

## 摘 要

随着信息技术的广泛运用和大量数控设备的出现,现在造船模式较之以往已经产生了重大的变化,集中表现在生产数据的交换量空前庞大和交换频率的增加,大量的生产加工信息都需要数据化处理后由数据信息在数控设备之间传递,这种模式带来了生产效率的大大提高。船舶制造模式上的变化必然要求设计模式进行相应的调整以满足实际生产上的需要,那就是设计的船型便于数据化以便数控加工设备能够识别,同时能够提供数据传输的畅通渠道。

船型的数据化导致了数学船型的出现。数学船型就是用数学函数表达和设计船体型线。它要求我们寻求并采用适当的数学表达式,找到某种有效的方法,在满足设计要求的前提下求出具有实用价值并且光顺的船体型线,为水动力计算以及建造施工提供所需要的数据。船型的数字化给船舶建造带来了极大的便利,同时也给设计本身带来了很大的方便。

本论文讨论且分析了国内外船舶型线设计的现状和发展趋势。针对面临的问题,提出了自己的思路,在数学计算方法上,由于三次准均匀 B 样条曲线良好的端部性质,这里用它来拟合船舶型线,首先根据所给的初步型值求出其控制顶点,然后根据相关公式进行插值便能得到曲线上的所有插值点。在小参数模型下对型线进行了光顺,另外通过对全船的插值,并可以插值得到任意位置的曲线及其型值。

为便于型值的查询和输出,本文还提供了数据接口以便从前期模块获得原始型线数据,经过绘图时的光顺和插值处理,型值表可以进行扩充,扩充后的型值表还可以方便地存储到一个数据文件里,然后可以独立于 AutoCAD,可以用数据管理模块对数据进行管理,包括数据的修改和查询等,还可以根据需要提供给相应的生产设备以便加工。

本文采用基于 C++ 语言的 ObjectARX 工具在 AutoCAD2002 平台进行二次开发,分别编制了绘图软件和数据管理软件,并结合一条实际的普通货船,进行型线设计,取得了良好的效果。

**关键词:** 均匀 B 样条, 型线设计, ObjectARX

## ABSTRACT

Nowadays, with the appearance of lots of numerical control equipments and development of information technology, there are many important changes in model of shipbuilding, a remarkable one is that exchange of data for shipbuilding becomes more and more frequently and the amount of data for exchange also increases rapidly. At the same time, more and more digital information is required to be numbered, and then, it is digital information that is transmitted between numerical control equipment, this model brings about remarkable increase of productivity. The design model of ship is required to adjust for the change of the manufacturing model. By the occasion, it is required to number the hull, which is convenient for the distinction of numerical control equipment. Meanwhile, it is required to provide an unblocked channel of communication for the data transmission.

Because of numbering the hull, it brings about the emergence of the mathematical-hull. It is defined the mathematical ship hull to use mathematical function in designing hull lines. We must figure out some practicable equations to calculate the hull lines that should be smoothness and fit the real ship lines. These data, which are already examined, can be the original data of fluid dynamic calculation and the original data of ship-manufacture. Numbering the hull is convenient for the shipbuilding, as well as for ship-design itself.

In this thesis, the practical effect and the development of hull line design are analyzed. It offers the idea according to problem faced, considering the good character of the ends, the adjusted cubic B-spline curve is used to fit ship lines. It finds out the control vertices according to a sequence of given points, and then, through particular formula, each point in the curve could be worked out. It smoothes lines with the method based on small-parameter. Through interpolation we can calculate every point of a ship or create a line wherever in the lines plan due to given parameter.

In order to inquire and output data conveniently , this thesis also offers the data interface which can obtain original data from preceding module . After smoothness and interpolation,the moulded-values-table could be expanded.the changed table could be saved in a data file.And then , the data Data Base Management program can manage the data including interpolation and query without AutoCAD. the data could be offered to the equipments according to the demand of manufacture .

It uses objectARX tools based on C++ language in AutoCAD 2002 flat to make a farther development .It worked out a corresponding soft ware , and drew the planes of a real ship , has obtained a good effect .

**Key words:** Uniform B-Spline , ship lines, ObjectARX

## 第 1 章 绪论

### 1.1 立题背景及研究意义<sup>[1, 2, 3, 4]</sup>

现代设计方法是技术、经济及社会环境因素融为一体，充分利用电子计算机进行计算、自动绘图和管理，面向总体功能和目标，实现方案与全局构成的多元性方法的综合，在常规的工程及产品的设计中已得到广泛的应用，达到了相当高的水准，但对于一些非常规的，特别是无法用数学模型表达的外形曲面，如飞机、船舶及汽车等的外形，实现设计与生产的完全快速的数据传递自动化还存在着相当的困难。以船舶为例，船舶外形目前尚不能以数学函数方式准确模拟，只能以成千上万个离散的样点予以模拟，传统的手工设计方法是把这些样点在样台上以 1:1 的比例实物绘制。在计算机上则是将样点输入计算机中，采用 AutoCAD 等绘图软件，人机对话方法进行模拟和光顺。随机的布置、结构、造型、性能等设计计算也是采取人机对话方法不停地采取样点数据进行计算、绘图，需要几天甚至几十天。这种人机对话的模式既浪费了大量的时间，还容易出现数据传送及设计中的错误，也没能充分利用计算机的强大计算功能。尽管一些厂家和研究所在进行研究和引进软件，但并没有完全解决这些问题。如果能将这类软件融入计算机集成制造系统 (CIMS)，则对后序工程如外板的展开、自动加工提供极大的便利，缩短建造工期，创造可观的经济效益。

非常规外形曲面的整体展开及分段展开在理论上已经作了许多研究，对于一些不可展的曲面，这些研究都有一些局限性，以实用的角度，还不能得到广泛的应用，还不得不借用手工展开的方法，在计算机上利用 AutoCAD 等软件，进行半手工操作。影响了设计周期和后续设计及建造周期和精度，不能充分快速利用自动加工建造设备。以船舶为例，船舶外形设计完成取样点后，必须要进行肋骨型线及外板展开图的设计，以及结构的外板展开和剖面设计，施工建造中的外板、框架加工等都要以这些样点为基础。由于在技术上还存在着难点，人们只能在计算机中反复测取各类样点，这就大大限制了计算机的速度和设计质量，如果能开发一种软件，能利用原始样点，自动完成后续工作的取样，解决设计和加工的速度和精



度，是科技发展到今天必须解决的问题。

## 1.2 相关领域技术的发展和国内外技术现状

### 1.2.1 船舶 CAD 的发展<sup>[2,3,4,6]</sup>

计算机辅助设计作为一门学科始于 60 年代初，一直到 70 年代，由于受到计算机技术的限制，CAD 技术的发展很缓慢，进入 80 年代以来，计算机技术突飞猛进，特别是微机和工作站的发展和普及，再加上功能强大的外围设备，如大型图形显示器、绘图仪、激光打印机的问世，极大地推动了 CAD 技术的发展，CAD 技术已进入实用化阶段，广泛服务于机械、电子、宇航、建筑、纺织等产品的总体设计、造型设计、结构设计、工艺过程设计等环节。而 CAD 技术本身已成为工业生产现代化的重要标志。它对加速工程和产品的开发、缩短产品设计制造周期、提高产品质量、降低成本、增强企业市场竞争能力与创新能力发挥着重要作用。它的应用及发展正引起一场产品工程设计与制造深刻的技术革命，并对产品结构、产业结构、企业结构、管理结构、生产方式以及人才知识结构方面带来巨大影响。船舶工程领域是最早运用计算机进行辅助设计的领域之一，计算机在船舶工程中的应用极为广泛，涉及到船舶设计和制造的各个环节，CAD/CAM 技术水平的高低，直接影响着造船企业在当今竞争激烈的造船市场上的竞争力。

