

索取号：\_\_\_\_\_ 密级：\_\_\_\_\_

# 南京师范大学

## 硕士学位论文



研 究 生

指导教师

教授

培养单位

一级学科

二级学科

完成时间

2019 年 3 月 日

答辩时间

2019 年 5 月 日

# 学位论文独创性声明

本人郑重声明：所提交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果。本论文中除引文外，所有实验、数据和有关材料均是真实的。本论文中除引文和致谢的内容外，不包含其他人或其它机构已经发表或撰写过的研究成果。其他同志对本研究所做的贡献均已在论文中作了声明并表示了谢意。

学位论文作者签名： 日 期：

# 学位论文使用授权声明

研究生在校攻读学位期间论文工作的知识产权单位属南京师范大学。学校有权保留本学位论文的电子和纸质文档，可以借阅或上网公布本学位论文的部分或全部内容，可以采用影印、复印等手段保存、汇编本学位论文。学校可以向国家有关机关或机构送交论文的电子和纸质文档，允许论文被查阅和借阅。（保密论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密论文，密级：\_\_\_\_\_ 保  
密期限为\_\_\_\_\_ 年。

学位论文作者签名: \_\_\_\_\_ 指导教师签名: \_\_\_\_\_  
日 期: \_\_\_\_\_ 日 期: \_\_\_\_\_

## 摘 要

随着科技的发展和信息技术的不断提高,计算机技术、网络技术等现代科学技术已经在各行各业得到了广泛的应用,在教育领域的应用则体现为计算机辅助教育(CBE: Computer-Based Education)。《国家中长期教育改革和发展规划纲要(2010-2020年)》文件中指出“加快教育信息化进程”的任务之一就是要“加强优质教育资源的开发与应用”。数字化教育资源作为计算机辅助教育中的重要组成部分,不仅是我国教育信息化发展进程中重要发展产物,同时也是发展阶段的重要体现。数字化教育资源为教与学带来了诸多益处,如在教学过程中,可将部分教学内容实现计算机可视化,用以指导相关课程内容的教与学。特别对于抽象不易理解的学科内容,借助计算机技术将其形象化,使其更易于教师教学和学生理解应用。

例如初高中物理学科,在物理教学中存在有许多较难理解的抽象概念,单凭语言很难将其阐述清楚,因此对教师的教育教学提出了较高的要求。比如电场这一类看不见摸不着的数值场,但其又确实存在的物理现象。这一部分学习内容抽象且理论性强,需要学生具备较强的空间想象力和逻辑思维能力,是学生普遍感到难学的内容。在传统教学中,电场演示性内容多采用定性的方式绘制,其存在误差较大的问题,并且不同教材画法差异较大。随着计算机技术在各学科中的应用,有学者提出利用计算机技术实现电场可视化,为学习者提供直观、科学的学习材料。

在前人研究的基础之上,本文围绕数值场绘制技术这一课题展开研究,提出了一种新的数值场绘制方法。主要研究内容包括:数值场建模、数值场可视化以及场点优化压缩三方面内容。文中主要工作如下:

1. 首先对所绘数值场进行数学建模,本文采用隐函数描述数值场。此步工作是绘制数值场的首要工作同时也是最为基础的。

2. 在得到数值场数学模型的后,即可实现数值场可视化。数值场可视化工作主要分为两步:如何获取场点和如何将连接场点逼近场线。获取离散场点是数值场绘制的根本和基础。本文算法借鉴了二维 MC 算法获取平面场点,该方法原理简单,可操作性强。对于获得的离散场点,若是将其简单的通过线段将点与点连接起来,得到的是一系列折线,其准确度和光滑度难以满足要求。本文经过对已有曲线光滑算法进行对比和研究,采用 Hermite 插值算法逼近场线。

3. 在实现数值场绘制的基础上本文算法还对场点进行了优化处理,在保证场线的光滑度和准确度的情况下,以较少的“关键”场点即可逼近场线。如此可

减少场点数据存储空间和加快场线绘制速度。

在上述算法理论研究基础之上，本文在一定软硬件环境下实现了该算法。并绘制了静电场点电荷数值场（电场）。实验结果表明，所绘数值场中场线连续光滑准确。

**关键词：**计算机辅助教育；数值场；隐式曲线；Hermite 插值

## Abstract

With the development of science and information technology, computer technology, network technology and other modern science technology have been widely used in all walks of life. Of course, these technologies are also applied in the field of education, which is called Computer-Based Education (CBE). *The Outline of the National Plan for Medium - and Long-term Education Reform and Development (2010-2020)* states that one of the tasks of "accelerating the process of educational informatization" is to "strengthen the development and application of high-quality educational resources". As an important part of computer-aided education, digital educational resources are not only an important development product in the development process of educational informatization in China, but also an important embodiment of the development stage. Digital educational resources have brought many benefits to teaching and learning. For example, in the teaching process, some teaching contents can be visualized by computer to guide the teaching and learning of relevant course contents. Especially for the abstract and difficult to understand the subject content, with the help of computer technology to visualize it, make it easier for teachers and students to understand and apply.

For example, in physics teaching in junior and senior high school, there are many abstract concepts that are difficult to understand, which are difficult to be explained by language alone. Therefore, higher requirements are put forward for teachers' education and teaching. For instance, the electric field is a kind of invisible numerical field, but it is a real physical phenomenon. This part of the learning content is abstract and theoretical, which requires students to have a strong spatial imagination and logical thinking ability. So it is generally difficult for students to learn. In the traditional teaching, electric field demonstrative content is mostly drawn in qualitative way, which has the problem of large error. And the different textbooks adopt different drawing method. With the application of computer technology in various disciplines, some scholars put forward the use of computer technology to achieve electric field visualization, providing intuitive and scientific learning materials for learners.

Based on the previous research, this paper focuses on the numerical field rendering technology, and proposes a new numerical field rendering method. The main research contents include: numerical field modeling, numerical field

visualization and field point optimization and compression. The main related work of this paper is as follows:

(1) In this paper, the numerical field is modeled by implicit function. This step is the first and most basic work for drawing the numerical field.

(2) Based on the mathematical model of numerical field, the visualization of numerical field can be realized. The visualization of numerical field is mainly divided into two steps: how to obtain field points and how to connect field points to obtain field lines. Obtaining discrete field points is the fundamental of numerical field rendering. In this paper, the two-dimensional MC algorithm is used for reference to obtain planar field points. For the obtained discrete field points, if they are simply connected with each other through line segments. Then we will get a series of broken line graphs, whose accuracy and smoothness are difficult to meet the requirements. This paper compared and studied the existing curve smoothing algorithms and used Hermite interpolation algorithm to connect the points.

(3) On the basis of realizing the numerical field rendering, this algorithm also optimizes the compression processing of field points. It can promise the field lines with fewer "key" field points to guarantee the smoothness and accuracy of field lines. In this way, the data storage space of field points can be reduced and the drawing speed of field lines can be accelerated.

Based on the theoretical study of the above algorithm, the numerical field (electric field) is realized in this paper. The experimental results show that the line is smooth and accurate.

**Key words:** computer-aided education; numerical field; implicit curve; Hermite interpolation

# 目录

摘 要.....	I
Abstract.....	III
第 1 章 引言 .....	3
1.1 背景与意义 .....	3
1.2 国内外研究现状 .....	4
1.2.1 场点获取.....	4
1.2.2 场点连接.....	6
1.2.3 场点压缩.....	9
1.3 主要工作与内容安排 .....	11
1.3.1 主要工作 .....	11
1.3.2 论文结构 .....	12
第 2 章 理论基础.....	13
2.1 数值场.....	13
2.1.1 数值场 .....	13
2.1.2 数值场可视化 .....	13
2.2 隐式曲线.....	14
2.2.1 隐函数及隐函数曲线.....	14
2.2.2 隐性数值场.....	15
2.3 Hermite 插值 .....	16
2.3.1 Hermite 插值定义.....	16
2.3.2 Hermite 插值多项式及插值方法.....	16
第 3 章 场线绘制方法研究.....	18
3.1 数值场数学模型定义与获取 .....	18
3.2 二维 MC 方法与场点捕捉.....	19
3.2.1 MC 方法.....	19
3.2.2 基于二维 MC 方法的场点获取 .....	20
3.2.3 消除二维 MC 方法的二义性.....	22
3.3 基于面积法的场点压缩 .....	23
3.3.1 线段求交算法.....	24

3.3.2 平面复杂多边形的面积计算.....	24
3.4 场线提取与绘制 .....	25
3.5 本章小结.....	25
第 4 章 算法实现与实验结果.....	27
4.1 实验环境平台 .....	27
4.2 算法实现 .....	28
4.2.1 整体流程.....	28
4.2.2 数据结构组织.....	28
4.2.3 算法实现关键代码.....	30
4.3 单条场线优化效果对比 .....	33
4.4 数值场绘制效果 .....	37
4.5 本章小结 .....	40
第 5 章 基于本文算法的静电场点电荷可视化系统设计与实现.....	41
5.1 系统需求分析.....	41
5.1.1 用户需求.....	41
5.1.2 功能需求.....	41
5.2 系统界面设计.....	42
5.3 系统实现效果.....	43
5.3.1 电荷操作.....	43
5.3.2 电场控制的相关功能.....	44
5.4 本章小结.....	45
第 6 章 总结与展望.....	46
6.1 主要工作及研究结论 .....	46
6.2 主要创新 .....	47
6.3 未来工作展望 .....	47
参考文献 .....	错误!未定义书签。
在读期间发表的学术论文及研究成果 .....	错误!未定义书签。
致 谢.....	错误!未定义书签。



---

## 第 1 章 引言

本章主要介绍数值场绘制技术研究背景、意义和研究目的，分别从场点获取、场点压缩和场点连接三方面回顾了绘制数值场的相关技术的国内外研究现状，基于此确立了本文的主要研究工作，最后简要介绍了本文的内容安排。

### 1.1 背景与意义

《国家中长期教育改革和发展规划纲要（2010-2020 年）》文件中指出“加快教育信息化进程”的任务之一就是要“加强优质教育资源的开发与应用”<sup>[1]</sup>。随着科技的发展和信息技术的不断提高，计算机技术、网络技术等现代科学技术已经在各行各业得到了广泛的应用，在教育领域的应用则体现为计算机辅助教育(CBE: Computer-Based Education)。其中数字化教育资源则是计算机辅助教育中的重要组成部分，不仅是我国教育信息化发展进程中重要发展产物，同时也是发展阶段的重要体现<sup>[2]</sup>。数字化教育资源是指通过计算机技术、信息技术及多媒体技术等，将传统教学资源以数字形式存取、发布的信息资源总和，其中包括数字形式的文本资源、图片资源、视频资源、软件资源等<sup>[3]</sup>。

数字化教育资源为教与学带来了诸多益处，如在教学过程中，可将部分教学内容实现计算机可视化，用以指导相关课程内容的教与学。特别是对于一些较为抽象不易理解的学科内容，借助计算机技术将其形象化，使其更易于教师教学和学生理解应用<sup>[4]</sup>。

例如初高中物理学科，在物理教学中存在有许多较难理解的、抽象概念，单凭语言很难将其阐述清楚，因此对教师的教育教学提出了较高的要求。例如电场、磁场以及波的衍射这一类型看不见摸不着的数值场，但其又确实存在的物理现象。这一部分学习内容概念抽象、理论性强，需要学生具备较强的空间想象力和逻辑思维能力，是学生普遍感到难学的内容。在传统教学中，教材中的电场线和磁场线等多采用定性的方式绘制，其存在误差较大的问题，并且不同教材画法差异较大<sup>[5]</sup>。

随着计算机技术在各学科中的应用，有学者提出利用计算机技术将电场线、磁场线进行可视化，为学习者提供直观、科学的学习材料。目前相关研究多是建立在推导出来的电场分布公式的基础上借助 MATLAB 工具实现场线可视化。如文献<sup>[6]</sup>中在推导出一维电荷分析系统的电场线方程的基础上通过 MATLAB 中的函数绘制出了一维电荷系统的电场线，结果表明此种场线绘制方式既科学又优于传统的手绘图像；另外在文献<sup>[7]</sup>中也提出了利用 MATLAB 中的图形绘制函数绘制场图，并且还实现了三维电荷交互的功能。但是目前相关场线绘制技术有一定的

局限性，多数只能服务于电场或磁场的场线绘制，不具有推广性。

基于此，本文在前人研究的基础之上，在已知数值场数学模型的基础上，提出一种基于C++语言和OpenGL图形接口的简便、准确且快速的数值场绘制技术。从理论上讲，该方法可推广应用于电场、温度场、波的衍射等场的绘制，丰富了物理教学内容中场的计算机可视化技术；从实践应用上来讲，数值场绘制的研究为有关物理图像演示性课件的制作显然有着重要的意义。

## 1.2 国内外研究现状

数值场又被称为数量场或标量场，它是指空间中的每个点仅通过数学函数或物理量确定的数值大小即可完全表征的场。在研究物理学中的温度、压力、密度等在一定空间中的分布状态时，仅用数学上的一个代数量（数学函数）即可进行描绘，这些代数量即确定出相应的数值场，即温度场、压力场、密度场。另外电势场、浓度场等也均可采用相应代数量表示，因此其也都属于数值场<sup>[8]</sup>。

在数值场绘制中，最为关键的工作即：场线绘制，因此本文国内外研究现状中的内容主要阐述目前场线绘制的相关研究内容。经查阅文献可发现，场线绘制过程一般可概括为两部分内容：获取场点和连接场点。另外为减少存储空间，加快场线绘制速度，须对场点进行精简优化处理。因此国内外研究现状相关内容将分别从场点获取方法、场点精简压缩和场点连接三部分内容出发梳理国内外研究现状。

### 1.2.1 场点获取

获取场点是场线绘制的基础和关键工作。经查阅文献，目前获取场点的方法主要有取点跟踪法和网格法两种<sup>[9]</sup>，并将其梳理如下：

#### 1、取点跟踪法

取点跟踪法主要思想是在数值场内某一已知场点的周围选取符合条件的下一点，再以所选点为已知点选取再下一点，依照此步骤绘制场线，效率较高。取点跟踪法主要有定向选点法和切向定点法两类，其中定向选点即从给定方向选下一场点，切向选点即根据场点切向选取下一场点。相关详细获取场点的方法例如TN法、正负法、短线截交法和场细分法<sup>[9]</sup>。

TN(Tanget Normal)方法<sup>[10]</sup>基本原理是根据切向的前进和法向的调整确定下一场点。例如在构造一条折线(或圆弧组合的曲线) $\bar{\Gamma} = M_0 M_1 \dots M_n$ 时，从数值场内某一已知场点 $M_0$ 出发，沿 $M_0$ 切向 $\vec{T}_0$ 以 $\eta$ 为步长前进至 $Q_0$ ，在 $Q_0$ 处再沿 $M_0$ 处的法向 $\vec{N}_0$ 前进 $\delta$ 步长至 $M_1$ 处，再以 $M_0 M_1$ 作为 $M_1$ 处的近似切向 $\vec{T}_1$ 前进 $\eta$ 步长至 $Q_1$ 处，并由 $Q_1$ 沿 $M_0 M_1$ 法向 $\vec{N}_1$ 前进 $\delta$ 步长至 $M_2$ 处，依次类推，直到得到最后一点 $M_n$ 。其

中  $\delta$  和  $\eta$  的值的确定方法为：法向步长  $\delta$  即逼近精度，可根据实际情况确立；切向步长即  $\eta = \sqrt{2R\delta - \delta^2}$ ，其中  $R$  为最小曲率半径。TN 法查找场点只需已知起点和终点坐标，起点的法向和切向，以及最小曲率半径  $R$ 。这种方法原理简单，容易实现，但是难点在于如何选择一个合适的起始点<sup>[11]</sup>。

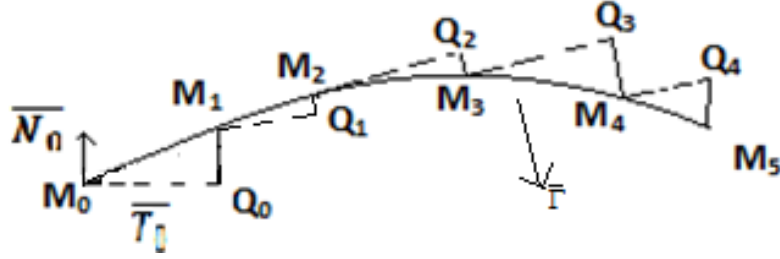


图 1.1 TN 法示意图

正负法<sup>[12]</sup>是一种基于数学函数  $F(x, y)$  值的大小来判别前进方向进而确定场点。例如：设有  $F(x, y) < 0$ ， $F(x, y) = 0$  和  $F(x, y) > 0$  是函数  $F(x, y)$  在平面上的点的集合，分别对应为： $G_-$ ， $G_0$  和  $G_+$ 。假设有满足  $F(x, y) = 0$  的起始点  $M_0 = (x_0, y_0)$ ，先沿  $\Delta x$  前进至  $p_1^- = (x_0 + \Delta x, y_0)$ ，因  $p_1^- \in G_-$ ，此时应在沿  $y$  选择沿正向或者负向向  $G_+$  区域前进。从图 1.2 中可以看出应该选择沿  $y$  轴正向前进  $\Delta y$ ，一直到刚好到达  $G_0$  或者进入到  $G_+$  区域。接着再选择合适方向向  $G_-$  区域前进，重复上述步骤，即可得到所有点。该方法中每次前进的步长均是固定的，若要得到光滑的曲线需将步长调到充分小<sup>[13]</sup>。

