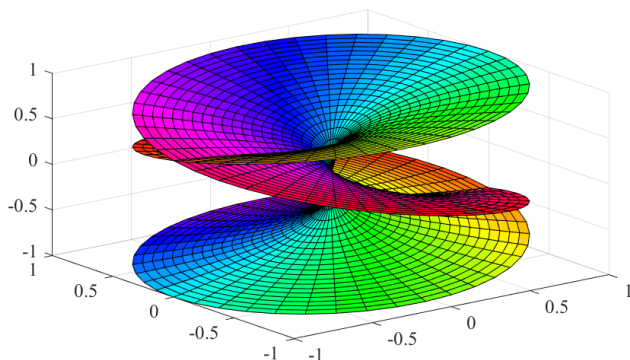


图 2-12 三次方根函数的 Riemann 面

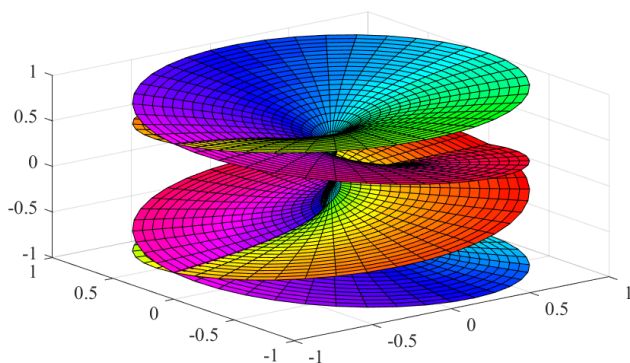


图 2-13 四次方根函数的 Riemann 面

从上面给出的 `cplxroot()` 函数可见，其局限性在于只能绘制出方根函数的 Riemann 面，对其他类型的多值复变函数无能为力，所以可以考虑扩展 `cplxmap()` 函数，将该函数另存为 `cplxmap1()` 函数，再删除掉其中的 `mesh()` 函数和 `hold` 语句，这样就可以考虑多值复变函数 Riemann 面的绘制了。

**例 2-21** 利用上述思路重新绘制  $\sqrt[3]{z}$  的 Riemann 面。

**解** 首先考虑重新绘制  $\sqrt[3]{z}$  函数的 Riemann 面。如果一个函数  $f_1(z)$  是  $f(z) = \sqrt[3]{z}$  的一个分支，则另两个分支可以由  $f_1(z)e^{-2j\pi/3}$  和  $f_1(z)e^{-4j\pi/3}$  求出。这样可以由下面语句直接绘制  $\sqrt[3]{z}$  的 Riemann 面，与图 2-12 给出的结果完全一致。

```
>> z=cplxgrid(30); f1=z.^(1/3); a=exp(-2i*pi/3);
    cplxmap1(z,f1), hold on; cplxmap1(z,a*f1)
    cplxmap1(z,a^2*f1); zlim([-1 1])
```

**例 2-22** 试绘制出复变函数  $f(z) = \sqrt{z^3 \sin z^2}$  的 Riemann 面。

**解** 可以生成复数型网格，求出第一 Riemann 叶的曲面数据，乘以标量  $e^{2\pi j/2}$

就能得到另外一个 Riemann 叶, 将它们在一个坐标系下绘制出来, 可以得出完整的 Riemann 面, 如图 2-14 所示。

```
>> z=cplxgrid(30); f=sqrt(z.^3.*sin(z.^2));
C=exp(pi*1i); cplxmap1(z,f), hold on
cplxmap1(z,f*C), zlim([-1.1 1.1])
```