计算机辅助设计在上个世纪得到快速发展，大致可以分为一下四个阶段：

(1) 低级阶段。设计人员将手工设计、绘图、加工等工序编制成程序，由计算机进行数控完成。节省了设计人员的一部分重复劳动。但是各个软件处于独立运行状态，没有完整的设计系统，各个程序之间的数据无法方便的传输和调用，需要重复地输入大量的数据，复杂而费时，并容易出错。

(2) 中级阶段。应用产品数据管理库 PDM 软件，针对早期阶段的孤岛式信息，这个阶段主要解决的是企业内部信息的集成，使信息能正确、高效地共享与交换。

(3) 中高级阶段。在信息集成的基础上，将产品开发过程中的串行过程尽可能地转换为并行过程，在并行设计的环境下，各阶段工作及时交流和

协调, 实现设计、制造的自动化。

(4) 高级阶段. 结合 CAD/CAM、柔性制造系统 (FMS) 及生产自动管理系统组成全方位的计算机辅助与控制系统, 进入计算机集成制造系统 (CIMS)。逐步实现概念设计、虚拟设计、敏捷制造等技术。这个阶段除了实现信息集成和过程集成, 还实现企业集成, 强调智能化, 使企业在瞬息万变的市场环境下, 能抓住稍纵即逝的机遇, 开发出能充分满足客户要求的产品。

### 1.2.2 国内外船舶 CAD 的研究现状<sup>[1,4,5]</sup>

随着近年来计算机硬件技术的迅猛发展, 国际上各种先进的 CAD/CAM 系统都有了很大的发展, 代表性的有:

1. 瑞典的 TRIBON 系统. 该系统以 Steerbeer 系统为主, 后来又合并了挪威的 Autokon、德国的 Schiffko 和英国的 BMT 等系统发展起来的。TRIBON 系统包括船舶的初步设计、生产设计、计划管理的全过程, 拥有自行开发的图形支撑软件及工程数据库平台, 能建立船舶产品电子信息模型, 而且在出图等方面表现出众, 具有参数化造型、线框建模等实体模型修改特点。

2. 西班牙的 FORAN 系统. 它是一个从基本概念设计到钢结构加工、机舱布置和船舶舾装作业的功能强大的造船集成系统。

3. 美国 PTC 公司的 CADD5 系统, 该系统包括船型与数据结构、钢结构设计和热力、通风、空调等模块, 该系统采用了 PTC 公司的 Behavior Modeling 先进技术, 具有远大的发展前景。

我国造船界从上世纪 60 末开始 CAD/CAM 的开发与应用, 也取得了相当的成就, 如原中船总组织开发的 CASIS-I、CASIS-II, CASIS-II 在 CASIS-I 的基础上提高了系统的集成度, 在总布置设计、基本结构设计、结构零件生成、船舶管系的生成等方面都有较大的突破, 对缩短造船周期、提高造船质量起到了一定的作用。一些相关院校也组织力量开发了一些 CAD 系统, 如“船舶曲面设计系统”、“内河船舶 CAD 系统”、“油船初步设计软机系统”等, 但总的说来, 在实用上都或多或少的存在一些问题, 特别是在贯穿全部设计、生产过程的船型数据, 样点及派生样点的管理等问题上还没有很好的解决, 大大限制了设计、生产的速度和企业的竞争力。

## 1.3 课题的主要研究内容和关键技术

### 1.3.1 总体目标

本课题是省科委的实际课题“复杂曲面的自动展开方法研究及实用软件系统的开发”的一部分,在宏观上,船舶制造业中已有较为完整的 CAD/CAM 集成系统,但在船舶型线样点以及外板展开、肋骨型线派生的型线样点没有实用的数据管理系统,只能利用人工操作多次反复地制作来完成,例如,实船设计中通常要绘制上百条的肋骨型线,这种人工操作大大影响了设计建造速度。本课题将研究开发实用型的样点数据管理系统,使本软件能够根据型值表自动产生型线图,在肋骨型线图中根据指定的位置绘制肋骨型线,并且保证在一定的精度内型线三向光顺,同时操作界面友好。另外由于本程序只是上述软件的一部分,还要使本程序与其他模块的方便对接。

### 1.3.2 研究内容与路线

为了实现课题的研究目标,本课题完成了以下工作:

#### 1) 建立自动绘图系统

根据型值表按照指定的要求自动绘出符合要求的线条,是本程序的核心内容。鉴于目前大部分的船舶工程图纸都以 AutoCAD 绘制,为了给后续工作提供便利,本程序绘图部分用 Visual C++ 语言来实现,由于本论文做的是 AutoCAD 的二次开发,绘图直接在 AutoCAD 下面操作,生成的是 AutoCAD 格式的图纸,在图纸的处理方面可以直接借用 AutoCAD 这方面完善的功能。

#### 2) 研究开发样点及派生样点的数据管理系统

本论文将建立样点数据管理工具,利用它可以根据需要扩展原始型值表,方便的查询船舶任意位置型线的型值,这对造船中肋骨的加工有重要的意义。

#### 3) 建立本程序与主程序系统及其他模块的接口系统

本程序作为一个软件模块,必须能被主程序方便地调用,服从整体功能;并且本程序的输入——型值表,可能是其他静水力计算模块的输出,当然也可以直接输入。而本程序的输出如肋骨型线图及其型值等可能是建造、结构布置等程序模块的输入。随着软件的发展,接口系统可以使各个模块方便对

接，自动调用，尽量减少人工过多干预。

实现上述研究内容的过程中,采用的技术路线归结起来主要有以下几条：

1)主要以三次 B 样条拟合船体型线。对于船舶首尾曲度较大的区域，酌情辅以其他样条曲线，要保证拟合的精度。

2) 以 ObjectARX 作为程序开发工具，它强大的功能和高效率代表着 AutoCAD 二次开发的方向。

3) 在充分利用 AutoCAD 现有界面的基础上，设计比较友好的用户界面。

### 1.3.3 关键技术

外形不可展曲面样点数据的访问技术将是本程序的基础，型线的插值与拟合是本课题的关键，特别是首尾端复杂曲线的拟合是本课题的难点。用户界面与接口也是能否赢得用户的重要因素。



## 第 2 章 B 样条相关理论和型线的插值

### 2.1 B 样条的概念<sup>[7, 9, 50]</sup>

B 样条的概念最初是由 Schoenberg 于上个世纪 40 年代提出来的, 如今已得到很大的发展。B 样条有许多种等价定义, 其表达方式虽然各不相同, 但是在数学上却有着内在的联系, 实质上是统一的。

B 样条曲线定义为:

$$P(t) = \sum_{i=0}^n V_i N_{i,k}(t) \quad (2-1)$$

其中,  $V_i (i=0, 1, \dots, n)$  是控制多边形的顶点,  $N_{i,k}(t) (i=0, 1, \dots, n)$  称为  $k$  阶 ( $k-1$  次) B 样条基函数, 其中每一个称为 B 样条, 它是一个称为节点矢量, 即非递减的参数序列  $T: t_0 \leq t_1 \leq \dots, t_{n+k}$  所决定的  $k$  阶分段多项式, 也称为  $k$  阶 ( $k-1$  次) 多项式样条。

B 样条还有有多种等价定义, 在理论上较多地采用截尾幂函数的差商定义。这里只介绍作为标准算法的 de Boor-Cox 递推定义, 又称为 de Boor-Cox 公式。约定  $0/0=0$ 。

$$N_{i,0}(t) = \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{其它} \end{cases}$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t) \quad (2-2)$$