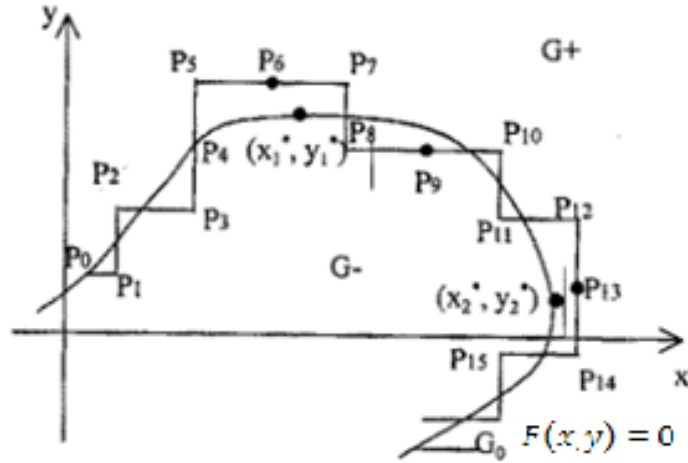


图 1.2 正负法示意图<sup>[13]</sup>

吴坚<sup>[14]</sup>等人提出短线截交法绘制场线，该方法主要是通过一系列平行于坐标轴的短线跟踪场线，当短线与场线相交时即利用二分法求出场点。首先设有  $F(x, y) < 0$ ， $F(x, y) = 0$  和  $F(x, y) > 0$  是函数  $F(x, y)$  在平面上的点的集合，分别对应为： $G_-$ ， $G_0$  和  $G_+$ ， $G_0$  即为场点集合， $G_-$  和  $G_+$  区域位于场线两侧。对于平面上任一点  $p$ ，若  $p \in G_+$  则将其极性记为正，若  $p \in G_-$  则将其极性记为负。开始

查找场点之前采用坐标轮换法确定位于场线上的起始点  $Q_0$ ，再利用短线截交法根据误差要求确定短线的搜索步长进而按照固定方向跟踪曲线，使其与  $G_0$  相交，通过二分法求解出交点即为所得场点。该方法具有简单、可靠、快速等特点，但是该方法的场点搜索过程较为盲目和繁琐<sup>[14]</sup>。

场细分法是一种常用的场线绘制方法<sup>[15]</sup>，该方法首先需要估计绘制区域，然后排除无关区域，并对有关区域不断细分，直到有关区域到达一个像素大小为止，如果排除不了，就将其绘制出来。该方法能可靠的画出曲线，且不需要对奇点进行任何处理，但是该方法过于保守，无法排除不在曲线上的场点，因此可能绘制出“胖”曲线<sup>[16]</sup>。寿华好<sup>[17]</sup>等人对细分法进行了改造，将自动微分法与泰勒方法相结合应用于细分法中，该方法可绘制任意复杂的场线，但是计算量较大。

## 2、网格法

网格法是一种较为“原始”的方法<sup>[18]</sup>，即通过将平面区域划分为  $n \times n$  的网格，再计算出每个网格边上的场点。例如绘制点电荷静电场电势线，采用网格法获取场点即将所给点电荷周围平面空间划分为网格，计算每个格点的电势，连接具有相同电势值的格点即成场线。这种方法计算量大，效率低。但是该方法整体性强。钱懿华等人对该方法进行了改进，以无穷远处的电势为 0，先计算可视化范围以内的电势，得到电势绝对值的  $U_{\max}$  和  $U_{\min}$ ，然后将  $U$  作如下变：

$$U' = -\ln \frac{U - U_{\min}}{U - U_{\max}} \quad (U > 0) \quad (1.1)$$

$$U' = -\ln \frac{-U - U_{\min}}{U - U_{\max}} \quad (U < 0) \quad (1.2)$$

再将根据  $U'$  差的分布绘制场线，这样画出的场线疏密得当<sup>[18]</sup>。

### 1.2.2 场点连接

数值场中的场线多为曲线，对于获取的场点如果仅仅是将其简单连接起来，那么得到的将是不具美观性的折现式的曲线，若应用于教学中易误导学生，不利于学生学习。因此需要对场点进行光滑处理。经查阅文献可知，B 样条曲线拟合、Bezier 曲线拟合、张力样条函数、分段三次多项式插值法均属于比较常见的曲线光滑处理算法。

#### 1、Bezier 曲线

Bezier 曲线是由法国雷诺汽车公司工程师皮埃尔·贝塞尔（Pierre Bézier）于 1962 年提出的<sup>[19]</sup>。对于给定空间  $n + 1$  个点  $P_i (i = 0, 1, 2, \dots, n)$ ，其 Bezier 曲线插值公式为：

$$C(u) = \sum_{i=1}^n B_{i,n}(u)P_i, \quad 0 \leq u \leq 1 \quad (1.3)$$

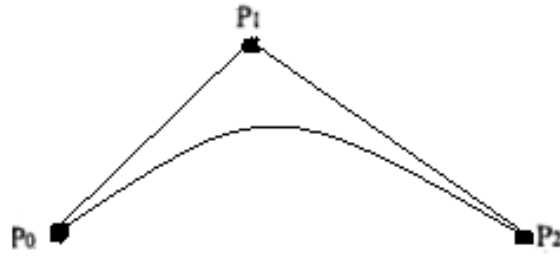
式中：  $B_{i,n}(u)$  是  $n$  次 Bezier 曲线的基函数，  $B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$ ， 其

中  $P_i$  为构成 Bezier 曲线的特征多边形，  $u$  为曲线的自变量参数<sup>[20]</sup>。

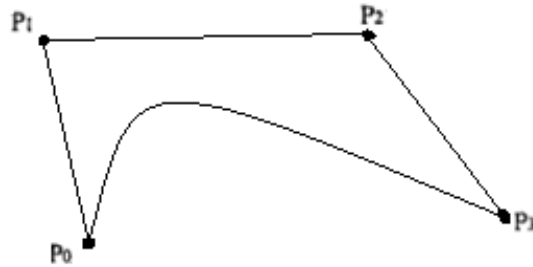
根据实际曲线绘制要求， 常用的是二次和三次 Bezier 曲线， 其相应示意图如图 1.3 (a) 和 (b)， 参数方程分别为式(1.4)和式(1.5)， 其中三次的 Bezier 曲线的连续性和光顺性较好。

$$B(u) = (1-u)^2 P_0 + 2u(1-u)P_1 + u^2 P_2, \quad 0 \leq u \leq 1 \quad (1.4)$$

$$B(u) = (1-u)^3 P_0 + 3u(1-u)^2 P_1 + 3u^2(1-u)P_2 + P_3 u^3, \quad 0 \leq u \leq 1 \quad (1.5)$$



(a) 二次 Bezier 曲线示意图



(b) 三次 Bezier 曲线示意图

图 1.3 Bezier 曲线示意图

从 Bezier 曲线绘制过程可以看出， 其一般是通过一组多边折线的顶点定义的， 且这些顶点中只有第一个顶点和最后一个顶点在曲线上， 其余的顶点则用于定义 Bezier 曲线的导数、 阶次和形状， 除此之外这组多边折线的第一条和最后一条表示了曲线两端点的切向。 这种曲线绘制曲线效果较好， 但是有一缺陷即： 若是改变曲线的一个控制点， 则会对整个曲线产生影响<sup>[21]</sup>。

## 2、 B 样条曲线

B 样条曲线拟合是在 Bezier 曲线基础之上发展起来的， 保留 Bezier 曲线的优点， B 样条中改变一个控制点的位置不会对整体曲线产生影响， 克服由于整体表示带来不具有局部性质的缺点<sup>[22]</sup>。 它是由  $n+1$  个控制点  $P_i$  ( $i = 0, 1, \dots, n$ ) 构成特征多边形的顶点。 例如已知三个平面离散点  $P_0$ 、  $P_1$ 、  $P_2$ ， 设这三点确立的二次抛

物线矢量参数方程为：  $P(t) = A_0 + A_1t + A_2t^2 (0 \leq t \leq 1)$ ，可推导出二次 B 样条曲线参数方程的矩阵形式为：

$$P(t) = \frac{1}{2} \cdot \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} \quad (1.6)$$

根据式 (1.6) 绘制出的 B 样条曲线如图 1.4 所示，由图可知曲线的起点是  $P_0$ 、 $P_1$  中点  $G_0$ ，终点是  $P_1$ 、 $P_2$  的中点，并分别与  $P_0P_1$  和  $P_1P_2$  两段线段相切<sup>[23]</sup>。

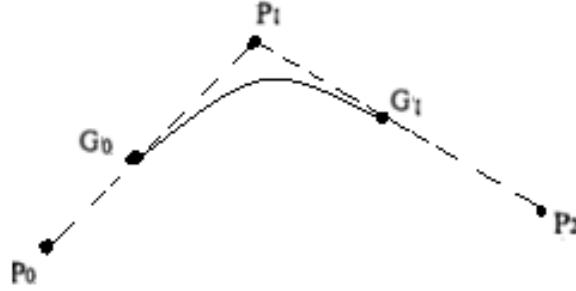


图 1.4 二次 B 样条曲线示意图

目前最常用的是二次和三次 B 样条曲线，一般二次 B 样条三个点绘制一条曲线，三次 B 样条四个点绘制一条曲线。

### 3、张力样条函数

张力样条函数是 D.G.Schweikert 于 1966 年提出的，目的是为消除三次样条产生插值曲线时产生多余“拐点”<sup>[24]</sup>。其基本思想为：设已知节点  $x_1 < x_2 < \dots < x_n$  与相应的函数值  $y_1, y_2, \dots, y_n$ ，根据张力样条分段插值函数  $f(x_i) = y_i (i = 1, 2, \dots, n)$  构造曲线。张力样条函数一般是由直线插值和两个双曲函数的线性组合：

$$f(x) = c_1 + c_2x + c_3sh\sigma x + c_4ch\sigma x \quad (1.7)$$

其中张力样条因子  $\sigma$  可控制拐点的位置和曲线的形态，给出不同的  $\sigma$  可得到不同拟合曲线。当  $\sigma \rightarrow 0$  时，可得到通过已知数据节点的光滑曲线；当  $\sigma \rightarrow +\infty$  时，张力样条函数退化为分段线性函数，并能保证线段不相交<sup>[25]</sup>。

### 4、分段三次多项式插值法

分段三次多项式插值法又叫五点光滑法，其基本思想是：设已知平面上有  $(x_i, y_i) (i = 1, 2, \dots, n)$  个数据点，每过两点  $P_i$  和  $P_{i+1}$  都有一个三次多项式函数和连续的一阶导数存在，使得其满足

$$y = a + bx + cx^2 + dx^3 \quad (1.8)$$

$$y = b + cx + dx^2 \quad (1.9)$$

求解 a、b、c、d 四个参数需建立以下四个方程：

$$y_i = f(x_i) ; y_{i+1} = f(x_{i+1}) \quad (1.10)$$



$$y'_i = f'(x_i), \quad y'_{i+1} = f'(x_{i+1})$$

即可得到分段三次多项式插值函数，进而求出插值点绘制曲线<sup>[26]</sup>。

比较这几种方法可知，B 样条法、Bezier 法拟合曲线只关心曲线的走向，不关心曲线是否经过各点。张力样条函数主要根据点的疏密程度自动调整张力系数，在点密集时张力样条函数退化为分段线性函数确保曲线光滑且不相交，点稀疏时以三次样条函数形式计算确保曲线既能通过点同时也很光滑，但是张力样条函数计算量较大。分段三次多项式插值法是在两个点之间建立三次多项式曲线方程，且要求该曲线有连续的一阶导数，各个节点的一阶导数是以一点为中心，相邻各两点来确定，因此也被称为五点光滑法，该方法绘制的光滑曲线通过已知的结点，但计算过程略显麻烦。

### 1.2.3 场点压缩

场点压缩通常是从生成场线的原始场点集中，在一定精度范围内抽取一个场点集，使该子集在数量上尽可能精简，并能满足场线绘制要求。其目的是删除冗余数据，减少数据储存量，节省存储空间，缩短图像绘制时间，从而加快绘制速度。目前，已有许多学者提出了关于点的精简压缩方法，例如自由压缩法、局部压缩法、Reumann-Witkam 法、光栅法、Duoglas — Puecker 法等。下面对这几类压缩方法做一介绍。

#### 1、自由压缩法

自由压缩法是一种原理比较简单的压缩方法，不用考虑相邻场点间的拓扑关系和数学特征，例如间隔取点与随机取点。间隔取点是在原始场点序列中，每隔一定数目的点就抽出点来组成新的场点序列，间隔点的数目可以根据不同精度要求进行选取。随机取点法顾名思义，即任意选择保留某些点。自由压缩法虽然压缩速度快，但会丢掉一些轮廓特征，对于高精度要求的曲线并不适用<sup>[27]</sup>。

#### 2、局部压缩法

局部压缩法是一种考虑到了相邻场点间的拓扑关系和数学特征的压缩方法。它包括垂距法、Walking-driver 法和曲率压缩法等。

垂距法是从场点的一端开始，按照顺序依次取三个场点  $P_{i-1}$ ,  $P_i$ ,  $P_{i+1}$ ，计算中间点  $P_i$  与其他两点连线的垂线距离  $d$ ，并与指定限定差值  $D$  比较。若  $d < D$ ，则去掉中间点，否则保留中间点。依次类推，直到这条曲线结束<sup>[27]</sup>。

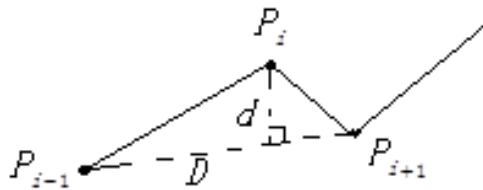


图 1.5 垂距法压缩场点示意图

Walking-driver 方法又叫法定长取点, 是由 Shelberg 于 1882 年提出的, 这种方法是通过一个分割器  $\lambda$  为唯一采样准则, 在曲线上每隔  $\lambda$  创建一个新的点, 使曲线发生位移, 该方法强调曲线的分形特征, 另外  $\lambda$  过大时, 会漏掉保形特征,  $\lambda$  过小时会产生大量的冗余数据 [28]。

曲率压缩法即根据场点曲率进行压缩, 其主要思想是设定曲率阈值  $T$ , 若原始场点集合中场点的曲率绝对值大于  $T$  或者该点的曲率是它某一领域的极值, 则保留此点。这种方法保留的场点质量较高, 但是阈值  $T$  的确定较为困难 [27]。

### 3、Reumann-Witkam 法

Reumann-Witkam 法是由 Reumann 和 Witkam 于 1974 年提出的一种非约束延伸式局部压缩法。该方法的主要思想是: 通过平行区域(条带)按顺序对场点依次搜索, 且搜索区域平行于开始的第一个线段, 结束于第一个相交的线段, 在搜索区域内对场点简化, 保留第一个场点和最后一个场点, 删除中间点。如图 1.6 所示, 条带区域与线段  $P_1P_2$  平行, 并且与线段  $P_3P_4$  相交于点  $P_3$ , 于是点  $P_2$ 、 $P_3$  均落在了条带区域, 因此我们可以舍去点  $P_2$ , 直接连接点  $P_1$ 、 $P_3$  [29]。

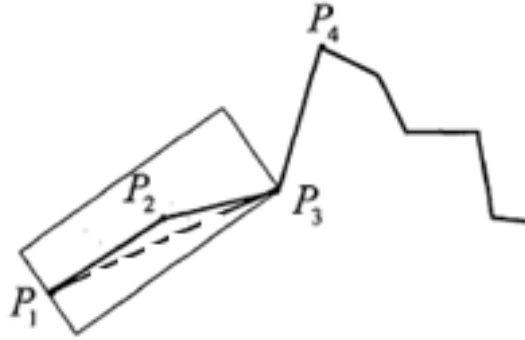


图 1.6 Reumann-Witkam 法压缩场点示意图 [27]

### 4、光栅法

光栅法又叫圆盘法, 也是一种非约束延伸式局部压缩法。该方法压缩场点的主要思想是: 先由点和圆盘(限差)来定义一个扇形, 再根据场点是否在扇形区域内来确定是否保留此点。如图 1.7 已知  $P_1$ 、 $P_2$ 、 $P_3$ 、 $P_4$  四个场点, 设定圆盘(限差)大小, 首先将圆盘置于点  $P_2$  处, 再由点  $P_1$  圆盘切线得到扇形  $AP_1B$ , 进而由此判断点  $P_3$  是否位于扇形区域内, 若是  $P_3$  不在扇形内, 保留  $P_2$ ; 若  $P_3$  在扇形内, 舍去  $P_2$ 。依次类推, 直到最后一个点 [27]。