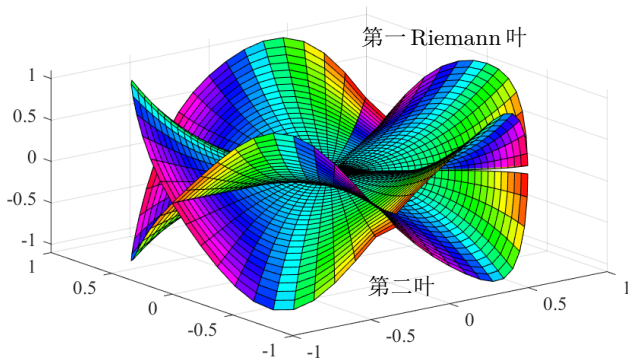


图 2-14 两个 Riemann 叶构成的曲面

## 2.5 序列与函数项序列

**定义 2-15** 按某种规律排列的数称为序列(sequence), 又称数列, 本书将统称为序列。

在日常生活与科学研究中经常会遇到序列, 例如

$$1, 2, \dots, n, \dots$$

$$2^0, 2^1, 2^2, \dots, 2^{n-1}, 2^n, \dots$$

在这两个序列中,  $n$  与  $2^{n-1}$  分别称为序列的通项。在 MATLAB 下可以用一个变量名描述序列的通项, 有了通项, 就相当于将序列直接描述出来了。

**定义 2-16** 序列的通项还可以为自变量  $x$  的函数, 例如

$$\frac{4 \sin x}{2\pi}, \frac{4 \sin 3x}{5\pi}, \frac{4 \sin 5x}{8\pi}, \dots, \frac{4 \sin(2n-1)x}{(3n-1)\pi}, \dots \quad (2-5-1)$$

这类序列称为函数项序列(functional sequence)。

当然, 如果用  $n$  的函数形式描述序列, 将使得其取值更方便。

**例 2-23** 试用 MATLAB 描述式 (2-5-1) 中给出的函数项序列, 并写出序列的第 1128 项。

**解** 其实在 MATLAB 下要描述函数项序列, 只需描述其通项就足够了, 该序列的第 1128 项为  $4 \sin(2255x)/(3383\pi)$ 。

```
>> syms n x; f(x,n)=4*sin((2*n-1)*x)/(3*n-1)/pi, f(x,1128)
```

例2-24 试观察下面的序列,考虑如果 $n$ 的值增大这个序列将如何变化。

$$x_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n} - \ln n$$

解 可以考虑用循环方式计算该序列的前 $n=40$ 项,序列的各项用火柴杆图的形式显示出来,如图2-15所示。可以看到,序列的值随着 $n$ 的增加将缓慢减小。

```
>> s0=1; n=40; for k=2:n, s0(k)=s0(k-1)+1/k; end
    stem(1:n,s0-log(1:n))
```

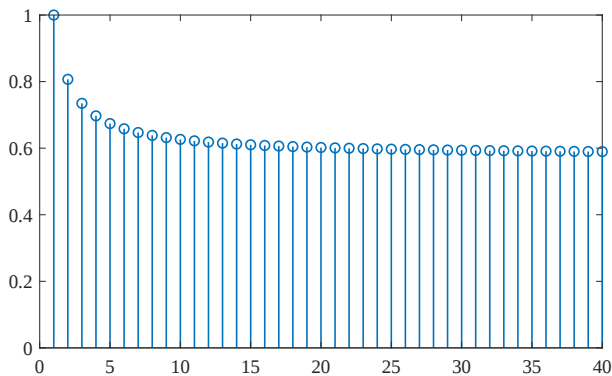


图 2-15 序列的演化趋势

可以预见,如果 $n$ 足够大,这个序列将收敛到某个值上。从理论上可知,这个收敛的值是Euler常数 $\gamma$ 。如果计算到第10000项,这个序列的值为0.577265664068165,比较接近于 $\gamma = 0.5772156649015328606065120900824$ 。

```
>> s0=1; n=10000; for k=2:n, s0=s0+1/k; end, s0-log(n)
    vpa(eulergamma)
```

上面语句中,如果加到10000000000项,则可以得出0.577215665057043。

## 本章习题

2.1 试证明下列恒等式

(1)  $e^{j\pi} + 1 = 0$

(2)  $\frac{1 - 2 \sin \alpha \cos \alpha}{\cos^2 \alpha - \sin^2 \alpha} = \frac{1 - \tan \alpha}{1 + \tan \alpha}$

2.2 若 $\phi(x) = (a^x + a^{-x})/2$ ,  $\psi(x) = (a^x - a^{-x})/2$ , 试证明

(1)  $\phi(x+y) = \phi(x)\phi(y) + \psi(x)\psi(y)$

(2)  $\psi(x+y) = \phi(x)\psi(y) + \psi(x)\phi(y)$

2.3 如果 $f(x) = \ln \left( \frac{1+x}{1-x} \right)$ , 试证明 $f(x) + f(y) = f \left( \frac{x+y}{1+xy} \right)$ 。

2.4 若  $f(x) = x^2 - x - 1$ , 试求  $f(f(f(f(f(f(f(f(f(x))))))))))$ 。如果结果是多项式, 多项式的最高阶次是多少?