该递推公式表明: 欲确定第  $i$  个  $k$  阶 B 样条  $N_{i,k}(t)$ , 需要用到  $t_i, t_{i+1}, \dots, t_{i+k}$  共  $k+1$  个节点, 称区间  $[t_i, t_{i+k}]$  为  $N_{i,k}(t)$  的支承区间。曲线方程中  $n+1$  个控制顶点  $V_i (i=0, 1, \dots, n)$ , 要用到  $n+1$  个  $k$  阶 B 样条基函数  $N_{i,k}(t)$ 。或者称  $K$  次 B 样条。它们支撑区间所含节点的并集就是定义了这一组 B 样条基的节点矢量  $T = [t_0, t_1, \dots, t_{n+k}]$ 。

## 2.2 准均匀三次 B 样条曲线<sup>[8, 10, 12, 51, 53, 54]</sup>

### 2.2.1 准均匀三次 B 样条及其正算

均匀 B 样条基在曲线定义域内各节点区间上具有用局部参数表示的统一的表达式, 使得计算与处理简单方便。但它定义的均匀 B 样条曲线有个缺点, 就是没有保留贝齐尔曲线的端点几何性质, 样条曲线的首尾端点不再是控制多边形的首末顶点, 高于二次的均匀 B 样条曲线在端点处不再与控制多边形相切。采用准均匀 B 样条曲线类型就是为了解决这个问题, 他可以使人们对曲线在端点的行为有较好的控制。

三次准均匀 B 样条曲线的节点矢量中两端节点具有重复度四, 所有内节点成均匀分布, 具有重复度一。由此决定的准均匀 B 样条基所定义的准均匀 B 样条曲线就具有同次贝齐尔曲线的端点几何性质。

在曲线定义域内, 除两端 3 个节点区间外, 三次准均匀 B 样条基在其他节点区间上有着与三次均匀 B 样条基相同的图形, 但当定义域内节点区间次数少于  $2(k-1)=4$  个时, 具有重复度四的两端节点对这些区间上的 B 样条基的影响将相互交织一起, 因而形成多种不同的情况。当把每个区间上的 B 样条基都从整体参数  $u \in [u_i, u_{i+1}]$  变换到局部参数  $t \in [0, 1]$  时, 这些受影响的节点区间上的 B 样条基系数矩阵也将出现不同的情况。这些将导致准均匀 B 样条曲线在计算与处理上皆要比均匀 B 样条曲线复杂的多。这也许时为了获得贝齐尔曲线的端点几何性质所要付出的代价。在实践中是采用均匀方案抑或采用准均匀方案, 应该根据需要平衡得失来确定。

对于船舶型线而言, 船舶曲线的端点往往是给定的边界条件, 控制顶点与型线的端点重合会给型线设计带来许多方便, 因此本论文中拟合型线大量运用的是三次准均匀 B 样条。

给定控制顶点  $V_i (i=0, 1, \dots, n)$ , 定义一条三次准均匀 B 样条曲线。在采用局部参数下, 第  $l$  段曲线方程可以写为:

$$s_l = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M \begin{bmatrix} V_l \\ V_{l+1} \\ V_{l+2} \\ V_{l+3} \end{bmatrix}, \quad 0 \leq t \leq 1, l=0, 1, \dots, n-3 \quad (2-3)$$

其中  $M$  是三次准均匀 B 样条基的系数矩阵，它随着节点个数的以及所处的区间的不同而呈现不同的形式，而不再象均匀 B 样条有那样统一的形式。这样对于任意的参数  $t$ ，都可以把曲线上相应位置的点计算出来，这个已知控制点求曲线上点的过程称之为 B 样条的正算。

对于船舶中的型线，通常都至少有 6 个以上的型值点，就需要针对不同的情况选择相应的系数矩阵。下面是给定 6 个以上型值点集合时最后一段曲线对参数  $u$  的插值函数：

```
void Arrsplines_last(double u, const AcGePoint3dArray& CtrsArray, AcGePoint3d
&ptu)
{
    double mat[4][4]={ {-1/6.0, 11/12.0, -1.75, 1.0},
                        { 0.5, -1.25, 0.75, 0 },
                        {-0.5, -0.25, 0.75, 0},
                        {1/6.0, 7/12.0, 0.25, 0}};

    int n=CtrsArray.length()-2;
    for(int i=0;i<4;i++)
    tmp[i]=u*u*u*mat[0][i]+u*u*mat[1][i]+u*mat[2][i]+mat[3][i];
    ptu.x=tmp[0]*CtrsArray[n-2].x+tmp[1]*CtrsArray[n-1].x+tmp[2]*CtrsArray[n].x
    +tmp[3]*CtrsArray[n+1].x;
    ptu.y=tmp[0]*CtrsArray[n-2].y+tmp[1]*CtrsArray[n-1].y+tmp[2]*CtrsArray[n].y+t
    mp[3]*CtrsArray[n+1].y;
}
```

### 2.2.2 准均匀三次 B 样条曲线的反算

已知控制顶点，根据上述公式可以计算 B 样条曲线上的每一个点。但是在船舶型线的设计中，往往给出的是通过母型船改造法、图谱法等方法初步设计出来的型值，要对这些型值进行 B 样条拟合和计算，必须首先求出某组型值的控制多边形的顶点，即根据给定的型质点计算控制多边形的顶点，较之于前面的过程，这种个过程称之为反算。

计算控制顶点的转换矩阵如下所示，有 5 种具有代表性的情况：

第一段曲线：

$$P_1(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & \frac{7}{4} & -\frac{11}{12} & \frac{1}{6} \\ 3 & -\frac{9}{2} & \frac{3}{2} & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \quad (2-4)$$

第二段曲线

$$P_2(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{4} & \frac{7}{12} & -\frac{1}{2} & \frac{1}{6} \\ \frac{3}{4} & -\frac{5}{4} & \frac{1}{2} & 0 \\ \frac{3}{4} & \frac{1}{4} & \frac{1}{2} & 0 \\ -\frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{7}{12} & \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} \quad (2-5)$$

第三段至 n-3 段(n 为型值点总个数)曲线

$$P_i(t) = \frac{1}{6} \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_i \\ V_{i+1} \\ V_{i+2} \\ V_{i+3} \end{bmatrix} \quad (2-6)$$

第 n-2 段(倒数第二段)曲线

$$P_{n-2}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{7}{12} & \frac{1}{4} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} V_{n-2} \\ V_{n-1} \\ V_n \\ V_{n+1} \end{bmatrix} \quad (2-7)$$

第 n-1 段(最后一段)曲线

$$P_{n-1}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{11}{12} & -\frac{7}{4} & 1 \\ \frac{1}{2} & -\frac{5}{4} & \frac{3}{4} & 0 \\ -\frac{1}{2} & \frac{1}{4} & \frac{3}{4} & 0 \\ \frac{1}{6} & \frac{7}{12} & \frac{1}{4} & 0 \end{bmatrix} \begin{bmatrix} V_{n-1} \\ V_n \\ V_{n+1} \\ V_{n+2} \end{bmatrix} \quad (2-8)$$

为简便计, 记乃  $P_i(0)$  为  $P_i$ , 则由式(2-4)至式((2-8), 可以得到下式:

$$\begin{cases} P_1 = V_1 \\ P_2 = \frac{1}{4}V_2 + \frac{7}{12}V_3 + \frac{1}{6}V_4 \\ P_i = \frac{1}{6}V_i + \frac{2}{3}V_{i+1} + \frac{1}{6}V_{i+2}, (i=3, 4, \dots, n-3) \\ P_{n-2} = \frac{1}{6}V_{n-2} + \frac{2}{3}V_{n-1} + \frac{1}{6}V_n \\ P_{n-1} = \frac{1}{6}V_{n-1} + \frac{7}{12}V_n + \frac{1}{4}V_{n+1} \\ P_n = V_{n+2} \text{ (最后一段曲线的末端)} \end{cases} \quad (2-9)$$