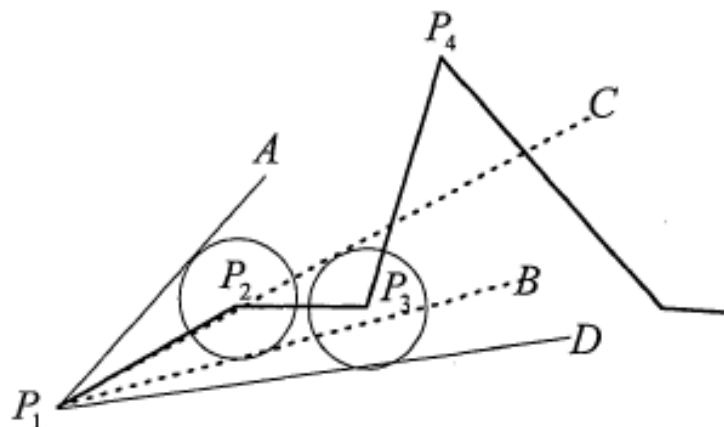


图 1.7 光栅法压缩场点示意图<sup>[27]</sup>

### 5、Duoglas — Puecker 法

Duoglas — Puecker 法是曲线压缩的经典算法，关注曲线的特征点。具体思想为：连接曲线的首末点成一条直线，求所有中间点到该直线的距离，找出最大距离  $d_{max}$ ，并将  $d_{max}$  与给定的限差  $D$  作比较。若  $d_{max} < D$ ，则这条曲线上的中间点可全部舍去，反之，保留  $d_{max}$  对应的坐标点，并以此为界点，把曲线分为两部分，对这两部分重复使用该方法。但是该方法计算工作量大，只有所有数据完成数字化后才能压缩<sup>[30]</sup>。

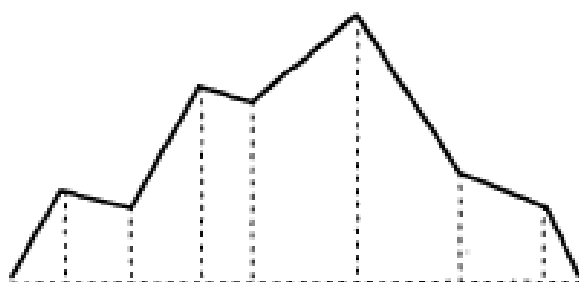


图 1.7 Duoglas — Puecker 法压缩场点示意图<sup>[27]</sup>

## 1.3 主要工作与内容安排

本文参照已有研究，提出了一种新的数值场绘制方法，首先对数值场进行数学建模，然后借鉴网格法和 MC(Marching cubes) 算法思想对绘制区域网格离散化获取场点；为加快场线绘制速度，减少场点存储空间，对场点数量进行了精简；绘制场线的过程则通过分段 Hermite 插值实现。从理论上来讲，该方法整体性较强，在可接受的计算量下，曲线的准确性和光滑度均较高。

### 1.3.1 主要工作

本文的主要内容包括以下几个方面：

- 1、数值场建模：在开始数值场绘制之前，本文首先根据具体物理规律和问题

---

情景进行数学建模，得出其数学函数。

2、数值场可视化：根据上一步获得的数值场函数，将其可视化是数值场绘制关键的一步。可视化主要涉及两步工作：如何获取场点和如何将场点连接为场线。如何获取离散场点是绘制场线的根本和基础。而将场点连接起来，看似简单，但若将其简单的通过线段将点与点连接起来，得到的是一系列折线图，其准确度和光滑度不满足要求，因此探寻如何准确逼近光滑的场线至关重要。

3、场点压缩：对获取的场点如何进行压缩精简，在保证场线的光滑度和准确度的情况下，以较少的“关键”场点即可逼近场线。如此可减少场点数据存储空间和加快场线绘制速度。此项内容对于高效实现准确度较高的场线的绘制具有重要意义。

### 1.3.2 论文结构

本文一共分为 5 章，各章的内容和结构组织如下：

第 1 章“引言”，本章主要介绍了数值场绘制技术的研究工作的背景和意义，从数值场绘制过程中的获取场点、压缩场点和连接场点三部分内容出发对国内外研究现状进行了分析，并阐述了本文的主要工作和内容安排。

第 2 章“理论基础”，本章论述了数值场绘制涉及到的相关理论，其中包括了数值场的定义与分类，隐式曲线的含义及其与数值场的关系，Hermite 插值定义及 Hermite 插值多项式的推导。

第 3 章“场线绘制方法”是本文重点内容，也是数值场绘制的核心内容，主要详细介绍了二维 MC 方法获取场点及如何消除场点二义性、面积法压缩场点过程、Hermite 插值连接场点等内容。

第 4 章“实验结果与分析”，主要介绍了实验环境，并介绍了本文中提出的数值场绘制技术的实验结果，然后对绘制过程中的相关数据进行了统计分析。

第 5 章“总结与展望”总结了本文的研究工作，指出了本文中的数值场绘制技术的不足，并提出了本文工作未来的研究方向。

---

## 第 2 章 理论基础

本章主要介绍本文所涉及到的相关理论，其中包括数值场的定义与分类，隐式曲线的含义及其与数值场的关系，Hermite 插值定义及 Hermite 插值多项式的推导。

### 2.1 数值场

场是由英国物理学家苏格拉底最早提出的一个物理学中的概念，它直观的描述了没有接触的物质粒子间的相互作用<sup>[31]</sup>。后来逐渐场被抽象为一个描述函数或者物理量的空间分布规律的数学概念：如果空间内的每个点都对应某个物理量或数学函数的一个确定值，则称在空间上确定了该物理量或数学函数的一个场<sup>[32]</sup>。通常以空间位置（坐标）为自变量的数学函数来描述场的属性，这样的数学函数包括矢量函数或标量函数，在物理学中分别对应矢量场和标量场。如电场、磁场属于矢量场，而如温度场、密度场则属于标量场。人们受到场概念的启发，将数域空间中的数据用场理论进行描述，于是就形成了数据场理论。在本文中所提到的数值场特指数据场理论中的标量场。

#### 2.1.1 数值场

数值场被称为数量场或标量场，它是指空间中的每个点仅通过数学函数或物理量确定的数值大小即可完全表征的场。在研究物理学中的温度、压力、密度等在一定空间中的分布状态时，仅用数学上的一个代数量（数学函数）进行描绘，这些代数量所确定出相应的数值场，即温度场、压力场、密度场，另外电势场、浓度场等也都属于数值场<sup>[33]</sup>。

#### 2.1.2 数值场可视化

在工程应用中，数据可视化技术主要分为标量场可视化技术、矢量场可视化技术和张量场可视化技术。本文只对标量场即本文中的数值场可视化技术进行研究。其中场可视化方法中主要包括：颜色映射法、等值线/等值面法、立体图法、层次分割法和体绘制法。

颜色映射法是通过建立数值和颜色的映射关系，将数据转换成彩色图形的技术和方法。立体图法、层次分割法和体绘制法均属于三维空间中的数值场绘制方法，其中立体图法是以三维立体图来展示平面数值场，层次分割法则是通过将层次和颜色进行映射从而得到立体图的分层效果图，而体绘制法主要是通过传递函数计算内部各区域的不透明度值，来去掉无关区域从而展示用户感兴趣的区域的可视化方法<sup>[33]</sup>。

等值线/等值面可视化方法中则是将数值场中的属性值相同的各场点连接起来，以曲线/曲面的形式表示该值在平面/空间中的分布情况<sup>[33]</sup>。例如：一个数值场  $U$  或  $V$  可以用数学函数  $U = (x, y, z)$  或  $V = (x, y)$  表示。其中  $U = (x, y, z)$  表示一个三维数值场，令  $U = (x, y, z) = C$  ( $C$  为任意常数)，该式在几何上表示一个曲面，虽然曲面上各点  $(x, y, z)$  不同，但是相应函数值相同均为  $C$ ，因此  $U = C$  的曲面也被称为数值场  $U$  的一个等值面。二维函数  $V = (x, y)$  则表示一个平面数值场，类似于三维数值场，平面数值场可通过  $V = (x, y) = C$  取不同的  $C$  值得到一系列不同的等值线（场线）。本文的工作则主要研究二维数值场的等值线可视化（绘制）技术。

文中二维数值场等值线可视化主要以隐函数形式的数值场数学模型为基础，通过二维 Marching Cubes 算法获取场点，在此基础上并对场点进行了优化处理，最后使用 Hermite 插值法连接场点实现数值场可视化。

## 2.2 隐式曲线

熟知的经典的曲线表示方法通常有隐式曲线和参数曲线两种。其中参数曲线容易构造、计算方便，具有细分性等特点成为几何设计中的比较流行的表示方法之一。但是在同样次数的情况下，隐式曲线有更高的自由度，所以相比参数曲线可以达到更高的光滑度和提供更多的形状控制<sup>[34]</sup>。因此，隐式曲线描述形状非常有效，进而得到了广泛使用<sup>[35]</sup>。

### 2.2.1 隐函数及隐函数曲线

#### 1、隐函数数学定理

设  $F : R^n \times R^m \rightarrow R^n$  是连续映射，且  $F(x_0, y_0) = 0$ ，分别存在  $x_0$  和  $y_0$  的开邻域  $A_0 \subset R^n$  和  $B_0 \subset R^m$ ，使得对任意的  $y \in B_0$ ，等式  $F(x, y) = 0$  有唯一解  $x = H(y)$ ，其中  $H$  是从  $B_0$  到  $A_0$  的连续映射，充分必要条件是分别存在  $x_0$  和  $y_0$  的开邻域  $A \subset R^n$  和  $B \subset R^m$ ，使得对任意的  $y \in B$ ， $f(\cdot, y) : A \rightarrow R^n$  是局部一一映射<sup>[36]</sup>。

#### 2、隐函数概念

我们知道平面曲线可以用参数形式表示为： $x = x(t)$ ,  $y = y(t)$ ，即可把曲线上的点的坐标  $(x, y)$  表示为参数  $t$  的函数。同时还可以将其表示为  $F(x, y) = 0$  的隐式曲线方程的形式。隐式曲线对应的隐函数是一种相对于显函数的一种函数形式，其因变量和自变量之间的对应法则是由一个方程式来表示<sup>[37]</sup>。

一般的，如果变量  $x, y$  满足一个方程  $F(x, y) = 0$ ，在一定条件下，当  $x$  取某一区间任意值时，相应地总有满足这一方程的唯一  $y$  值存在，那么就说方程  $F(x, y) = 0$  在该区间内确定了一个隐函数<sup>[38]</sup>。

函数一定是方程，而方程不一定是函数，隐函数不一定是函数，而是方程。。例如定义函数  $F(x, y) = x^2 + y^2$ ，那么  $x^2 + y^2 = 1$  既是方程也属于隐函数。

方程（隐函数） $x^2 + y^2 = 1$  的所有解的集合构成单位圆。然而圆上的点无法用统一的  $y = g(x)$  的形式表示的，每个  $x \in (-1, 1)$  都有两个  $y = \pm\sqrt{1 - x^2}$  的值与之对应。当然可以局部地用  $x$  来表示  $y$ ，如给定圆上一点  $(x, y)$ ，若  $y > 0$ ，该点则圆的上半部分，那么在这一点附近  $y$  可以写成关于  $x$  的函数： $y = \sqrt{1 - x^2}$ ；若  $y < 0$ ， $y$  可以写成关于  $x$  的函数： $y = -\sqrt{1 - x^2}$ 。因此，相较于此，用隐函数表示曲线更为方便<sup>[38]</sup>。

### 3、隐函数曲线

平面曲线既可以用参数形式  $\{x(t), y(t) : t \in R\}$  表示，也可将其表示为  $F(x, y) = 0$  的隐式曲线方程的形式。目前关于参数曲线的绘制技术相对成熟，而隐式曲线的绘制却并不那么简单<sup>[17]</sup>。根据相关文献显示，隐式曲线绘制方法大致分为连续跟踪法和细分法，这两种方法已经在第一章中的国内外研究现状进行了回顾，此处不再赘述。

由于参数形式比隐式形式更适合计算曲线上的点列，同时它在定义与变换自由形式的曲线几何造型中提供了极大的便利和灵活性因此成为形状数学描述的标准形式，大量实际的图像系统均是使用参数形式设计进行曲线造型<sup>[37]</sup>。

然而参数形式又存在不易判定给定点是否在曲线上以及给定点与曲线的相对位置等局限性，因此近些年来，隐式形式曲线越来越受到重视，相比于参数形式的曲线，隐式形式的曲线虽然在绘制上存在困难，但它在具有以下不可替代的优点：

（1）从理论上讲，隐式曲线能保持参数形式的所有性质，并比参数形式有更多的自由度。与参数曲线相比，相同次数的隐式形式曲线可达到更高的光滑度或提供更多的形状控制。

（2）隐式形式表示不仅有紧凑简洁的表达形式，而且提供了足够灵活性使其可以拟合任何复杂形状的曲线。同时，通过参数的消去，参数形式是隐式形式的特例。

## 2.2.2 隐性数值场

数值场可视化技术有多种，相关内容已阐述于 2.1.2 节中。在本文中即采用隐函数形式来表示数值场，我们称之为隐性数值场隐式形式表示的数值场紧凑简洁，提供了足够的灵活性使其可以拟合复杂形状的曲线。在隐性数值场中，隐式曲线即场线，即在已知数值场隐函数模型的情况下，采用等值线/等值面实现数值场可视化。

## 2.3 Hermite 插值

Hermite 插值是一种具有保形特征的插值函数，其不仅能够保证插值点处函数值与真实函数一致，而且能够实现插值点处一阶导数乃至更高阶导数与真实函数相同，这便使得 Hermite 插值更加易于逼近真实函数。本文即采用三阶 Hermite 插值多项式，即能同时保证插值函数在插值点处的函数值和一阶导数与原函数一致。

[39]

### 2.3.1 Hermite 插值定义

Hermite 插值是利用未知函数  $f(x)$  在插值节点上的函数值及导数值来构造插值多项式。文献[40]中 Hermite 插值有如下定义：给定节点处的函数值与一阶导数值

$$H(x_i) = y_i, \quad H'(x_i) = y'_i \quad (i = 0, 1, \dots, n) \quad (2.1)$$

从上式中可看出满足插值条件的单一插值多项式的最低次数为  $2n + 1$  次，且可唯一表示成

$$H(x) = \sum_i y_i A_i(x) + \sum_i y'_i B_i(x) \quad (2.2)$$

其中

$$A_i(x) = \left[ 1 - \frac{w^n(x_i)}{w(x)} (x - x_i) \right] L_i^2(x) \quad (2.3)$$

$$B_i(x) = (x - x_i) L_i^2(x) \quad (2.4)$$

$$L_i^2(x) = \frac{w(x)}{\dot{w}'(x)(x - x_i)} \quad (2.5)$$

$$w(x) = \prod_{i=0}^n (x - x_i) \quad (2.6)$$

高次 Hermite 插值多项式在稳定性、收敛性等方面存在着本质困难。因此，在实际应用中常用到分段低次的 Hermite 插值。下面给出分段三次的 Hermite 插值多项式的推导过程。

### 2.3.2 Hermite 插值多项式及插值方法

分段三次 Hermite 插值是函数拟合和插值的基本方法[41]。对于给定  $n + 1$  个插值点  $x_0 < x_1 < \dots < x_n$  和这些插值点上关于函数  $f(x)$  的函数值  $y_0, y_1, y_2, \dots, y_n$  及导数值  $m_0, m_1, \dots, m_n$ ，对任意  $i = 0, 1, 2, \dots, n$ ，在区间  $[x_i, x_{i+1}]$  构造次数不大于 3 的多项式的分段插值，且有  $H(x_i) = y_i$  和  $H'(x_i) = m_i$ ，那么函

数  $H(x)$  即为插值点  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$  的分段三次 Hermite 插值函数。

若已知两点  $P_0$ 、 $P_1$  及其切向矢量  $P'_0$ 、 $P'_1$ ，即可根据两点构成的三次 Hermite 参数方程求出插值点进而描述点  $P_0$ 、 $P_1$  间的这段曲线。

为推导出三次 Hermite 参数方程<sup>[42]</sup>，首先假设平面上一条三次参数曲线表达式为

$$\begin{cases} x = a_x + b_x \cdot t + c_x \cdot t^2 + d_x \cdot t^3 \\ y = a_y + b_y \cdot t + c_y \cdot t^2 + d_y \cdot t^3 \end{cases} \quad (2.7)$$

根据式 (2.7)，设该曲线的矢量表达式为

$$\bar{P}(t) = \bar{a} + \bar{b} \cdot t + \bar{c} \cdot t^2 + \bar{d} \cdot t^3 \quad (2.8)$$

其一阶求导式为

$$\bar{P}'(t) = \bar{b} + 2 \cdot \bar{c} \cdot t + 3 \cdot \bar{d} \cdot t^2 \quad (2.9)$$

其中  $0 \leq t \leq 1$ ， $\bar{P}$ 、 $\bar{a}$ 、 $\bar{b}$ 、 $\bar{c}$ 、 $\bar{d}$  均为矢量，根据几何定义可知，当  $t = 0$  时，曲线表达式值即为端点  $P_0$ ；当  $t = 1$ ，曲线表达式值即为另一端点  $P_1$ 。然后结合式 (2.8) 和式 (2.9) 即可求得  $P_0$ 、 $P_1$ 、 $P'_0$ 、 $P'_1$  的关系式，从而解得

$$\begin{cases} \bar{a} = P_0 \\ \bar{b} = P_0 \\ \bar{c} = 3(P_1 - P_0) - 2P'_0 - P'_1 \\ \bar{d} = 2(P_0 - P_1) + P'_0 + P'_1 \end{cases} \quad (2.10)$$