2.5 Chebyshev 多项式的数学形式为  $T_1(x) = x$ ,  $T_2(x) = 2x^2 - 1$ ,  $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$ ,  $n = 3, 4, \dots$ 。试计算  $T_{10}(x)$ 。

2.6 试判定下面函数的奇偶性

$$(1) \sqrt[3]{(x+1)^2} + \sqrt[3]{(x-1)^2}$$

$$(2) \ln(x + \sqrt{1+x^2})$$

2.7 试求出下面函数的反函数

$$(1) y = \sqrt[3]{1-x^3}$$

$$(2) y = \ln(x/2)$$

$$(3) y = 2x + 3$$

2.8 试绘制下面的隐函数曲线

$$(r-3)\sqrt{r} + 0.75 + \sin 8\sqrt{r} \cos 6\theta - 0.75 \sin 5\theta = 0$$

其中  $r = x^2 + y^2$ ,  $\theta = \arctan(y/|x|)$ 。

2.9 试绘制下面给出的参数方程曲线

$$(1) x(t) = t^2 - t, y(t) = t^2 - t^3$$

$$(2) x(t) = a(\cos t + t \sin t), y(t) = a(\sin t - t \cos t)$$

2.10 给定参数方程  $x = \sin t$ ,  $y = \sin at$ ,  $z = \sin bt$ , 针对下面的有理数与无理数  $a, b$  取值, 试绘制出对应的二维、三维 Lissajous 图形

$$(1) a = 1/2, b = 1/3$$

$$(2) a = \sqrt[5]{2}, b = \sqrt{3}$$

2.11 试用符号元素工具箱支持的方式表达函数  $f(x) = x^5 + 3x^4 + 4x^3 + 2x^2 + 3x + 6$ , 并令  $x = (s-1)/(s+1)$ , 将  $f(x)$  表示成  $s$  的函数。试在  $x$  左半平面内生成一组样本点, 并观察这些样本点如果映射到  $s$  平面映射点的分布。

2.12 试绘制并旋转观察下面的参数方程表面图

$$\begin{cases} x = (3 + \cos u/2 \sin v - \sin u/2 \sin 2v) \cos u \\ y = (3 + \cos u/2 \sin v - \sin u/2 \sin 2v) \sin u \\ z = \sin u/2 \sin u + \cos u/2 \sin 2v \end{cases}$$

其中  $0 \leq u \leq 2\pi, 0 \leq v \leq 2\pi$ 。

2.13 假设某三维曲面由隐函数表示如下, 且感兴趣的区域为  $x, y, z \in (-1, 1)$ , 试绘制其三维曲面。

$$\psi(x, y, z) = x \sin(y + z^2) + y^2 \cos(x + z) + zx \cos(z + y^2) = 0$$

2.14 分别选取合适的  $\theta$  范围, 绘制出下列极坐标图形

(1)  $\rho = 1.0013\theta^2$

(2)  $\rho = \cos 7\theta/2$

(3)  $\rho = \sin \theta/\theta$

(4)  $\rho = 1 - \cos^3 7\theta$

2.15 试用 MATLAB 表示下面函数项序列的通项

$$12 \sin x, \frac{3}{2} \sin 2x, \frac{4}{9} \sin 3x, \dots, \frac{12}{n^3} \sin nx, \dots$$

2.16 试用图解法判定下面序列可能最终收敛到的值。

$$\sqrt{2}, \sqrt{2+\sqrt{2}}, \sqrt{2+\sqrt{2+\sqrt{2}}}, \sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2}}}}, \dots$$