式(2-9)即为根据给定型值点列求解插值 B 样条曲线控制点的基本方程组。

### 2.2.3 端点条件和构造插值三次 B 样条曲线的方程组

式((2-9)中的方程的数目为  $n$ , 欲求顶点数为  $n+2$ , 还缺少两个条件。为此, 在这里补充两个端点条件。设首、末端切矢为已知, 则可以得到:

$$\text{首端切矢: } P_1' = 3(V_2 - V_1) = 3V_2 - 3P_1 \quad (2-10)$$

$$\text{末端切矢 } P_n' = 3(V_{n+2} - V_{n+1}) = -3P_n - 3V_{n+1}, \quad (2-11)$$

综合式((2-8)、式(2-9)和式(2-10), 可以得到根据给定的型值点计算控制点的方程组如下:



$$\begin{bmatrix}
 3 & 0 & & & & & & & & & \\
 & \frac{1}{4} & \frac{7}{12} & \frac{1}{6} & & & & & & & \\
 & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & & & & \\
 & & & \ddots & \ddots & \ddots & & & & & \\
 & & & & \ddots & \ddots & \ddots & & & & \\
 & & & & & \frac{1}{6} & \frac{3}{2} & \frac{1}{4} & & & \\
 & & & & & & \frac{1}{6} & \frac{7}{12} & \frac{1}{4} & & \\
 & & & & & & & 0 & 3 & & 
 \end{bmatrix}
 \begin{bmatrix}
 V_2 \\
 V_3 \\
 V_4 \\
 \vdots \\
 \vdots \\
 \vdots \\
 V_{n-1} \\
 V_n \\
 V_{n+1}
 \end{bmatrix}
 =
 \begin{bmatrix}
 P'_1 + 3P_1 \\
 P_2 \\
 P_3 \\
 \vdots \\
 \vdots \\
 \vdots \\
 P_{n-2} \\
 P_{n-1} \\
 3P_n - P'_n
 \end{bmatrix}
 \quad (2-12)$$

解方程组((2-12))就可以得到  $V_2$ 、 $V_3$  .....  $V_{n+1}$ ，因为已知  $V_1 = P_1$ ，

及  $V_{n+2} = P_n$ ，从而求得全部顶点。

两端取自由端条件时，则式((2-12))将变为如下方程组：

$$\begin{bmatrix}
 9 & -3 & & & & & & & & & \\
 & \frac{1}{4} & \frac{7}{12} & \frac{1}{6} & & & & & & & \\
 & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & & & & \\
 & & & \ddots & \ddots & \ddots & & & & & \\
 & & & & \ddots & \ddots & \ddots & & & & \\
 & & & & & \frac{1}{6} & \frac{3}{2} & \frac{1}{4} & & & \\
 & & & & & & \frac{1}{6} & \frac{7}{12} & \frac{1}{4} & & \\
 & & & & & & & -3 & 9 & & 
 \end{bmatrix}
 \begin{bmatrix}
 V_2 \\
 V_3 \\
 V_4 \\
 \vdots \\
 \vdots \\
 \vdots \\
 V_{n-1} \\
 V_n \\
 V_{n+1}
 \end{bmatrix}
 =
 \begin{bmatrix}
 P_1 \\
 P_2 \\
 P_3 \\
 \vdots \\
 \vdots \\
 \vdots \\
 P_{n-2} \\
 P_{n-1} \\
 6P_n
 \end{bmatrix}
 \quad (2-13)$$

其首端顶点  $V_1 = P_1$ ，末端顶点  $V_{n+2} = P_n$ 。

将式(2-12)和式(2-13)合并，可以得到如下公式：

$$\begin{bmatrix}
 b_1 & c_1 & & & & & & & & & \\
 & \frac{1}{4} & \frac{7}{12} & \frac{1}{6} & & & & & & & \\
 & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & & & & \\
 & & & & & \ddots & \ddots & \ddots & & & \\
 & & & & & & \ddots & \ddots & & & \\
 & & & & & & & \frac{1}{6} & \frac{3}{2} & \frac{1}{4} & \\
 & & & & & & & \frac{1}{6} & \frac{7}{12} & \frac{1}{4} & \\
 & & & & & & & & a_n & b_n & \\
 & & & & & & & & & & 
 \end{bmatrix} \cdot \begin{bmatrix} V_2 \\ V_3 \\ V_4 \\ \vdots \\ \vdots \\ V_{n-1} \\ V_n \\ V_{n+1} \end{bmatrix} = \begin{bmatrix} d_1 \\ P_2 \\ P_3 \\ \vdots \\ \vdots \\ P_{n-2} \\ P_{n-1} \\ d_n \end{bmatrix} \quad (2-14)$$

式中  $b_1, c_1, d_1, a_n, b_n, d_n$  分别按如下公式选取:

当给定端点切矢时

$$\begin{aligned}
 b_1 &= 3, \quad c_1 = 0, \quad d_1 = P_1' + 3P_1 \\
 a_n &= 0, \quad b_n = 3, \quad d_n = 3P_n + 3P_n'
 \end{aligned} \quad (2-15)$$

当取自由端条件时

$$\begin{aligned}
 b_1 &= 9, \quad c_1 = -3, \quad d_1 = 6P_1 \\
 a_n &= -3, \quad b_n = 9, \quad d_n = 6P_n
 \end{aligned} \quad (2-16)$$

两种端点条件可以混合使用。但是它们的首、末顶点都分别与首、末型

值点重合, 即  $V_1 = P_1$  及  $V_{n+2} = P_n$ 。

这样求得了型值点的控制顶点, 就可以按照前面提供的公式, 通过插值得到拟合曲线上的任意一点, 连接这些点便得到所求的拟合曲线。

下面是端点自由支持条件下反求控制点的函数原型。

```

void GetCtrs(const cGePoint3dArray& MArray, AcGePoint3dArray& CtrArray)
{ .....
    const int n=MArray.length();
    CtrArray.setLogicalLength(n+2);
    double *a,*b,*c,*l,*u,*d,*t;
    a=(double *)calloc( n, sizeof( double ) );//1---n-1
    b=(double *)calloc( n, sizeof( double ) );//0---n-1
    c=(double *)calloc( n, sizeof( double ) );//0---n-2
    d=(double *)calloc( n, sizeof( double ) );//0---n-2
    }
    
```

```

l=(double *)calloc( n, sizeof( double ) );//1---n-1
t=(double *)calloc( n, sizeof( double ) );//0---n-1,
u=(double *)calloc( n, sizeof( double ) );//0---n-1
//初始化转换矩阵
//a[i] b[i] c[i]
.....
//计算
for(i=0;i<n-1;i++)
d[i]=c[i];
u[0]=b[0];
for(i=1;i<n;i++)
{ l[i]=a[i]/u[i-1]; u[i]=b[i]-l[i]*c[i-1]; }
//求控制点的横坐标 x
t[0]=6*MArray[0].x;//d1=6P1
for(i=1;i<n-1;i++)
{ t[i]=MArray[i].x-l[i]*t[i-1]; }
t[n-1]=6*MArray[n-1].x-l[n-1]*t[n-2];//dn=6Pn
CtrArray[n-1].x=t[n-1]/u[n-1];
for(i=n-2;i>=0;i--)
{ CtrArray[i].x=(t[i]-c[i]*CtrArray[i+1].x)/u[i];}
//求控制点的纵坐标 y
t[0]=6*MArray[0].y;//d1=6P1
for(i=1;i<n;i++)
{ t[i]=MArray[i].y-l[i]*t[i-1]; }
t[n-1]=6*MArray[n-1].y-l[n-1]*t[n-2];//dn=6Pn
CtrArray[n-1].y=t[n-1]/u[n-1];
for(i=n-2;i>=0;i--)
{ CtrArray[i].y=(t[i]-c[i]*CtrArray[i+1].y)/u[i]; }
CtrArray[n+1]=MArray[n-1]; //尾 端点
for(i=n;i>0;i--)
CtrArray[i]=CtrArray[i-1];
CtrArray[0]=MArray[0]; //首端点
.....