最后将式 (2.10) 中的各系数值带入式 (2.8) 中即可得三次 Hermite 插值参数函数一般式

$$P(t) = P_0(2t^3 - 3t^2 + 1) + P_1(3t^2 - 2t^3) + P'_0(t^3 - 2t^2 + t) + P'_1(t^3 - t^2) \quad (2.11)$$

本文中在实现数值场绘制过程中即采用三次 Hermite 插值参数函数即式 (2.11) 计算采样点间的插值点进而数值场中的场线。

## 第 3 章 场线绘制方法研究

本章“场线绘制”是本文的核心内容，也是数值场绘制的重要内容，主要详细介绍了二维 MC 方法获取场点及如何消除场点二义性、面积法压缩场点过程、Hermite 插值连接场点等内容。

### 3.1 数值场数学模型定义与获取

获得数值场的数学模型是场线绘制的最基础工作。本文主要以物理中的电势场数值场为示例对相应内容进行描述。

在物理中电势是标量，它是描述静电场特性的基本物理量之一。库伦定律指出：两个静止点电荷之间的相互作用力是向心力，其方向沿两者的连线其大小只依赖于两者的距离。静电场中的环路定理即若实验电荷在静电场中沿闭合路径移动一周，则静电力对它所做的功为零。环路定理表明电场是保守场或势场，存在着一个可以用来描述静电场特性的、只与位置有关的标量函数——电势。

静电场中某一位置处的电势公式为：

$$V = \frac{q}{r} \quad (3.1)$$

其中  $V$  代表电势值， $q$  为点电荷带电量， $r$  则代表两点电荷之间的距离。

例如在模拟位于  $(a, b)$  的单个点电荷  $q$  在  $x - y$  面内的电势场（数值场）时，电势为  $V_0$  的等势线（场线）方程，即相应数学模型可以写为：

$$V(x, y) = \frac{q}{r} = \frac{q}{\left[ (x - a)^2 + (y - b)^2 \right]^{\frac{1}{2}}} = V_0 \quad (3.2)$$

此电势线（场线）即为圆心在  $(a, b)$ ，半径为  $\frac{q}{V_0}$  的圆。

相应两个点电荷相应的数学模型也可得出，设两个点电荷的位置分别为  $(x_1, 0)$  和  $(x_2, 0)$ ，其带电量分别为  $q_1$  和  $q_2$ ，则这两个点电荷在  $x - y$  平面内的电势场（数值场）方程满足：

$$V(x, y) = \frac{q}{r_1} + \frac{q}{r_2} = V_0 \quad (3.3)$$

其中  $r_1 = \left[ (x - x_1)^2 + y^2 \right]^{\frac{1}{2}}$ ， $r_2 = \left[ (x - x_2)^2 + y^2 \right]^{\frac{1}{2}}$ 。为简单起见，在公式 3.2 和公式 3.3 中已省略了系数  $\frac{1}{4} \pi \epsilon_0$ ，该系数只影响场线的大小，而不影响其形状，因而作图时可不予考虑。

同理，依次类推 3 个点电荷乃至更多点电荷的电势场（数值场）可按上述方式推导出其数学模型。



## 3.2 二维 MC 方法与场点捕捉

二维 MC 方法获取场点是以三维 Marching Cubes 等值面构造方法为基础的，通过网格离散化后获取数值场场点。下面首先介绍 Marching Cubes 算法。

### 3.2.1 MC 方法

W.E.Lorenson 和 H.E.Cline 在 1987 年提出了一种典型的基于体元绘制面的 Marching Cubes 方法（下文中将其称为 MC）<sup>[43]</sup>。MC 算法被认为是目前最流行的等值面抽取算法之一，对科学计算可视化领域产生了深远的影响<sup>[44]</sup>。MC 方法是一种经典的基于体元构造等值面的算法，它最小的等值面搜索单元是数据场中的相邻两层上的 4 个像素顶点组成的立方体（也被称为体元）。该算法的主要思想是逐个处理数据场的体元，查找出等值面相交的体元，再通过线性插值计算出等值面与体元各边相交的交点，根据体元的顶点与等值面的相对位置，将各个交点按照一定方式连接起来作为体元内的等值面逼近。所有体元中的等值面即构成整个数据场的等值面。等值面的最后绘制过程还需知道等值面中各三角片顶点处的法向量，一般先采用中心差分计算出各体元顶点处的梯度，再用线性插值计算等值点处的梯度值作为该点的法向量<sup>[45]</sup>。

MC 算法的基础即确定体元内部的等值面的分布。不难发现该算法的基本假设是沿着体元的各边的数据场呈线性变化，也就是说，如果一条边的两个顶点的数据场值分别大于和小于等值面的值，则该条边上有且仅有一点是等值面与该边的交点。MC 算法可简单的总结为以下三个主要步骤：

- （1）通过线性插值计算出体元各边与等值面的交点；
- （2）将交点按照一定方式连接成等值面；
- （3）线性插值计算等值面中各等值点处的法向量，为后续绘制提供依据。

该算法中在计算体元与各等值面的交点中，首先对立方体体元的 8 个顶点进行分类以便于判断顶点与等值面的位置关系，为确定体元内等值面的连接方式提供依据。假设等值面值为  $C_0$ ，体元顶点可分为以下两类：

- （1）若  $C_0$  大于体元顶点的数据场值，则定义该点位于等值面之外，标记为“+”；
- （2）若  $C_0$  小于体元顶点的数据场值，则定义该点位于等值面之内，标记为“-”。

因为一个立方体体元有 8 个顶点，每个顶点可能存在两种状态，所以这个体元就有 256 中组合状态。但是由于立方体的旋转对称性，256 种组合状态中有重复的情况，Lorenson 将所有状态情形总结为了以下 14 种基本模式<sup>[46]</sup>

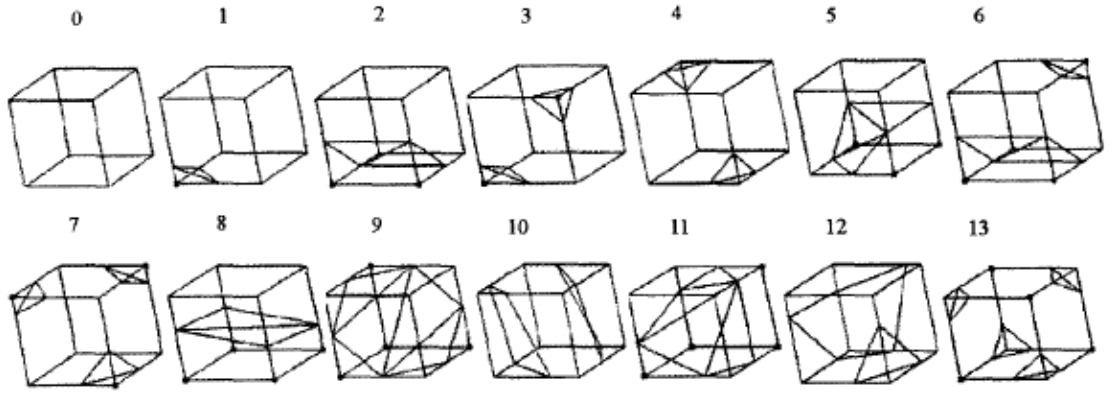


图 3.1 MC 算法中 14 种基本体元模式

MC 方法的提出提供了一种精确的定义体元以及体元内等值面的生成方法，定义体元为由相邻层之间八个网格点组成的数据单元，它提供了一种更一般化的体元定义，从造型的观点看，所有三维标量场都可以由这种一般意义下的体元或单元组成，因而标量场中的等值面的生成均可以在这些组成标量场的体元内完成，这位散乱数据场和曲面分布数据场中等值面的构造提供了非常简单有效的方法<sup>[45]</sup>。但算法在构造等值面的过程中，太依赖与直观的构造，构建体元状态模型的过程中，对于对称、旋转等情况的处理缺乏全面的考虑，忽略了立方体内部可能存在的换装结构和存在的临界，直接使用求得的边界等值点，根据简单的体元状态模型，简单连接成三角片，因此存在许多缺点，例如面二义性<sup>[47]</sup>和体二义性<sup>[48]</sup>。其中 Dürst<sup>[47]</sup>最早发现了 MC 算法的二义性问题，如果体元的一个边界面上标号为+和标号为一一的顶点分别位于对角线的两端，则该表面上存在四个等值点，那么就会有两种可能的等值线的连接方式。如果二义性问题不解决，就会导致最后生成的等值面可能有“孔洞”存在，即不能保证曲面的封闭性，从而将空间分为“内”和“外”，我们称这种二义性为面二义性。MC 算法面世以来就受到了极大的关注，该领域研究人员在应用中不断对其加以改进。

MC 方法也可降维应用于二维等值线的绘制(称其为二维 MC 算法)，具体内容为将绘制区域网格离散化，并对每个网格单元顶点分类标记，然后对顶点标记互异的边通过插值法求出采样点，最后在对采样点不做任何处理的情况下将其连接起来，从而绘制出曲线。该方法虽然在场点连接时会出现二义性问题，但该问题容易解决，原理简单，容易实现。

### 3.2.2 基于二维 MC 方法的场点获取

本文在三维 MC 方法的基础之上提出了二维的获取场点的 MC 方法。其具体步骤为：

- (1) 首先设定绘制区域  $[(x_0, y_0), (x_1, y_1)]$  并构造  $N \times N$  的网格单元。在构造

网格时， $x$  方向和  $y$  方向均从点  $(x_0, y_0)$  分别按照  $\nabla x = \frac{(x_1 - x_0)}{N}$  和  $\nabla y = \frac{(y_1 - y_0)}{N}$  依次递增从而获得各网格顶点坐标  $(x, y)$ 。在已知各个网格顶点坐标  $(x, y)$  后，即可根据 3.1 节推导得到的数学模型  $F(x, y) = 0$  分别计算每个网格顶点  $N_i (i = 0, 1, 2, \dots, (n+1)^2)$  的函数值  $V_{N_i}$ ，再根据场线（等值线）的值  $C$  对网格顶点进行分类标记：若  $V_{N_i} \geq C$ ，则定义该网格顶点位于场线之外，标记为“+”；若  $V_{N_i} < C$ ，则定义该网格顶点位于场线之内，标记为“-”，当然若是  $V_{N_i} = C$  则该网格顶点正好位于场线之上。本文为考虑算法的简洁性，将  $V_{N_i} = C$  的顶点也标记为了“+”。直到网格中所有顶点

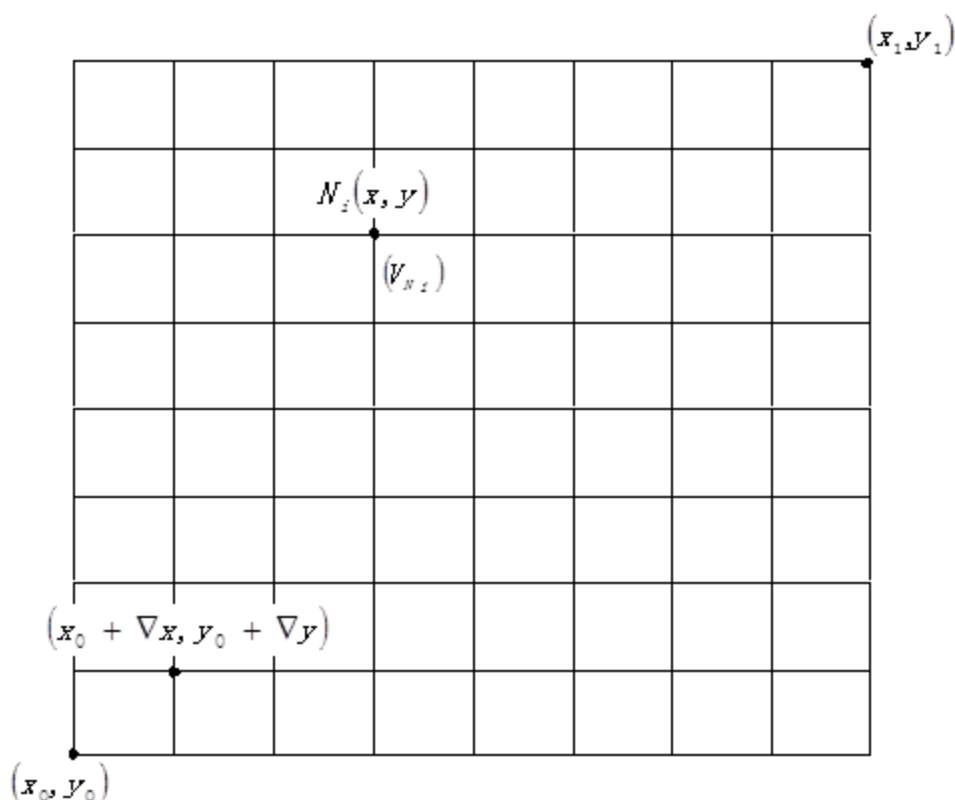


图 3.2 二维 MC 算法网格分布

(2) 线性插值获取网格单元边上的场点。在实现线性插值之前应该判断网格单元边上是否存在场点，其判断原则是根据上一步工作中的网格单元边上顶点标记，若一条边上的两顶点标记同为“+”或“-”，则该边上无场点；若一条边上的两顶点标记是“一正一负”，则该边上应存在场点。对存在场点的网格单元边计算线性插值的过程如下：设有如图 3.3 所示的由顶点  $N_0$ 、 $N_1$ 、 $N_2$ 、 $N_3$  构成的网格单元，相应顶点标记分别为“+”、“-”、“-”、“+”，可判断  $N_0N_1$ 、 $N_0N_2$ 、 $N_1N_3$ 、 $N_2N_3$  这 4 条边上均存在场点。以  $N_0N_2$  为例计算该边上的场点  $P_0(x', y')$ ，其中顶点  $N_0(x_{N_0}, y_{N_0})$  和  $N_2(x_{N_2}, y_{N_2})$  对应的函数值为  $V_{N_0}$ 、 $V_{N_2}$ ，首先计算场线（等值线）的值  $C$  与线段  $N_0N_2$  两顶点的比例关系，其比例为：

$$K_{N_0N_2} = \frac{C - V_{N_0}}{V_{N_2} - V_{N_0}} \quad (3.4)$$

则场点  $P_0$  的横纵坐标分别为:

$$x' = x_{N_0} + K_{N_0N_2} * (x_{N_2} - x_{N_0}) \quad (3.5)$$

$$y' = y_{N_0} + K_{N_0N_2} * (y_{N_2} - y_{N_0}) \quad (3.6)$$

依次类推即可求得各个网格单元边的各个场点。

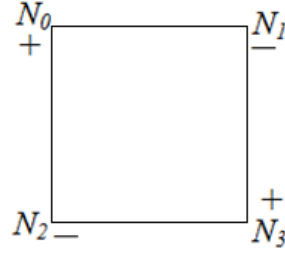


图 3.2 网格单元示例

如上步骤即为二维 MC 方法获取场点的完整过程，与三维 MC 方法一样，二维 MC 方法在实现过程中也存在二义性问题，下一节即详细阐述如何消除二义性。

### 3.2.3 消除二维 MC 方法的二义性

二维 MC 方法获取的场点在连接时部分场点存在一定的二义性，比如图 3.4 中所示，在某一网格单元四条边上分别有  $A$ 、 $B$ 、 $C$ 、 $D$  四个场点，那么这四个场点的连接方式将有图 3.4 中 (a)、(b) 两种情况。

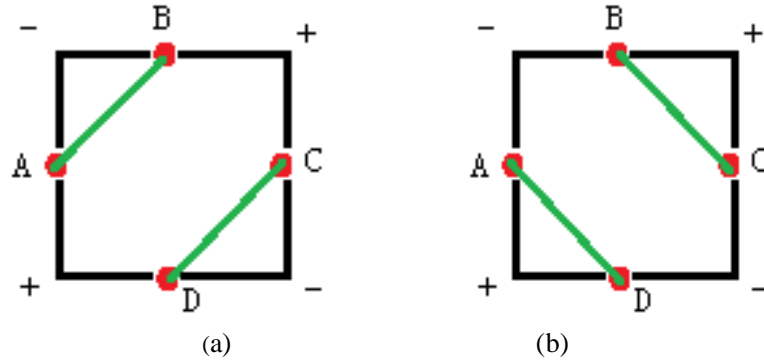


图 3.4 二义性情况

为解决本算法中的二义性问题，首先计算出每个网格单元中心点  $K_i$  ( $i = 0, 1, 2, \dots, N^2$ ) 坐标、函数值  $V_{K_i}$  并判断其标记，进而在每个网格单元中构造 4 个三角形如图 3.5 所示。查找场点的方式由之前的以正方形网格单元为单位转向了由三角形为单位，其计算方式不变。以  $\Delta v_0 v_1 v_2$  为例，场点的情况将会有图 3.6(a)-(d) 四种情况，其中 (a) 图三角形三个顶点同号，该种情况下三角形各边上无场点，(b) 和 (c) 两图中均属于三角形边上存在两个场点的情况，(d) 图则属于三角形边上存在 3 个场点的情况。如此便可避开图 3.4 中可能出现的二义性。