```

}

## 2.3 B 样条拟合船舶型线<sup>[7, 9, 13, 15]</sup>

### 1. 单根型线的拟合

在给定了一组型值及其边界条件以后, 先利用三次准均匀 B 样条的反算求得型线的控制顶点, 然后进行 B 样条的正算过程, 便得到拟合曲线上的所有型值点。在每一段上取足够小的步长, 把插值得到的点连接起来, 便能得到足够光滑的拟合曲线。

设某船半条水线的型值如下表所示: (单位: m)  $y'_1 = 0.51$ ,  $y'_8 = 0$

$X_i$	0.0	3.3	6.6	9.9	13.2	19.8	26.4	33.0
$Y_i$	1.27	2.68	4.10	5.2	5.86	6.37	6.40	6.40

表 2-1

用准三次 B 样条拟合, 则可算得有 8 个控制点, 如下表所示:

$X_i$	0.0	1.06	3.33	6.54	9.84	14.26	19.85	26.4	30.8	33.0
$Y_i$	1.27	1.82	2.62	4.15	5.27	6.11	6.45	6.39	6.4	6.4

表 2-2

然后根据控制点对每一段曲线插值, 在步长为  $t=0.001$  下, 插得的系列点连为曲线如下图:

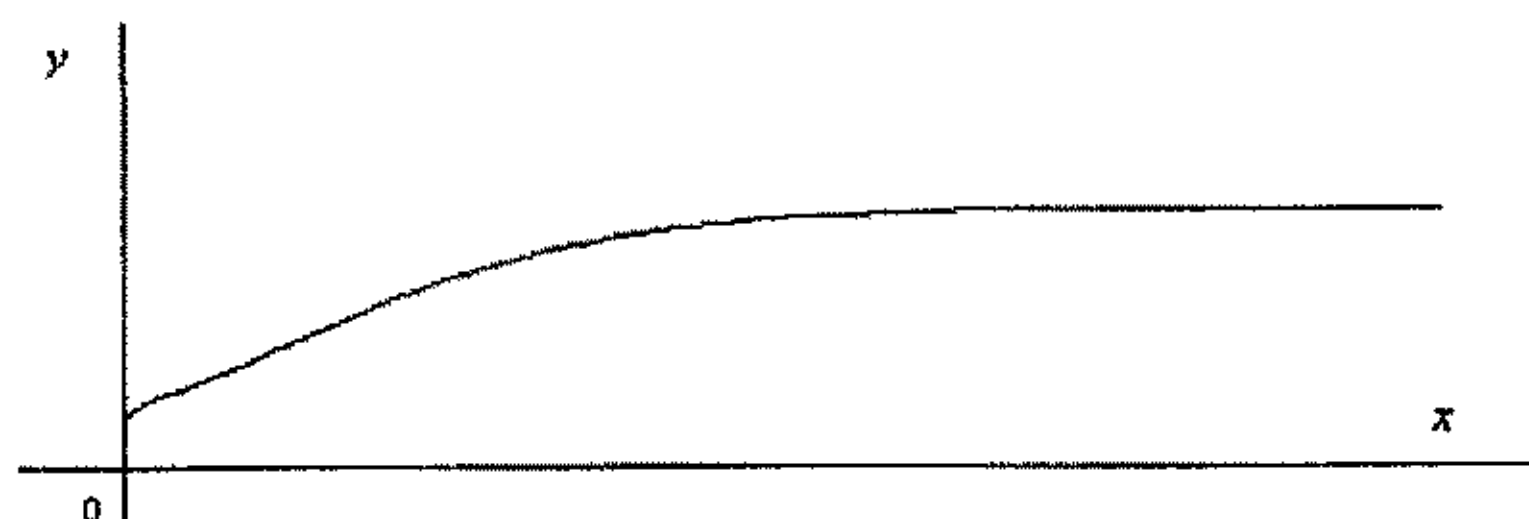


图 2-1 拟合的半条水线

可见拟合效果还是可以接受的。将控制多边形与曲线同一个图中绘制, 左端经放大则如图 2-12 所示:

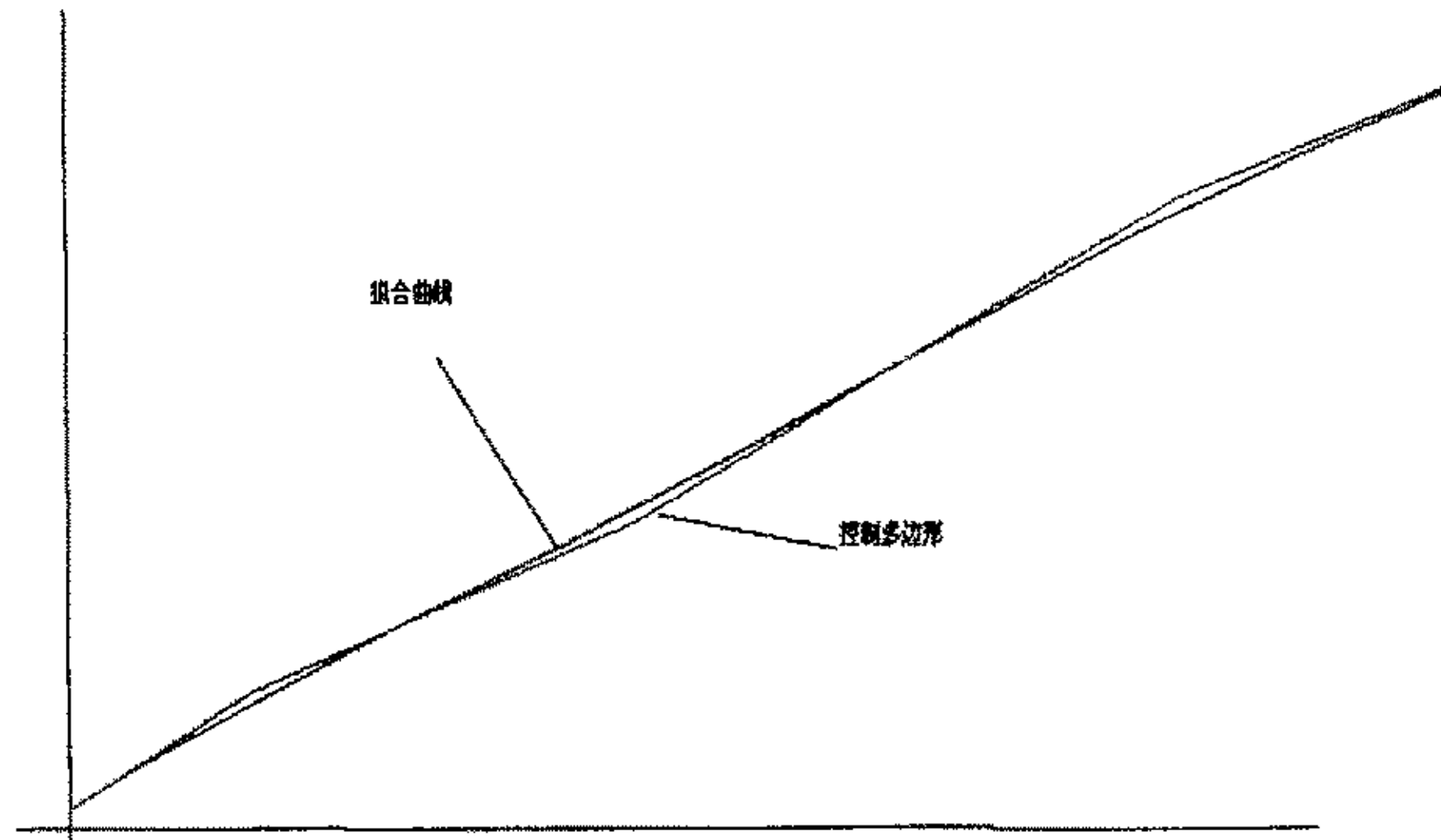


图 2-2 拟合曲线及其控制多边形（局部）

从表 2-1，表 2-2 和图中都可以看出，控制多边形和曲线的端点重合，并且性也得到了体现。

上述计算过程是固定的，只要步长取的足够小，总能满足插值的精度，唯一的变数就是端部条件，同样的一组型值，端部条件的不同可以使型线的形状发生很大的变化，下面同一组型值在端部条件不同时拟合得到的两条曲线：（左端点导数分别为 0.87 和 0.0）

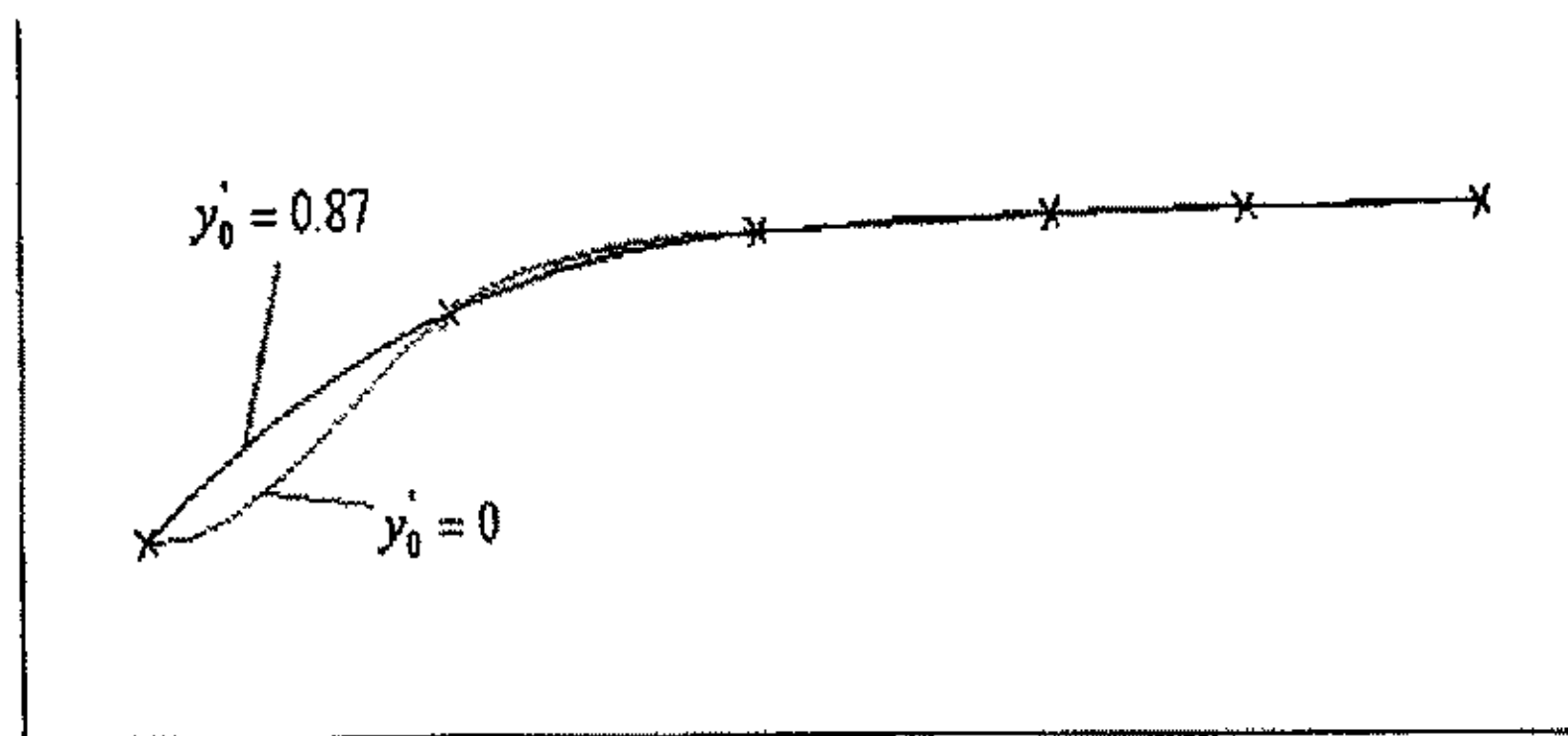


图 2-3 端点条件对曲线拟合的影响

由此可以看出，对船舶型线拟合效果的好坏，直接取决于对船舶每条型线端部条件的设置是否合理。然而，船舶型线的端部条件非常复杂，具有很大的随机性，离开人工的干预，想把每条型线的端部条件都设置合理是不可能的，我们只能针对不同的船型，找出大致的规律。本文计算的实船是一条普通货船，型线拟合中对型线端部条件的设置注意的主要有以下几点：



1).半宽水线图中水线、主甲板等与首端圆弧的切点估算以后,根据圆的规则性可以比较方便的计算出此处的切向,作为线条主体部分的前端部切向。对于位置比较高的水线和主甲板线,由于曲线尾部的比较平直,可以看作是自由支持。而位置较低的水线和船底板,由于与水线图的基线夹角很小,斜率绝对值不超过 0.4。

2)纵剖线中的主甲板线,舷樯顶线,首尾楼甲板边线由于比较平缓,都可以看作是两端自由支撑,轮廓线近似为直线段,故首端点也可以看作是自由支持。

3)除了首尾附近的横剖线,船中部大多数横剖线在下端点处与船底想切,由于本论文计算的实船没有舦升高,直接与平第相切而斜率为零,主甲板,首尾楼甲板,舷樯顶线在中线面处的端点的斜率也为零。

这里只是讲了一些端点情况比较容易确定的地方,并且也只限于本文所计算的船的情况,对于不同的船舶,要视具体情况而定。

### 2.1.1 线性推广：参数物理模型

$$y=Y(x)$$


梁段的两个横剖面各为  $aa$  和  $bb$ , 经放大后, 令  $\rho$  是梁的中心轴上的点  $A$  处的曲率半径, 而  $BB$  和  $bb$  是梁上的同一剖面在受弯以前和受弯以后的位置, 在  $bb$  剖面上距离中心轴为  $y$  的地方任取一点  $P$ , 而  $Q$  是点  $P$  点在弯曲后的位置。根据受力平衡条件, 则梁在受力平衡后, 满足以下条件:

$$\frac{d^2}{dx^2} \frac{y''}{(1+y'^2)^{3/2}} = \frac{d^2}{dx^2} \left( \frac{1}{EJ} M \right) = 0 \quad (3-1)$$

(3-1) 是在  $x$  不是处于压铁压点位置推导出来的, 在  $x$  处于压点位置时, 样条的剪切力将有间断点, 因此  $Y(x)$  的三阶导数将有间断点, 但是  $Y(x)$  的一阶、二阶导数则是处处连续的; 这里把具有一阶、二阶连续导数且在相邻型值点之间满足微分方程 (3-1) 的一类曲线称为样条曲线。