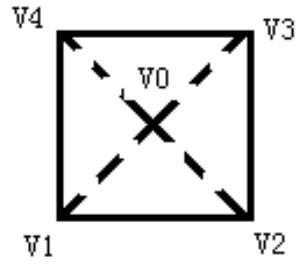


图 3.5 网格单元中构造 4 个三角形

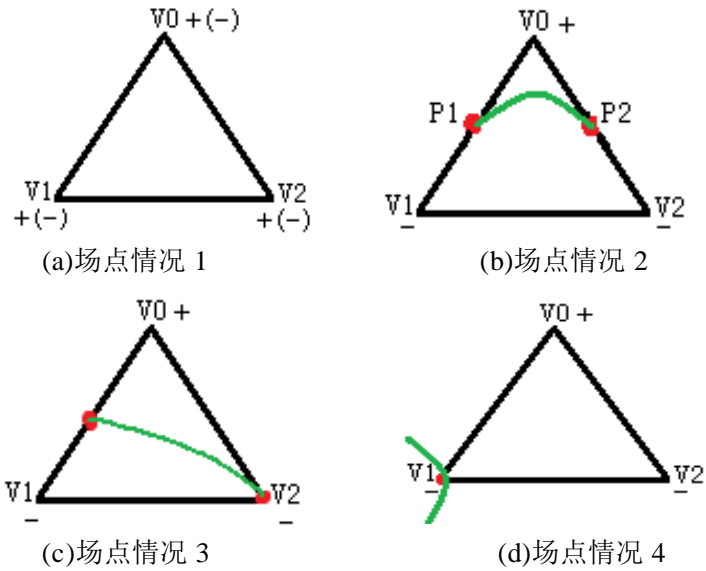


图 3.6 网格单元及采样情况

### 3.3 基于面积法的场点压缩

为减少数值场中的数据量，加快场线绘制速度，本算法对场点进行了优化。基于已有研究，本文提出了一种面积误差法优化压缩场点。场点能否被压缩取决于压缩前后对场线产生的影响。因此本算法中判断场点能否被压缩的要求是：场点压缩前后构成的曲线之间产生的误差大小。如图 3.7 所示，设有待压缩场点 N、D、B、A、M 5 点连接而成的原始折线段 Line，由 Hermit 插值法计算出 N、M 之间的插值点 N、a、b、M 构成的折线段来逼近 Line。不难发现场点压缩前后产生的误差可由图中阴影部分面积 S 表示。假设允许产生的误差不能超过  $\mu$ ，若  $S \leq \mu$ ，满足优化要求，即可精简 D、B、A 三点，反之则不能精简。那么计算出阴影面积 S 即成为场点优化的关键。本算法中将计算阴影面积 S 分为两步：求两条折线段的交点和计算多边形面积。

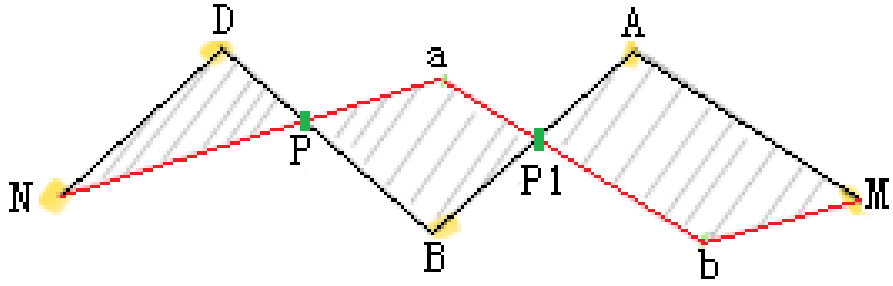


图 3.7 误差示例

### 3.3.1 线段求交算法

求解两线段交点的计算方法以图 3.7 中 DB 与 Na 两折线段为例，设其交点为 P。设线段 DB、Na 的矢量方程分别为

$$\vec{P} = \vec{D} + (\vec{B} - \vec{D}) \cdot t_0 (t_0 \in (0,1)) \quad (3.7)$$

$$\vec{P} = \vec{N} + (\vec{a} - \vec{N}) \cdot t_1 (t_1 \in (0,1)) \quad (3.8)$$

其中  $t_0, t_1$  为定比参数。假设两折线段有交点，联立式(3.7)、式(3.8)可得

$$\vec{D} + (\vec{B} - \vec{D}) \cdot t_0 = \vec{N} + (\vec{a} - \vec{N}) \cdot t_1 \quad (3.9)$$

求解之即可得  $t_0, t_1$  有无穷解，则两线段关系为平行或重合；若  $t_0, t_1$  有唯一解，

且  $t_0, t_1$  的值均在  $(0,1)$ ，即可判定两线段有且只有一个交点，将其代入式(3.7)或式(3.8)可求出交点 P。

### 3.3.2 平面复杂多边形的面积计算

多边形面积的计算方法以图 3.7 中的多边形 NDP 面积为例。其计算方法为在所求多边形平面内任取一点 O，如图 3.8 所示，通过三角形面积向量公式计算出  $\triangle NOP$ 、 $\triangle DOP$ 、 $\triangle DON$  的面积，其面积之和的绝对值即为  $\triangle NDP$  面积，即  $S_{NDP} = |S_{NOP} + S_{DOP} + S_{DON}|$ 。按照此方法依次类推，计算出图 3.7 中所有多边形面积之和为  $S = S_{NDP} + S_{PaP1B} + S_{P1AMb}$ ，S 即为折线段 Line 精简前后的误差。判断 S 与  $\mu$  的大小关系，即可判断场点能否优化。

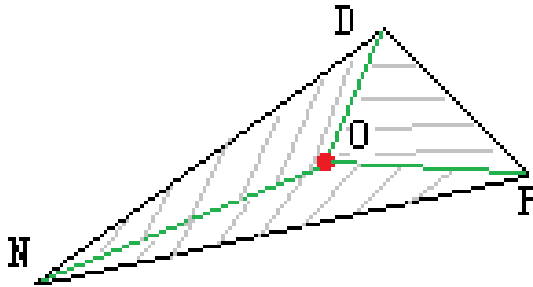


图 3.8 多边形 NDP

对于优化后的场点集，分别计算出各点切向，通过三次 Hermite 插值函数(如式(2.11)所示)按点序分段插值逼近隐式曲线。其中插值点数量根据实际曲线光滑

度要求进行设置。

### 3.4 场线提取与绘制

为方便说明场线的绘制效果，本小结以如下隐函数数学模型为例绘制出相应场线：

$$x^2 + y^2 - 0.5 = 0 \quad (3.10)$$

$$x^6 + 10y^6 - 0.5 = 0 \quad (3.11)$$

$$x^2 + y^2 + 2x - 2(x^2 + y^2)^{\frac{1}{2}} = 0 \quad (3.12)$$

$$8x + 9x + 19xy + 18y^2 + 13x^2 + 20xy^2 + x^2y + 19x^2y^2 - 362 = 0 \quad (3.13)$$

$$5(x^2 + y^2)^{\frac{1}{2}} - 9y^2(x^2 + y^2)^{-\frac{1}{2}} + 24y^4(x^2 + y^2)^{-\frac{3}{2}} - 16y^6(x^2 + y^2)^{-\frac{5}{2}} - x^2 - y^2 - 1.5 = 0 \quad (3.14)$$

上面 5 个数学模型图像对应为圆、圆角矩形、心形曲线、四角形曲线、和花型曲线,利用 3.3 节中的场线绘制方法绘制出该 5 个隐函数的在 30\*30 的网格中的图像如图 3.9 所示。从图中可以看到曲线较为光滑。具体场点优化前后的对比效果将在第四章中进行详述。

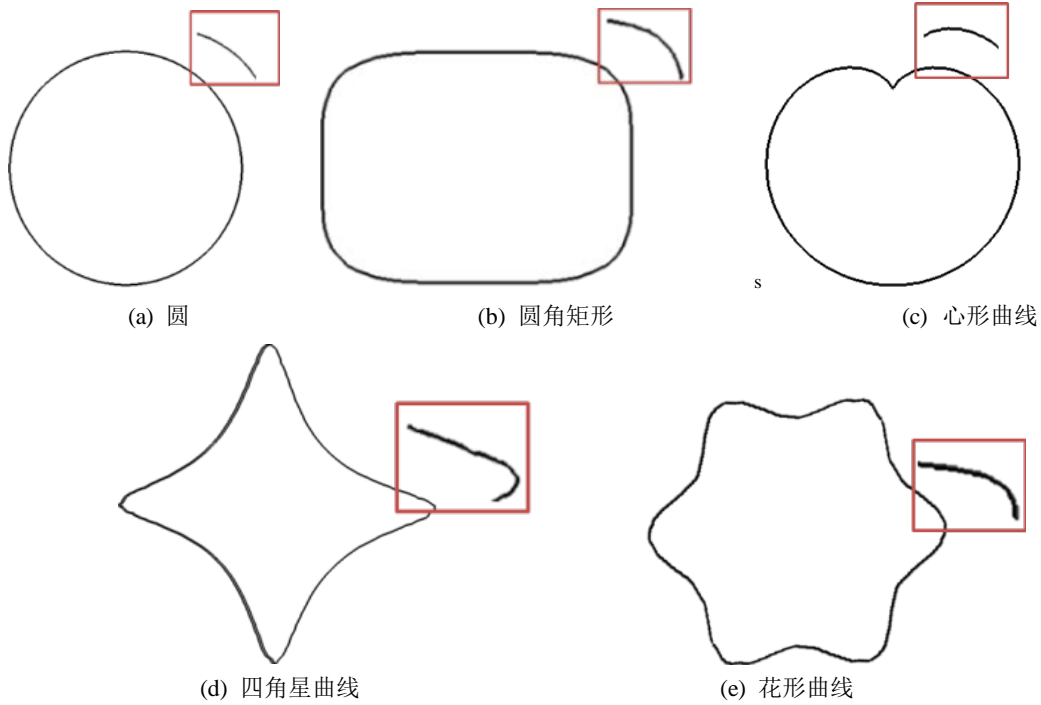


图 3.9 各隐式曲线绘制情况

### 3.5 本章小结

本章主要对数值场绘制中最重要的步骤——场线绘制做了分析和介绍。在详细获取到的数值场数学模型的基础之上，通过二维 MC 算法获取场点，并利用误

---

差面积法优化上一步获得的场点，最后对场线的绘制结果进行了展示。



---

## 第 4 章 算法实现与实验结果

在第三章中主要介绍了场线绘制算法的设计。本章主要介绍了数值场场线绘制算法的实现，并对第三章中公式 3.10 到公式 3.14 这 5 个数学模型绘制的场线进行了优化前后的对比，并以 3.1 节中的电势场数学模型为基础绘制了部分相应场的实例效果图。在算法实现一节中主要介绍了整个算法的流程、算法的数据结构、场点优化等环节的实现。

### 4.1 实验环境平台

本文算法采用 C++ 和 OpenGL 在硬件配置为 Intel Core2 2.66 GHz CPU，4GB 内存的电脑上实现的。

其中 OpenGL (Open Graphics Library) 作为一个专业的图形程序接口，它广泛应用于 2D 和 3D 的图像绘制与交互，是一个开放的图形软件包。OpenGL 独立于视窗操作系统，可以在不同的平台如 Windows、Unix、Linux、MacOS 等之间转换，因此支持 OpenGL 的软件有很好的移植性。另外 OpenGL 与 Visual C++ 有紧密接口，可以准确方便的实现有关图形计算的算法。

OpenGL 具有以下 7 大主要功能：

(1) 建模：OpenGL 图形库除了提供基本的点、线、多边形等绘制函数外，还提供比较复杂的三维物体（如：球、锥、多面体等）、复杂曲线曲面等绘制函数。

(2) 变换：在 OpenGL 中主要包括了两种变换：基本变换和投影变换。其中基本变换有平移、旋转、缩放等，投影变换包括平行投影和透视投影。图形库中的变换方法有助于减少算法的运行时间、提高图形的显示速度。

(3) 颜色模式：OpenGL 提供两种颜色模式，即 RGBA 模式和颜色索引 (Color Index) 模式。

(4) 光照和材质设置：OpenGL 图形库提供 4 种光：环境光 (Ambient Light)、漫反射光 (Diffuse Light)、反射光 (Emitted Light) 和镜面光 (Specular Light)。在 OpenGL 绘制的二维/三维场景中，二维/三维物体最终反映到人眼的颜色是材质的红蓝绿分量的反射率与光的红蓝绿分量相乘后形成的颜色。

(5) 纹理映射 (Texture Mapping)：通过 OpenGL 图形库中的纹理映射功能能逼真的呈现出物体表面的细节。

(6) 位图显示和图像增强：为增强二维/三维场景中被仿物体的真实感和图形的现实效果，OpenGL 库还提供图形的基本拷贝、像素读取以及融合 (Blending)、反走样 (Antialiasing)、雾 (Fog) 等图像处理效果。

(7) 双缓存动画 (Double Buffering)：OpenGL 库提供的双缓存即包括前台缓

存和后台缓存。在后台生成动画，缓存计算场景，前台即显示后台缓存已计算好的画面。

OpenGL 作为一种过程性的图形接口，相应程序执行的主要过程为：①配置参数，以 OpenGL 中的几何图元（如点、线、面等）构建相应形状；②上一步输入的几何图元再通过不同的着色器进行计算，进而判断其相应的位置和颜色等信息；③此步骤为光栅化，即将输入图元像素的数学表达式变换成与屏幕对应位置的像素；④为确定像素片元最终的颜色和位置，此步骤需将光栅化阶段中构建的所有片元进行着色操作。

## 4.2 算法实现

本文基于 C++ 和 OpenGL 软件环境下实现了本算法，其中算法步骤包括数值场场点捕捉、压缩场点以及连接场点。本小节将从算法整体流程、算法数据结构组织以及算法相关关键代码三个方面对本算法进行阐述。

### 4.2.1 整体流程

本文提出的数值场场线绘制算法的整体流程如图 4.1 所示，相应步骤包括：①输入数值场隐函数模型，②通过对绘制区域离散化处理捕捉场点，③为了实现场点优化还对捕捉到的场点进行了压缩处理，④对于优化处理好的场点通过 Hermite 插值法逼近场线。

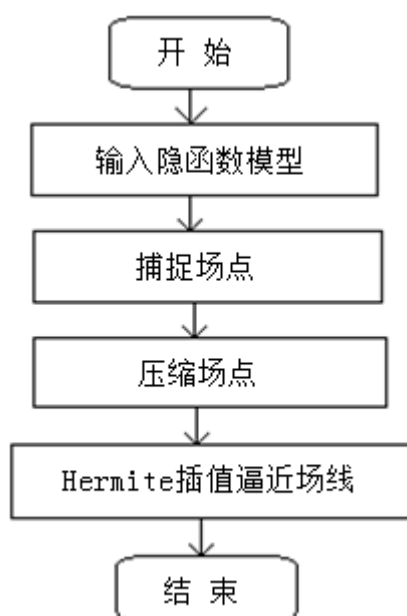


图 4.1 数值场场线绘制算法的整体流程

### 4.2.2 数据结构组织

数据结构的组织与设计是实现高效算法的重要一步。本文算法借鉴了网格法

和 MC(Marching cubes) 思想对绘制区域网格离散化获取场点；为加快场线绘制速度，减少场点存储空间，对场点数量进行了精简；绘制场线的过程则通过分段 Hermite 插值实现。

通过分析可发现算法中基础核心对象即点（网格顶点和场点），其中包括网格顶点和场点，在算法实现过程中用 GridData 表示网格顶点，PolyData 表示场点。网格顶点 GridData 数据结构中包括网格顶点的坐标（pos\_[0], pos\_[1]）和函数值 val\_，函数值 val\_ 即网个顶点坐标（pos\_[0], pos\_[1]）代入隐函数  $F(x, y) = 0$  中计算得到的。数值场场点 PolyData 数据结构中包括场点的位置坐标（pos\_[0], pos\_[1]）和切线方向（dir\_[0], dir\_[1]）。

```
struct GridData {           //格网中的顶点类型与单元类型
    Type pos_[2]; //位置坐标
    Type val_;    //函数值
};

struct PolyData {           //数值场场点
    Type pos_[2]; //场点的位置坐标
    Type dir_[2]; //场点的切线方向
};
```

为实现方便快捷优化场点，本算法采用双向链表存储数据场中的场点，相应双向链表中的结点（场点）Node 的数据结构为：

```
struct Node {               //双向链表中的场点（结点）结构
    PolyData field_piont; //双向链表中的场点
    int flag_; //结点标志
    struct Node* same_; //对偶顶点
    struct Node* poly_; //多边形指针
    struct Node* prev_; //前指针
    struct Node* next_; //后指针
Node(void) : flag_(0), same_(NULL), poly_(NULL), prev_(NULL), next_(NULL) {}
};
```

双向链表结点 Node 的数据结构中的 flag\_ 取值分别包括 0、1、2、3、4，其中 flag\_=0 表示场线中的中间点，flag\_=1 表示场线中起始点，fag\_=2 表示场线中的终止点，fag\_=3 或 fag\_=4 表示场线优化时场点压缩前后连接的两折线之间的交点。为了方便后期计算多边形面积，设计了 Node\* same\_ 指针指向同属于两个多边形的顶点，Node\* poly\_ 指针是指向多边形的指针，Node\* prev\_ 和 Node\* next\_ 即为双向链表的前后指针。

### 4.2.3 算法实现关键代码

本小节将呈现出第三节中提到的绘制场线关键步骤的部分代码，其中包括：提取二元隐函数数值场场线的 MC 离散格网、捕捉场点、计算误差面积以及逼近场线。

#### 1、捕捉场点

根据 MC 算法的思路，设计了提取二元隐函数数值场场线的 MC 离散格网，为了避免出现二义性的情况，算法中还提取了每个网格的中心点的坐标位置及其带入数值场隐函数模型中的函数值，为下一步获取场点做准备。其关键代码如下所示：

```
Grid(void) const {
    int i, j;去
    for (i = 0; i < size_; ++i) {
        for (j = 0; j < size_; ++j) {
            //v0、v1、v2、v3、v4 分别为网格单元的中心点和四个顶点，具体可参见图 3.5
            const GridData& v0 = cell_[i][j];
            const GridData& v1 = node_[i][j];
            const GridData& v2 = node_[i + 1][j];
            const GridData& v3 = node_[i][j + 1];
            const GridData& v4 = node_[i + 1][j + 1];
        }
    }
}
```

在获取到离散网格之后，即可获取场点。为避免二义性，本算法将以前从单个网格获取的方式改为将单个网格分解为四个三角形，分别从四个三角形中捕捉场点，相关原理已在 3.2.3 节中做出阐述。其关键代码如下：

```
_tri(const float _val, const GridData& _v0, const GridData& _v1, const GridData&
_v2) {
    int p[3][2]; //三角形中场线的端点
    int i = 0; //端点计数器
    float tmp; //比率变量
    //如果三角形的边与场线相交，则采用线性插值计算交点(端点)
    //此处仅以三角形中的一条边为实例，其余可依次类推
    if ((_v0.val_ >= _val && _v1.val_ <= _val) || (_v0.val_ <= _val &&
_v1.val_ >= _val)) {
```

---

```

        tmp = (_val - _v0.val_) / (_v1.val_ - _v0.val_);    //计算比率
        p[i][0] = _v0.pos_[0] + tmp*(_v1.pos_[0] - _v0.pos_[0]); //计算场线与此边
的交点
        p[i][1] = _v0.pos_[1] + tmp*(_v1.pos_[1] - _v0.pos_[1]);
        ++i; //交点计数加 1
    }
    .....
}

```

## 2、计算误差面积

本算法中较为关键的一部分即优化场点，其优化方法即比较场点优化前后产生的误差面积大小。具体原理已在 3.3 节中进行阐述，此处只呈现实现场点优化的关键代码。首先是计算优化场点前后构成的折线段之间的交点，部分关键代码如下：

```

bool intersect ( Polydata& _sect, Node* _seg0, Node* _seg1) {
    //求解交点的过程即为求解线性方程组解的过程
    Polydata    p; //存储交点 p 的结构体
    Mat2x2    a;    //系数矩阵 a
    Vec2f    x, b; // 矩阵 x, b
    a[0][0] = _seg0->next_->pn.pos_[0] - _seg0->pn.pos_[0];
    a[1][0] = _seg0->next_->pn.pos_[1] - _seg0->pn.pos_[1];
    a[0][1] = _seg1->pn.pos_[0] - _seg1->prev_->pn.pos_[0];
    a[1][1] = _seg1->pn.pos_[1] - _seg1->prev_->pn.pos_[1];
    b[0] = _seg1->pn.pos_[0] - _seg0->pn.pos_[0];
    b[1] = _seg1->pn.pos_[1] - _seg0->pn.pos_[1];
    if (a[0][0] * a[1][1] != a[0][1] * a[1][0]) {
        solve(a, b, x); //矩阵求解，解存放在 x 中
        if ((0< x[0] && x[0]<1) && (0<x[1] && x[1] < 1))    {
            p.pos_[0]=_seg0->pn.pos_[0]+(_seg0->next_->pn.pos_[0]-_seg0->pn.pos_[0])*x[0];
            p.pos_[1]=_seg0->pn.pos_[1] + (_seg0->next_->pn.pos_[1] - _seg0->pn.pos_[1])*x[0];
            _sect = p;
            return true;    }    }
    return false;
}

```

本算法中采用面积法计算压缩前后的误差，在计算出交点之后，对于压缩前

---

后的折线段“围出”区域，具体可参见图 3.7 和图 3.8，分别计算出区域中每个多边形面积，即可求出误差面积。部分关键代码如下所示：

```
float calcPoly(void)
{ Node *ptr0;    //新建遍历时要用的移动的临时节点
  ptr0 = head_; //从头节点开始遍历
  Node *temp;    //新建临时节点（保存多边形起始节点的指针）
  Node *p;
  float s_error = 0;
  do{ //开始查找多边形.
    temp = ptr0; //记下查找时该多边形的起始节点
    float s_poly = 0;
    do{
      float s1;
      Polydata P;
      P.pos_[0] = P.pos_[1] = 2;
      s1 = triangle(ptr0->pn, ptr0->next_->pn, P); //求解三角形面积
      ptr0 = ptr0->next_; //查找下一节点
      s_poly = s_poly + s1;
      if (ptr0->flag_ == 4) break;
      if ((ptr0->same_ != NULL) && (ptr0->flag_ = 3)){
        p = ptr0;
        ptr0 = ptr0->same_;
        assert(ptr0->flag_ = 4);
      }
      if (ptr0 == tail_) {
        s1 = triangle(ptr0->pn, ptr0->next_->pn, P);
        s_poly = s_poly + s1;
        ptr0 = tail_->next_;
        p = tail_;
      }
    } while (ptr0 != temp);
    s_error = s_error + fabs(s_poly);
    ptr0 = p;
  } while (p != tail_);
```

```

    return(s_error);
}

```

### 3、逼近场线

对于压缩后的场点，本算法采用 Hermite 插值法逼近场线，相关原理已在 2.3.2 节中进行了推导。相应关键代码如下：

```

void Hermit (PolyData _v0, PolyData _v1, int n){
    float f1, f2, f3, f4;
    double deltaT = 1.0 / n;
    glColor3f(1, 0, 1);
    glLineWidth(2);
    glBegin(GL_LINE_STRIP);
    for (int i = 0; i <= n; i++){
        double T = i*deltaT;
        f1 = 2.0*pow(T, 3) - 3.0*pow(T, 2) + 1.0;
        f2 = (-2.0)*pow(T, 3) + 3 * pow(T, 2);
        f3 = pow(T, 3) - 2.0*pow(T, 2) + T;
        f4 = pow(T, 3) - pow(T, 2);
        float x1, y1;
        x1 = f1*_v0.pos_[0] + f2*_v1.pos_[0] + f3*_v0.dir_[0] + f4*_v1.dir_[0];
        y1 = f1*_v0.pos_[1] + f2*_v1.pos_[1] + f3*_v0.dir_[1] + f4*_v1.dir_[1];
        glVertex2f(x1, y1);
    }
    glEnd();
    glFlush();
}

```

## 4.3 单条场线优化效果对比

数值场由多条场线组成，场线与场线之间有一定的相似性。比较单条场线绘制结果即可预知数值场绘制效果。因此在本小节中主要呈现第三章中隐函数(3.10) — (3.14) 这 5 个数学模型绘制的场线优化前后的对比结果。

为方便说明和比较优化结果，所有曲线均在  $30 \times 30$  的网格中获取场点进行绘制，其中各隐式曲线场点未压缩的绘制情况如图 4.2 所示。同时也分别将各曲线在允许误差  $\mu \leq 10^{-6}$ 、 $\mu \leq 10^{-5}$ 、 $\mu \leq 10^{-4}$ 、 $\mu \leq 10^{-3}$  各情况下场点优化后的绘制图像呈现于图 4.3、图 4.4、图 4.5、图 4.6。

各个曲线在不同允许误差  $\mu$  的情况下场点数量优化情况如表 4.1 所示。从表中可看到，允许误差  $\mu$  越大，优化率越大，优化后场点个数越少；反之允许误差  $\mu$  越小，其优化率越小，优化后场点个数越多。同时结合绘制结果可知同一场线随着允许误差  $\mu$  逐渐增大，曲线光滑度也在变化，其中  $\mu \leq 10^{-6}$  和  $\mu \leq 10^{-5}$  情况下，线光滑度较为乐观，在  $\mu \leq 10^{-3}$  情况下，曲线光滑度较低，例如图 4.7 所示的圆角矩形在不同允许误差情况下的绘制情况。

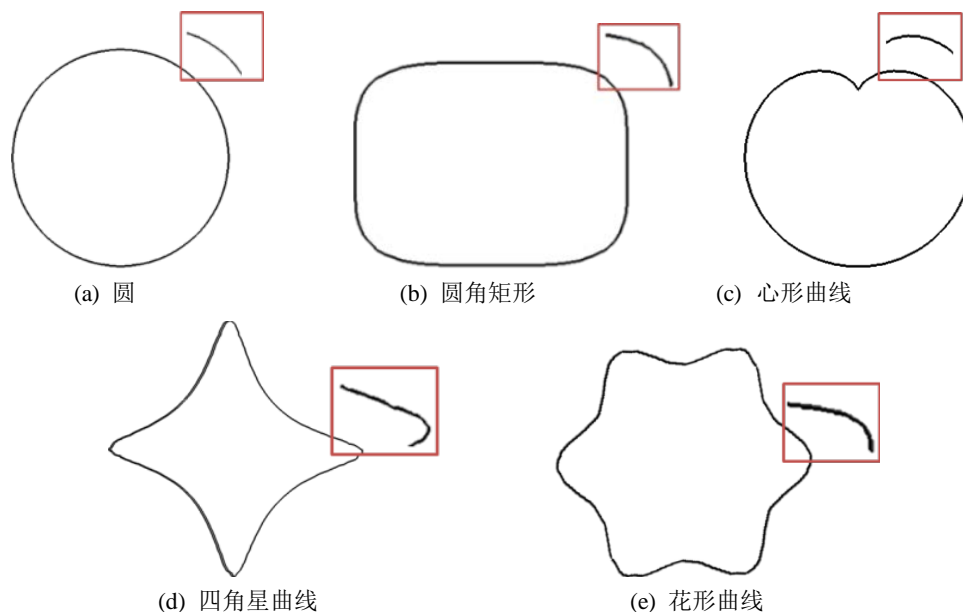


图 4.2 各隐式曲线场点未压缩绘制情况

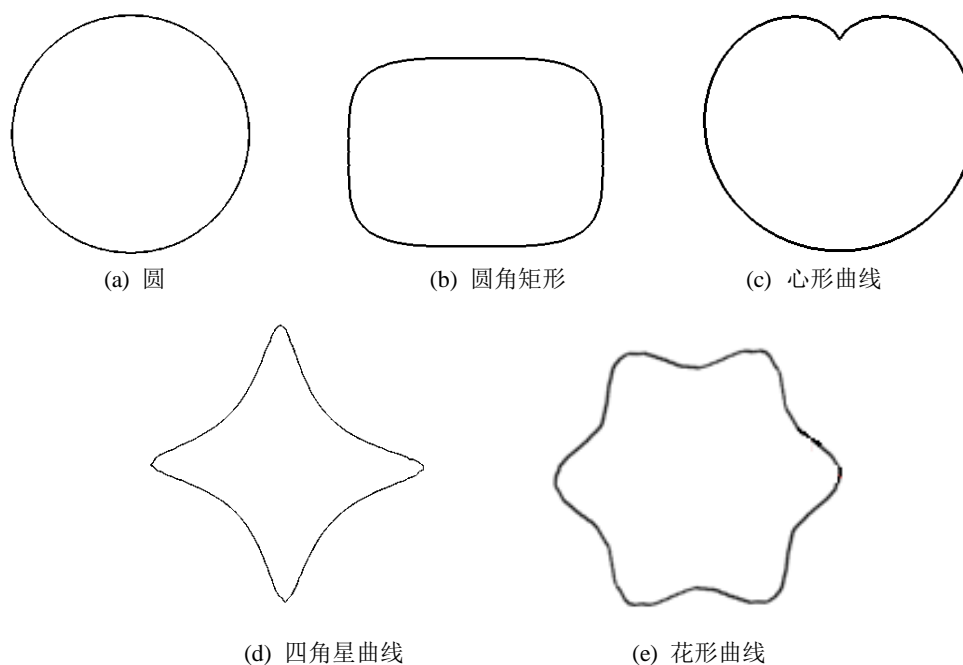


图 4.3 各隐式曲线在允许误差为  $\mu \leq 10^{-6}$  情况下的绘制情况



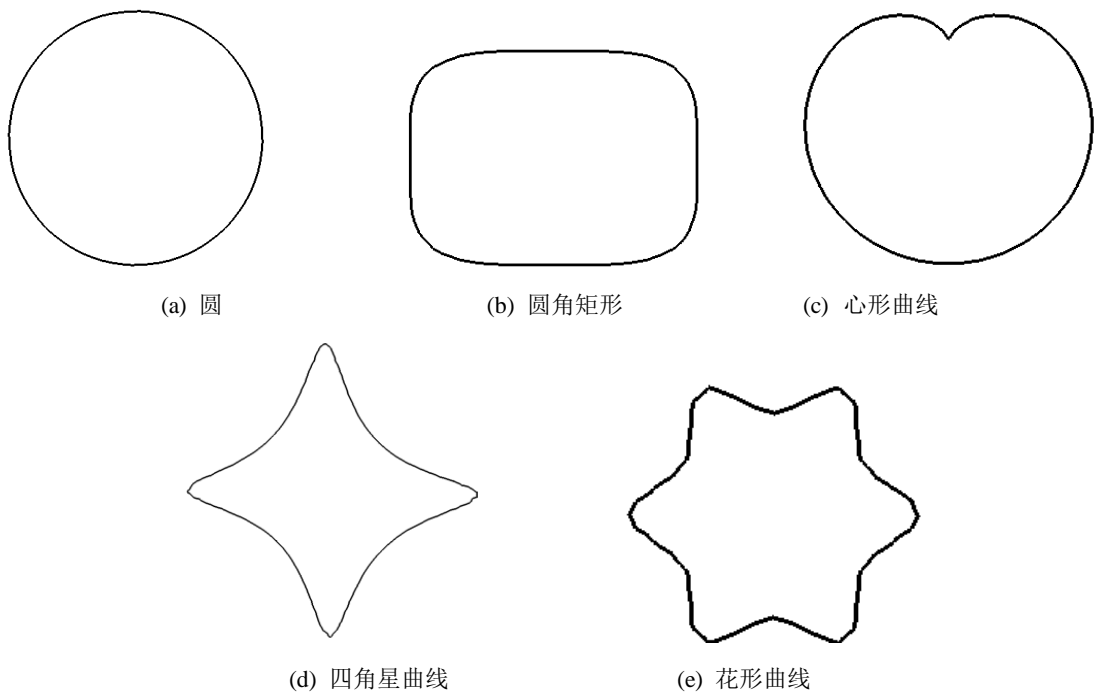


图 4.4 各隐式曲线在允许误差为  $\mu \leq 10^{-5}$  情况下的绘制情况

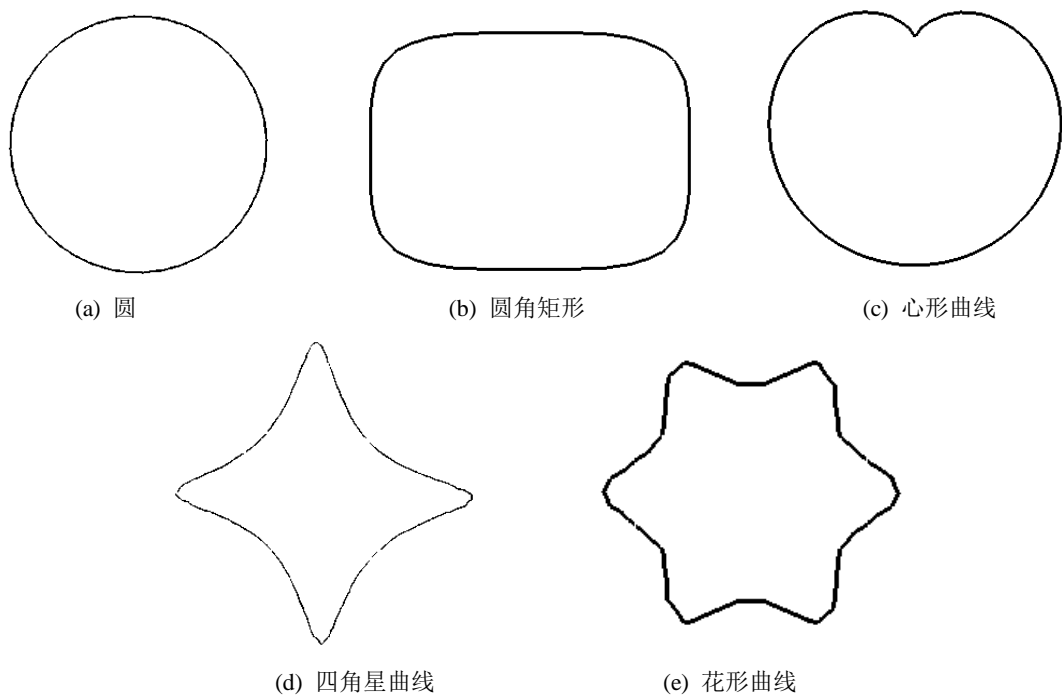
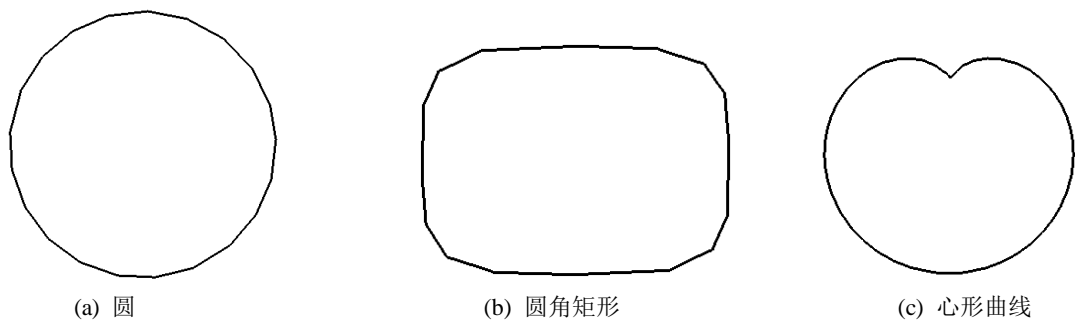