用样条曲线来描述船体型线, 不仅能保证型线具有足够的光滑性, 而且由于样条曲线的曲率具有分段成线性变化这一性质, 可知样条在相邻型值点之间至多只有一个拐点, 这个性质对于考查和调整型线在各部分的弯曲的情况是很有用的。

### 3.1.2 两点边值问题及求解<sup>[18,20,21]</sup>

求解样条曲线就是要分段求解微分方程 (3-1), (3-1) 是一个四阶的常微分方程, 需要有四个条件来作为其定解条件, 下面我们将研究具有下列定解条件的定解问题:

$$\begin{cases} \frac{d^2}{dx^2} \frac{y''}{(1+y'^2)^{3/2}} = 0 \\ Y(a) = y_a, Y(b) = y_b, m(a) = m_a, m(b) = m_b \end{cases} \quad (3-2)$$

其中

$$m(x) = \frac{y''}{(1+y'^2)^{3/2}}$$

称之为样条函数  $Y(x)$  的曲率函数。

问题 (3-2) 的求解特称之为两点边值问题, 它是求解样条曲线的中的子模块之一。

在用小参数法求解样条函数的两点边值问题时要求  $|Y'_\epsilon(x)|$  必须充分的小,其实质是要保证样条是的变形在弹性范围内。因此,在用小参数法求解样条曲线时,就势必要考虑条件

$$|Y'(x)| \ll 1 \quad (3-3)$$

对于近似解精确度的影响。

条件 (3-3) 对于某些部位的船型线,例如,在靠近船体舭部的水线上,这个条件满足的很好,但在靠近船底的横剖线部分特别是舭部,这个条件就不能满足。

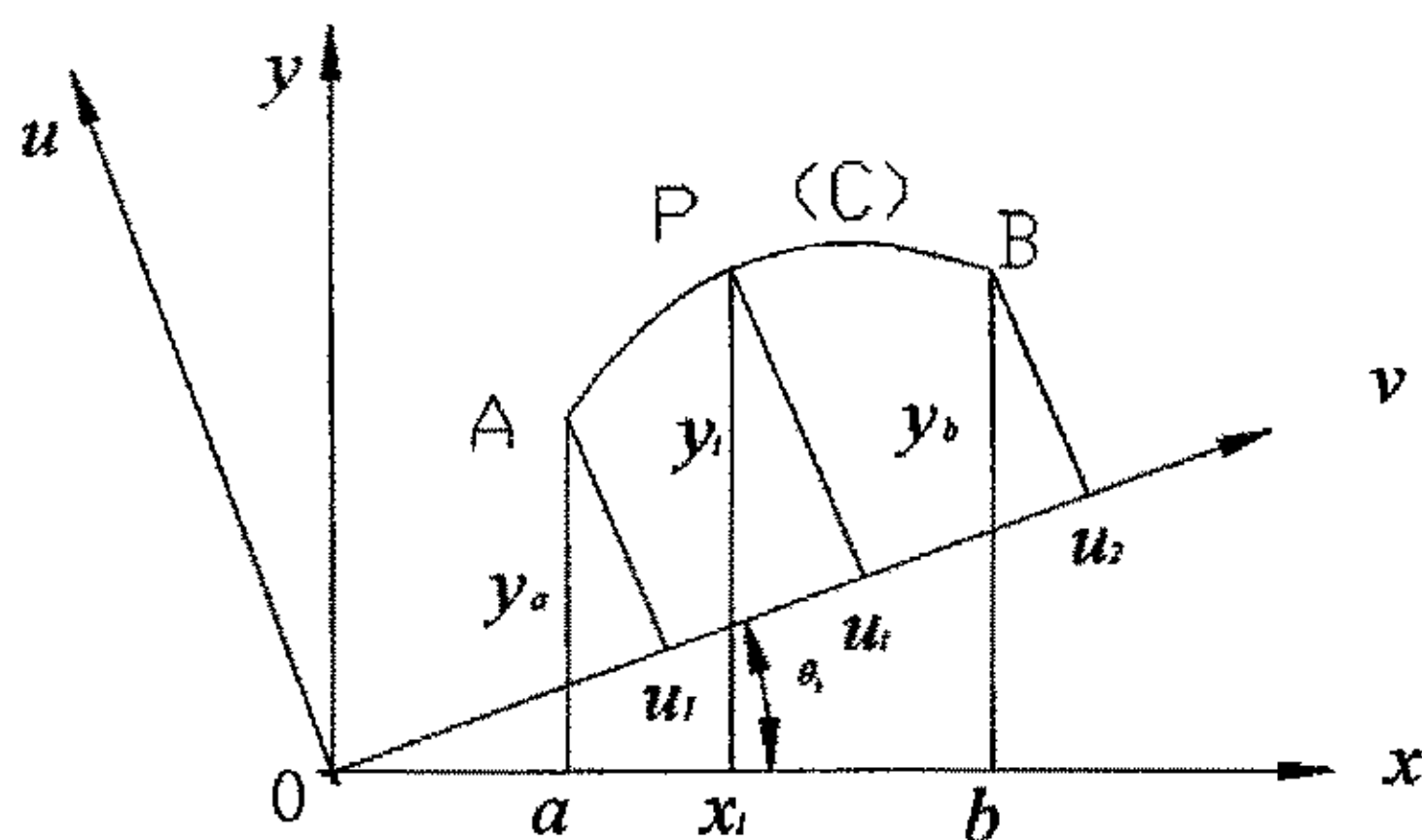
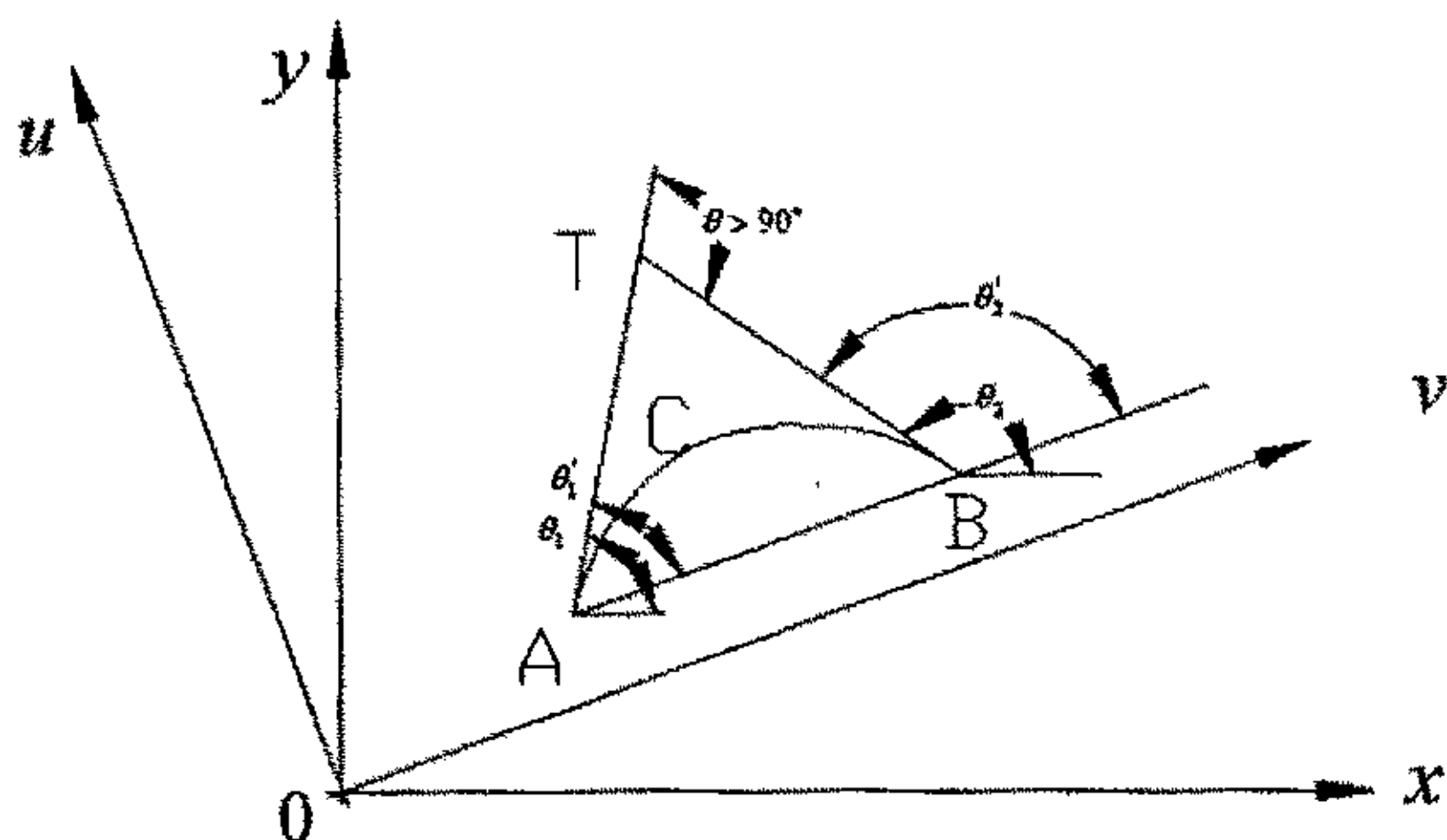