图 4.5 各隐式曲线在允许误差为  $\mu \leq 10^{-4}$  情况下的绘制情况



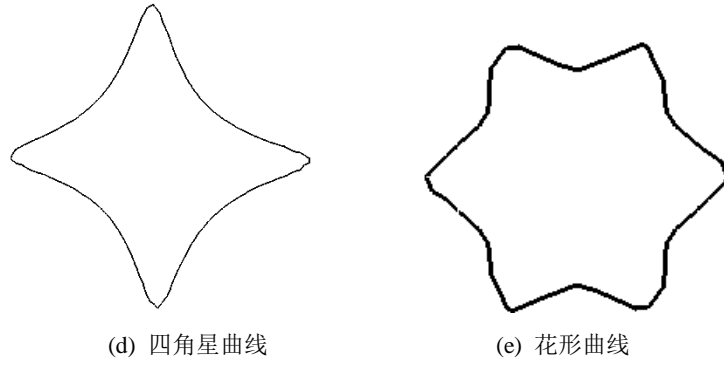


图 4.6 各隐式曲线在允许误差为  $\mu \leq 10^{-3}$  情况下的绘制情况

表 4.1 不同允许误差  $\mu$  情况下场点个数及压缩率比较

名称	优化前数据/点个数	$\mu \leq 10^{-6}$		$\mu \leq 10^{-5}$		$\mu \leq 10^{-4}$		$\mu \leq 10^{-3}$	
		A	B	A	B	A	B	A	B
式 3.10 (圆)	390	98	74.8%	81	79.3%	45	88.5%	22	94.3%
式 3.11(圆角矩形)	464	133	63.8%	72	80.4%	40	89.1%	18	95.1%
式 3.12(心形曲线)	376	174	59.6%	116	69.1%	79	79.0%	51	86.4%
式 3.13(四角星曲线)	496	238	52.0%	215	56.6%	149	70.0%	84	83.1%
式 3.14(花形曲线)	240	91	62.1%	53	77.9%	32	86.7%	20	91.2%

注:A 表示优化后场点个数, B 表示优化率.

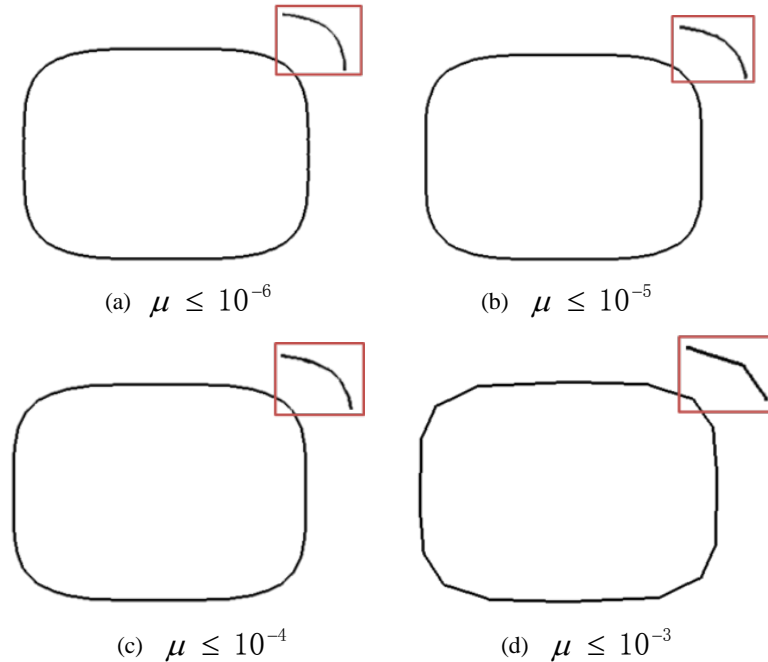


图 4.7 圆角矩形曲线在不同允许误差下的绘制情况

因本算法借鉴了 MC 算法思想获取场点, 将本算法与二维 MC 算法绘制的情况作对比有一定的可比性, 较能说明本算法中场点优化情况。通过实验可发现, 在同等网格数量  $30 \times 30$  条件下, 二维 MC 算法绘制的场线光滑度和准确度与本算

法允许误差  $\mu \leq 10^{-6}$  的情况下相似，对比这两种情况下（如图 4.1 和图 4.2 所示）绘制的场线的场点数量更有说服力。其对比情况如图 4.8，从图中可看出，本算法若要达到与 MC 算法同等的光滑度，对于较简单的曲线(如圆、圆角矩形等)本算法可通过其场点数量的1/4即可实现，对于特征较多的曲线(如心形曲线、四角星曲线)，本算法可通过其场点数量的1/2即可实现。

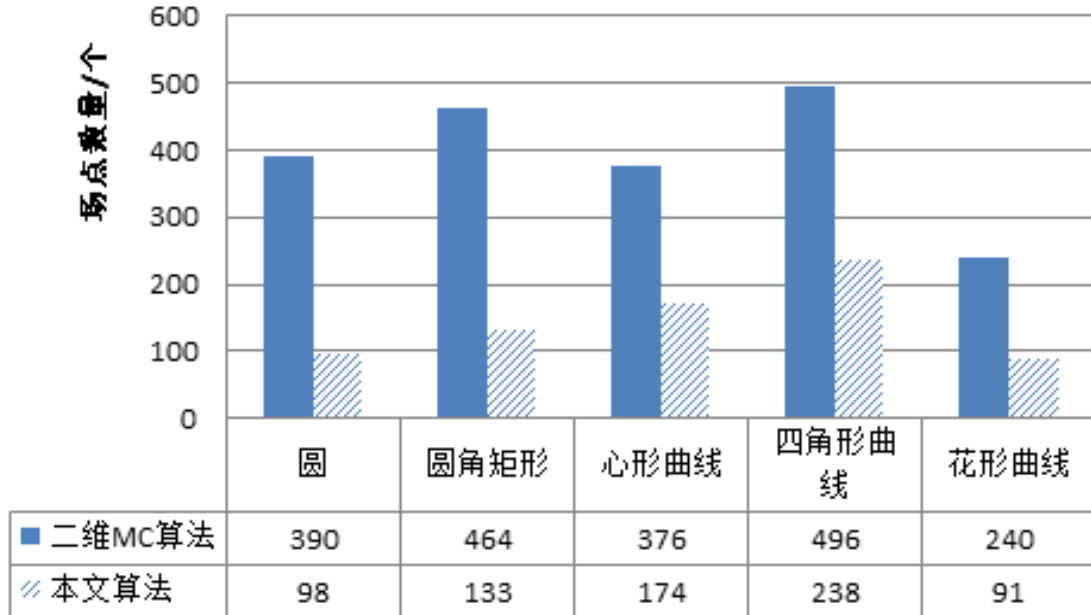


图 4.8 二维 MC 算法与本文算法允许误差为  $\mu \leq 10^{-6}$  情况下场点数量对比图

#### 4.4 数值场绘制效果

在前面工作的基础之上，为验证本算法绘制数值场的效果，此处分别绘制出了电势场的部分实例。本节中主要示例了静电场中部分点电荷绘制结果，相应算法中的允许误差均控制在  $\mu \leq 10^{-6}$ 。

首先运用本算法绘制 1 个  $1\mu C$  的电荷在 (0,0) 位置处的电势场（等势线），根据式 3.1 和式 3.2 可得其隐函数模型为：

$$V(x, y) = \frac{1}{\sqrt{x^2 + y^2}} \quad (4.1)$$

根据式 4.1 隐函数模型设置不同电势值  $V_0$ ，即可绘制出相应数值场，如图 4.9 所示。

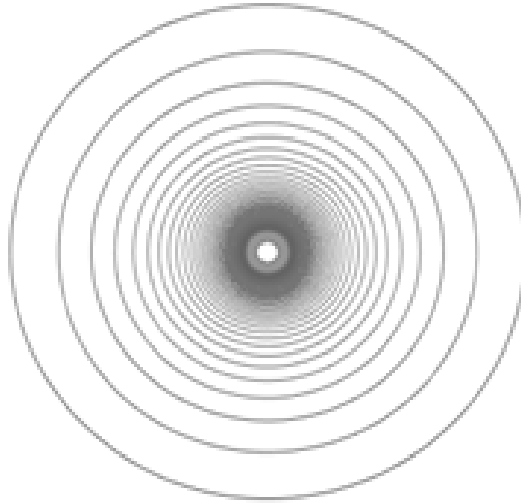


图 4.9 1 个  $1\mu C$  的电荷在  $(0,0)$  位置处的电势场

另外两个电荷值均为  $1\mu C$  的异性点电荷，其中电性为负的电荷所在位置为  $(1,1)$ ，电性为正的电荷所在位置为  $(-1,1)$ ，根据式 3.1 和式 3.2 可推导得其隐函数模型为：

$$V(x, y) = \frac{-1}{\sqrt{(x-1)^2 + (y-1)^2}} + \frac{1}{\sqrt{(x+1)^2 + (y-1)^2}} \quad (4.2)$$

根据式 4.2 隐函数模型即可绘制出相应数值场，如图 4.10 所示。

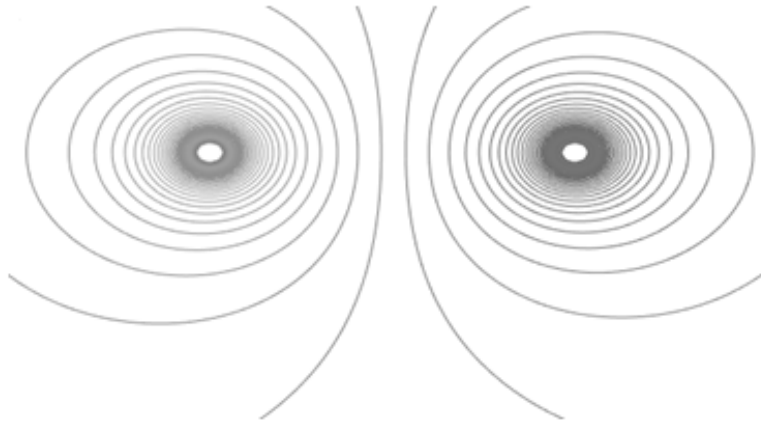


图 4.10 2 个电荷量均为  $1\mu C$  异性点电荷的电势场

两个电荷值分别为  $5\mu C$  和  $3\mu C$  的同性点电荷，其中电荷所在位置分别为  $(0,0)$  和  $(1,1)$ ，根据式 3.1 和式 3.2 可推导得其隐函数模型为：

$$V(x, y) = \frac{5}{\sqrt{x^2 + y^2}} + \frac{3}{\sqrt{(x-1)^2 + (y-1)^2}} \quad (4.3)$$

根据式 4.3 隐函数模型即可绘制出相应数值场，如图 4.11 所示。

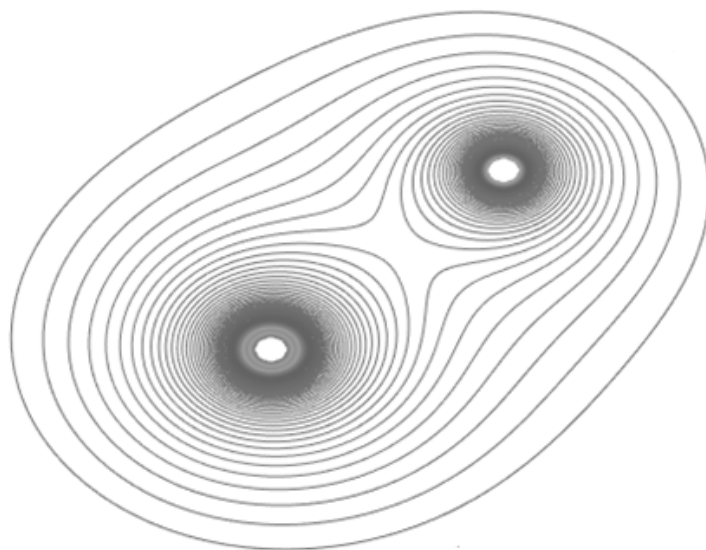


图 4.11 两个电荷数量分别为  $5 \mu C$  和  $3 \mu C$  的同性点电荷数值场绘制情况

同时还验证了三个异性点电荷的数值场绘制效果。三个点电荷数值大小分别为  $1 \mu C$ ， $-1 \mu C$ ， $2 \mu C$ ，位置分别为  $(0,0)$ ， $(1,1)$ ， $(-1,1)$ ，根据推导可得其数学模型为：

$$V(x, y) = \frac{1}{\sqrt{x^2 + y^2}} + \frac{-1}{\sqrt{(x-1)^2 + (y-1)^2}} + \frac{2}{\sqrt{(x+1)^2 + (y-1)^2}} \quad (4.4)$$

根据上述隐函数模型可绘制出相应数值场，如图 4.12 所示。

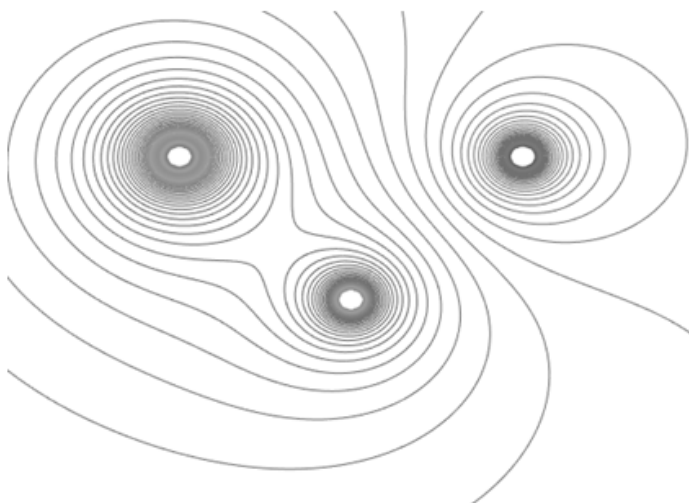


图 4.12 三个电荷数量分别为  $1 \mu C$ ， $-1 \mu C$ ， $2 \mu C$  的异性点电荷数值场绘制情况

最后还绘制了四个同性点电荷的数值场。四个点电荷数值大小分别为  $1 \mu C$ ， $2 \mu C$ ， $3 \mu C$ ， $4 \mu C$ ，位置分别为  $(0,0)$ ， $(1,1)$ ， $(-1,1)$ ， $(1.5,1.3)$ ，根据推导可得其数学模型为：

$$V(x, y) = \frac{1}{\sqrt{x^2 + y^2}} + \frac{2}{\sqrt{(x-1)^2 + (y-1)^2}} + \frac{3}{\sqrt{(x+1)^2 + (y-1)^2}} + \frac{4}{\sqrt{(x-1.5)^2 + (y-1.3)^2}} \quad (4.5)$$

根据上述隐函数模型可绘制出相应数值场，如图 4.13 所示。

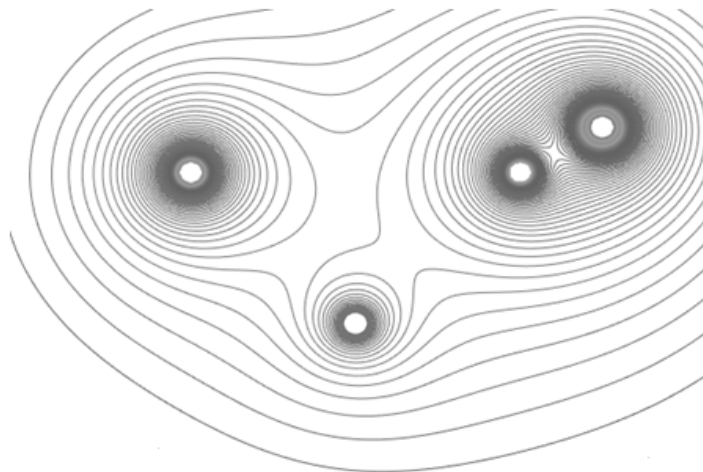


图 4.13 四个电荷数量分别为  $1 \mu C$ ， $2 \mu C$ ， $3 \mu C$ ， $4 \mu C$  的异性点电荷数值场绘制情况

通过本文提出的算法实现的电势场的模拟，相应绘制效果较好，场线光滑流畅准确，能满足一般教学的辅助。

## 4.5 本章小结

本章主要对本文所提出的算法进行了测试实验。算法实验平台选择的是 OpenGL 图形设备接口和 Visual C++。为显示本文算法的优化效果，我们将本算法与二维 MC 算法作比较，我们发现本算法若要达到与 MC 算法同等的光滑度，对于较简单的曲线(如圆、圆角矩形等)本算法可通过其场点数量的1/4即可实现，对于特征较多的曲线(如心形曲线、四角星曲线)，本算法可通过其场点数量的1/2即可实现。在此基础上，为验证数值场绘制效果，本文选择通过该算法实现静电场中点电荷的数值场（电势场），其绘制效果较好，场线光滑流畅准确，能满足一般教学的辅助。

## 第5章 基于本文算法的静电场点电荷可视化系统设计与实现

基于前面第3章和第4章对数值场绘制算法的设计与实现，在本章中将在该算法的基础之上，根据教学的需求设计并实现静电场中点电荷的可视化系统。本章中具体内容包括：静电场点电荷可视化系统的需求分析、界面设计以及系统的实现效果。

### 5.1 系统需求分析

需求分析是设计与实现系统的一项非常重要的工作，是系统开发的基础<sup>[1]</sup>。静电场点电荷可视化系统应用对象定位于教师或者学生，相应系统功能设计应当遵从实际教学需要。因此本节分析系统需求时将从用户需求和功能需求两方面出发进行阐述。

#### 5.1.1 用户需求

本系统面向对象为教师以及学生，相关功能等设计应遵从教学大纲，满足实际教学需要。（<http://www.docin.com/p-782109039.html>，“电场强度”教案（人教版《物理》选修3-1））。本文在设计静电场点电荷可视化系统时参照的教材为人教版《物理》选修3-1，涉及到的相关内容包括：电场强度、点电荷电场以及电场强度的叠加。

对于初学静电场的学生来讲，这一部分内容比较抽象，不易理解和掌握。学生学完该节内容应掌握以下内容<sup>[2]</sup>：

- （1）知道电荷间的相互作用是通过电场发生的，看不见摸不着的电场是客观存在的一种特殊物质形态。
- （2）理解电场强度的相关概念，能够根据电场强度的相关公式进行计算。
- （3）根据库仑定律和电场强度的定义推导点电荷场强的计算公式，并能用该公式进行相应的计算。
- （4）理解电场叠加原理。

学生学习本节内容的重点是理解电场和电场强度，教学的难点是学生对电场强度概念的建立以及对公式  $E = \frac{F}{q}$  和  $E = k \frac{Q}{r^2}$  的理解。

#### 5.1.2 功能需求

根据上一小节中所分析的用户需求，本节对系统的相关功能需求进行了分析和阐述。相关功能包括：点电荷基本操作、电场控制（场线疏密、电场缩放、场强值）以及呈现相关公式以及原理等。

(1) 其中点电荷基本操作包括：加入、删除、移动点电荷以及改变电荷数量。

(2) 电场控制功能及包括场线疏密程度的控制、缩小放大电场以及选择是否呈现出每条场线的电场强度值。

(3) 为了方便教与学，系统中还加入了根据当前系统中所选择的点电荷数量、电荷量、距离呈现相关定义和推导公式。

## 5.2 系统界面设计

根据 5.1 节中的用户需求和系统功能分析，本文设计实现了静电场点电荷可视化系统。为了方便师生操作，系统界面包括了菜单栏和快捷图标。

系统中的所有功能都可以在菜单栏中的菜单按钮、电荷操作按钮、电场控制按钮以及相关原理按钮中的子菜单找到。其中菜单按钮包括视图设置、窗口、文件、帮助等。电荷操作按钮即包括加入、删除、移动点电荷等操作。电场线控制、缩放电场以及控制是否呈现电场场线数值等功能均包含于电场控制按钮中。若学生在自学时需对电场有个更全面的了解，其可在相关原理按钮中查看电场的相关定义以及推导公式。其首页界面以及相关按钮的功能如图 5.1 所示



图 5.1 静电场点电荷可视化系统首页界面及按钮相关功能简介



## 5.3 系统实现效果

根据第 4 章的实验结果，为保证电场场线的绘制效果，系统中相应算法的允许误差均设置为  $\mu \leq 10^{-6}$ 。因篇幅有限，本节主要对电荷操作、电场控制中相应部分关键功能进行阐述和展示。

### 5.3.1 电荷操作


系统中电荷操作功能包括添加、删除、移动点电荷等操作。为了方便说明，此处本文首先通过系统生成了电荷量为  $1\mu C$  的点电荷在原点( $x=0,y=0$ )处的电场，具体实现效果如图 5.2 所示，相应添加、移动点电荷均是建立在此基础之上。相应实现效果具体如下。

#### 1、添加点电荷

在如图 5.2 系统所生成的电场的基础之上，在( $x=1,y=1$ )处添加电荷量为  $2\mu C$  的点电荷，具体效果如图 5.3 所示。根据图示，可以看到两个点电荷的电场产生了叠加。

#### 2、移动点电荷

移动点电荷的操作是在上一步添加点电荷产生的电场效果基础之上实现的。

通过快捷按钮  选中需要移动的点电荷，如图 5.3 所示，可移动点电荷 q1 或者 q2。本文中选择移动点电荷 q2，移动后的位置为  $(x_2' = -2, y_2' = -2)$ ，具体效果如图 5.4 所示。

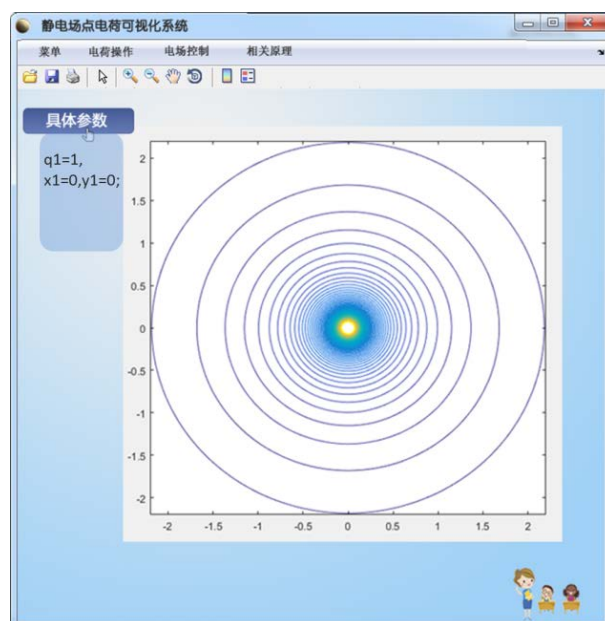


图 5.2 电荷量为  $1\mu C$  点电荷在原点 ( $0,0$ ) 处的电场

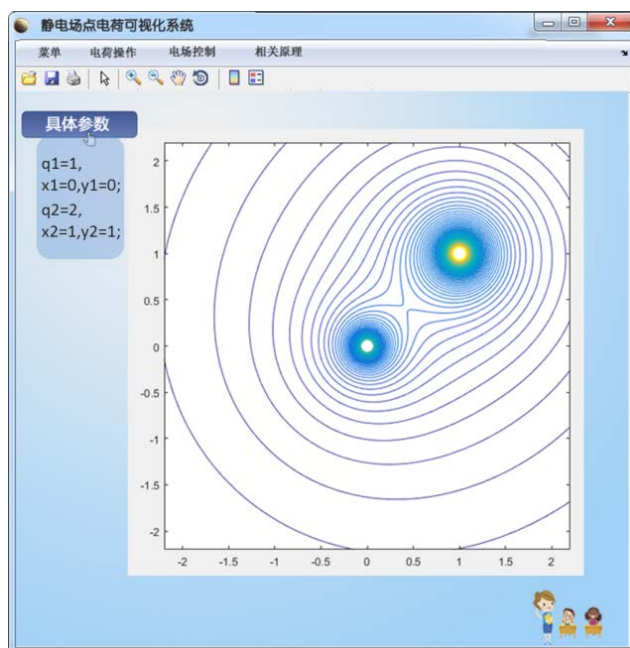


图 5.3 添加电荷量为  $2\mu\text{C}$  的点电荷于坐标  $(1,1)$  处的电场

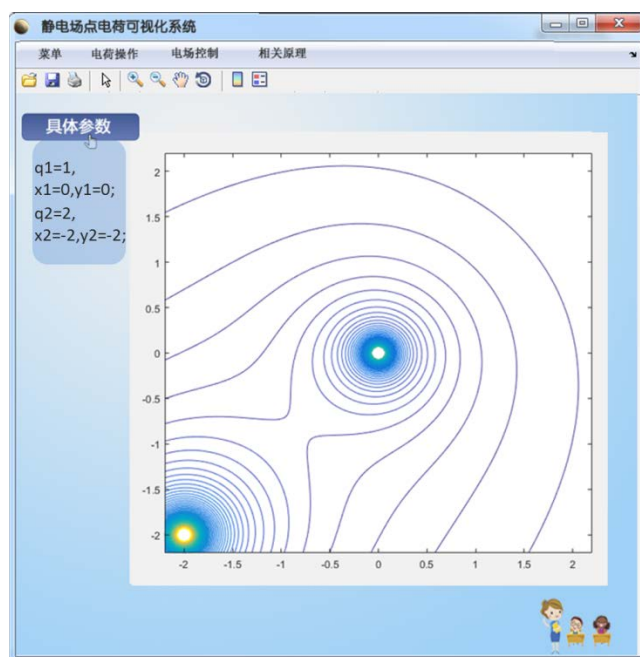



图 5.4 移动电荷量为  $2\mu\text{C}$  的点电荷于坐标  $(-2,-2)$  处的电场

### 5.3.2 电场控制的相关功能

电场控制中包括电场线控制、缩放电场以及控制是否呈现电场场线数值等功能。其中场线控制即数值场场线疏密控制。缩放电场即放大缩小电场，缩放功能的设计方便学生查看电场细节，本节中重点呈现该项功能。

电场缩放功能可以在菜单栏中的“电场控制”按钮中找到，也可以直接通过图标按钮实现。此处以图 5.4 绘制出的电场实现放大功能为例，其实现鲜

果如图 5.5 所示。

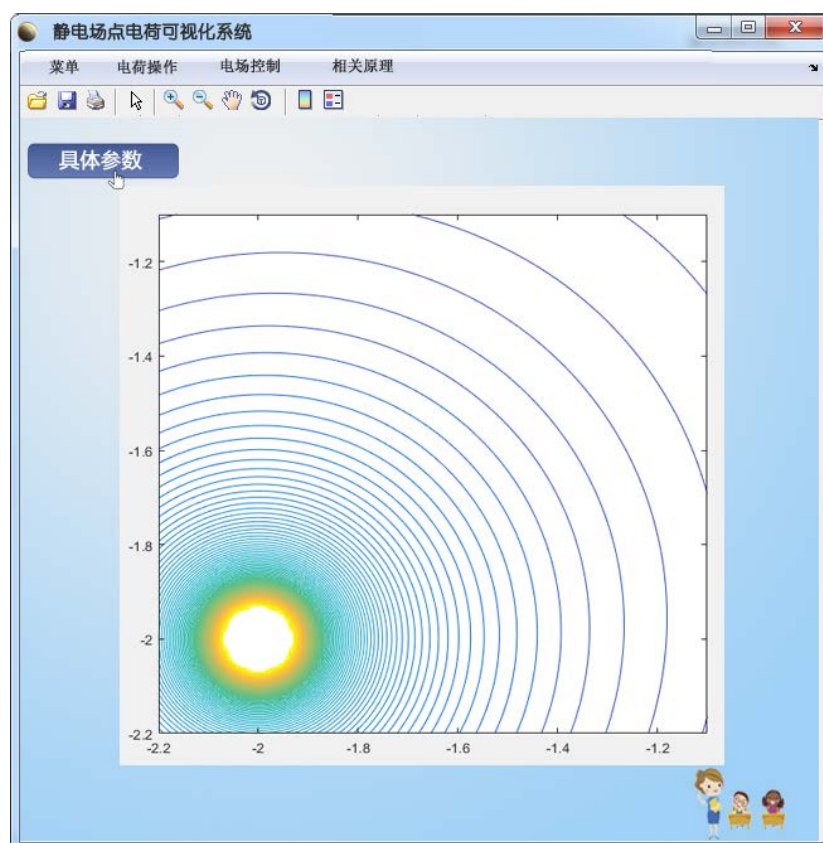


图 5.5 系统放大点电荷电场效果图

## 5.4 本章小结

本章主要对静电场点电荷可视化系统的用户需求和功能需求进行了分析，并简要阐述了系统的界面。并在 5.3 节中对系统中的部分主要关键功能进行了介绍和展示。

---

## 第 6 章 总结与展望

本文围绕数值场绘制技术问题进行了探索性研究，提出基于二维 MC 算法获取场点，并通过误差面积法对场点进行优化，减少存储空间，加快场线绘制速度。并通过该算法绘制了静电场中点电荷的电势场，场线绘制效果准确连续光滑。本文提出的数值场绘制技术，从理论上讲，不仅适用于电场的绘制，而且可以推广应用于温度场、浓度场等数值场的绘制；从实践应用上讲，本文的研究对有关物理图像演示性课件的制作提供了新的参考。本章主要总结了本文的主要工作，并指出了本文算法的创新之处，同时也总结了本文算法存在的不足之处和未来发展方向。

### 6.1 主要工作及研究结论

本课题主要围绕数值场绘制技术展开研究，对相关理论基础和技术进行了相应分析，具体内容如下：

(1) 首先对数值场绘制技术进行拆分，将场线绘制列为本项研究中最重要的工作。因此文中的综述分析内容主要从场点获取、场点优化以及连接场点三方面展开。经过几十年来的发展，相应方法技术有多种，本文结合前人研究，对其进行了归纳和总结。目前场点获取目前主要包括取点跟踪法和网格法两种；场点优化算法众多，本文对自由压缩法、Walking-driver 算法、Reumann-Witkam 算法、光栅法、Duoglas — Puecker 法等算法的原理进行了阐述，并分析了各个算法的优点和不足之处；有关场点连接的方法，本文罗列了部分常见经典的算法，如 B 样条法、Bezier 法、张力样条函数、分段三次多项式插值法，并对相应算法的优缺点进行了分析。

(2) 其次，针对本文的研究内容，详细阐述了本文所涉及到理论基础。如数值场、矢量场、隐函数、隐式曲线、MC 算法以及 Hermite 插值等相应概念和内容进行解释分析。

(3) 然后对本课题所设计的数值场绘制算法相关原理进行了详细阐述。数值场绘制的第一步即获取场点，本文根据相应的数值场数学模型通过二维 MC 算法获取场点，在此基础之上，利用面积误差法对场点进行优化压缩处理，最后借助 Hermite 插值算法连接优化后的场点进而得到场线。

(4) 最后介绍了本文算法的实验环境、实验结果对比以及数值场绘制实例。将本算法与二维 MC 算法作比较，我们发现本算法若要达到与 MC 算法同等的光滑度，对于较简单的曲线(如圆、圆角矩形等)本算法可通过其场点数量的 $1/4$ 即可实现，对于特征较多的曲线(如心形曲线、四角星曲线)，本算法可通过其场点数量

---

的1/2即可实现。在此基础上,为验证数值场绘制效果,本文选择通过该算法实现静电场中点电荷的数值场(电势场),其绘制效果较好,场线光滑流畅准确。

## 6.2 主要创新

本文主要贡献在于提出了一种数值场绘制算法,通过对数值场建模,用隐函数表示数值场,在此基础上采用二维 MC 算法获取场点,面积误差法优化场点,最后通过 Hermite 插值法绘制场线,进而得到数值场。其主要创新之处包括:

(1) 在前人研究的基础之上,本文提出一种新的数值场绘制方法,数值场的绘制只需将其数学模型表示为  $F(x, y) = 0$  的隐式形式,即可绘制出相应数值场。

(2) 本文所提出的数值场绘制算法对获取的场点通过误差面积法进行了优化压缩处理,绘制出的图像不仅光滑度和准确度仍较高,另外绘制出的数值场图像占用空间将减少,以便于网络传输等。

## 6.3 未来工作展望

在前人研究基础上,本文提出了一种新的数值场绘制算法。该算法虽然具有一定的创新之处,但也存在一定的不至于,在未来的时间里,需要考虑从以下 2 个方面进行深入研究:

(1) 本研究中的算法在文中仅对静电场中点电荷的电势场进行了绘制,实验对象相对较少。在以后的研究中应考虑本算法的适用性,需多方面验证本算法能否实现其他相关数值场的绘制,如浓度场、密度场、温度场等。

(2) 本文中数值场绘制算法,从第 4 章中的实验对比结果和绘制效果图可以知道,算法的场点优化率客观,能有效节省存储空间,并且还能根据不同误差要求控制场点优化数量和曲线光滑度,虽然本算法能较好地处理切向矢量值变化较大的场点,但是在绘制类似四角形 4 分顶点时,其光滑度不理想,这将在后续工作中有待进一步改进。