图 3-2 坐标轴旋转

$Y'(x)$  的几何意义是船型线的切线斜率,其数值的大小,表示船型线在当地的陡峻或平坦的程度。随着坐标系的选择的不同,陡峻与平坦可以相互转化,如图 3-2,曲线 (C) 在  $xoy$  坐标系中是比较陡的,但在  $vou$  坐标系中,它就变得比较平坦了,可见利用坐标系的旋转,可以把斜率比较大的曲线化为斜率比较小的曲线,但是在旋转坐标轴的同时,也可以发生这样的情况:即曲线的某一部分的斜率在旋转以后,其绝对值确实变小了,而另一部分曲线的斜率却反而变大了,当曲线在全长上的切线的转角大于  $90^\circ$  度时,如图 3-3,就会有这种情形。


 图 3-3 切线转角增量大于  $90^\circ$  的情形

在图 3-3 中弧 ACB 是一段凸曲线，当 C 沿着曲线从 A 变化到 B 时，其切线的转角  $\theta$  在 B 处达到极大，当  $\theta$  大于  $90^\circ$  时，无论怎样旋转坐标轴，例如把  $ox$  轴旋转到与 AB 平行的  $ou$  的位置，则 A, B 处的切线与  $ou$  轴的交角： $\angle TAB$  与  $\angle TBA$  中必将有一个是大于  $45^\circ$  的，因此无论怎样旋转坐标轴，弧 ACB 上都会有切线斜率的绝对值大于 1 的情况，这时如果直接用小参数法，将对计算产生不利的影响。

如果把整条曲线分成几段，坐标轴也分作几次旋转，如图 3-4 所示，在每次旋转坐标轴时，都使新的横坐标轴与曲线弧段端点连线平行，从每一个局部坐标系去观察每一局部的弧，它们就变的十分平坦了。当每一局部弧段上的倾角不大于  $20^\circ$  时，相应的  $Y'(x)^2$  将不大于 0.13，在这个基础上使用小参数法，更有理由相信能得到较好的效果。

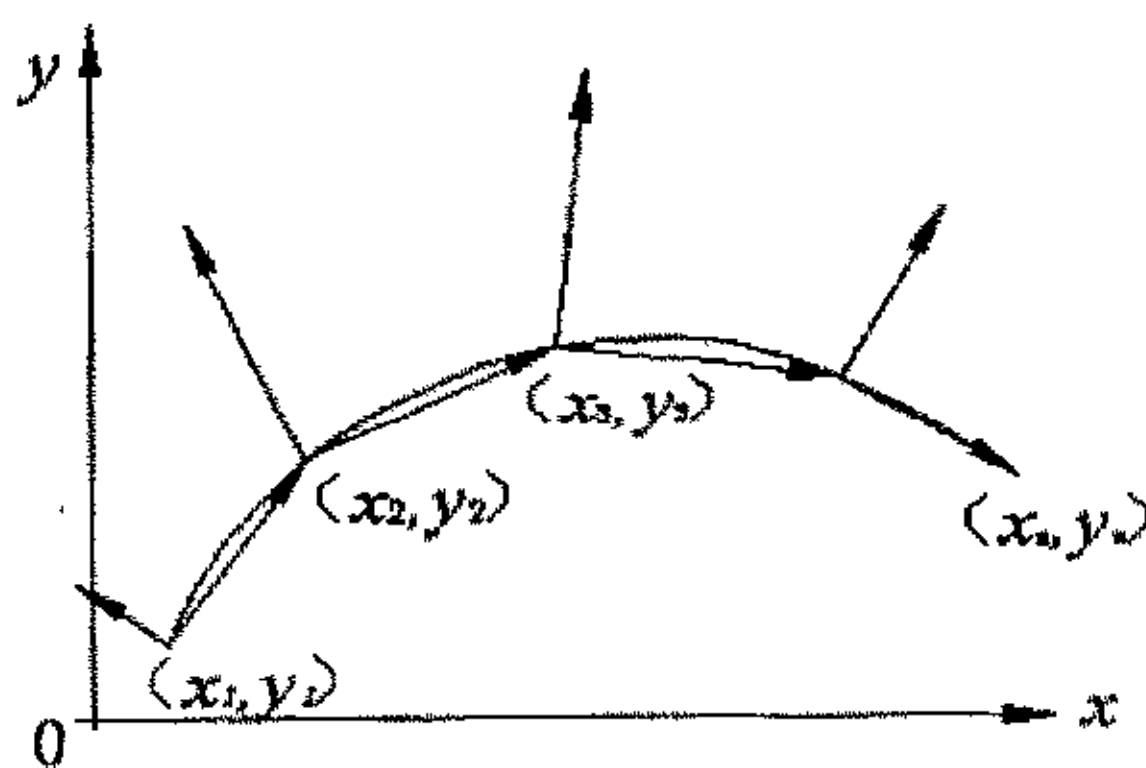


图 3-4 分段旋转坐标系



相关文献上已经证明, 由于坐标轴的旋转, 描述样条曲线的微分方程形式不变。这样在在坐标轴没有旋转时推导的公式就可以直接运用。

在求解具有  $n$  个型值点的样条函数时, 所需的独立的定解条件共有  $n+2$  个, 其中的  $n$  个来自型值表, 另外两个来自端点的导数值条件。当我们使用分段旋转坐标系, 在型线的每一弧段上求解边值问题时, 端点处的曲率条件, 开始时是任意给定的, 这样, 当把各段求得的曲线拼接成一条完整的曲线时, 无法保证曲线的导数在交接点处也是连续的, 拼接出来的曲线在交接点处的导数可能不连续的原因在于各节点处的曲率值是任意选取的。

令  $V'[i, u_1]$  为第  $i$  段曲线在第  $i$  个坐标系下的解在左端点处的导数,  $V'[i, u_2]$  右端点处的导数。当在第  $i-1$  段曲线和第  $i$  段曲线上讨论连接问题时, 如果相连接的两段曲线在二者交点处的切线方向一致, 我们就认为曲线是光滑连接的。

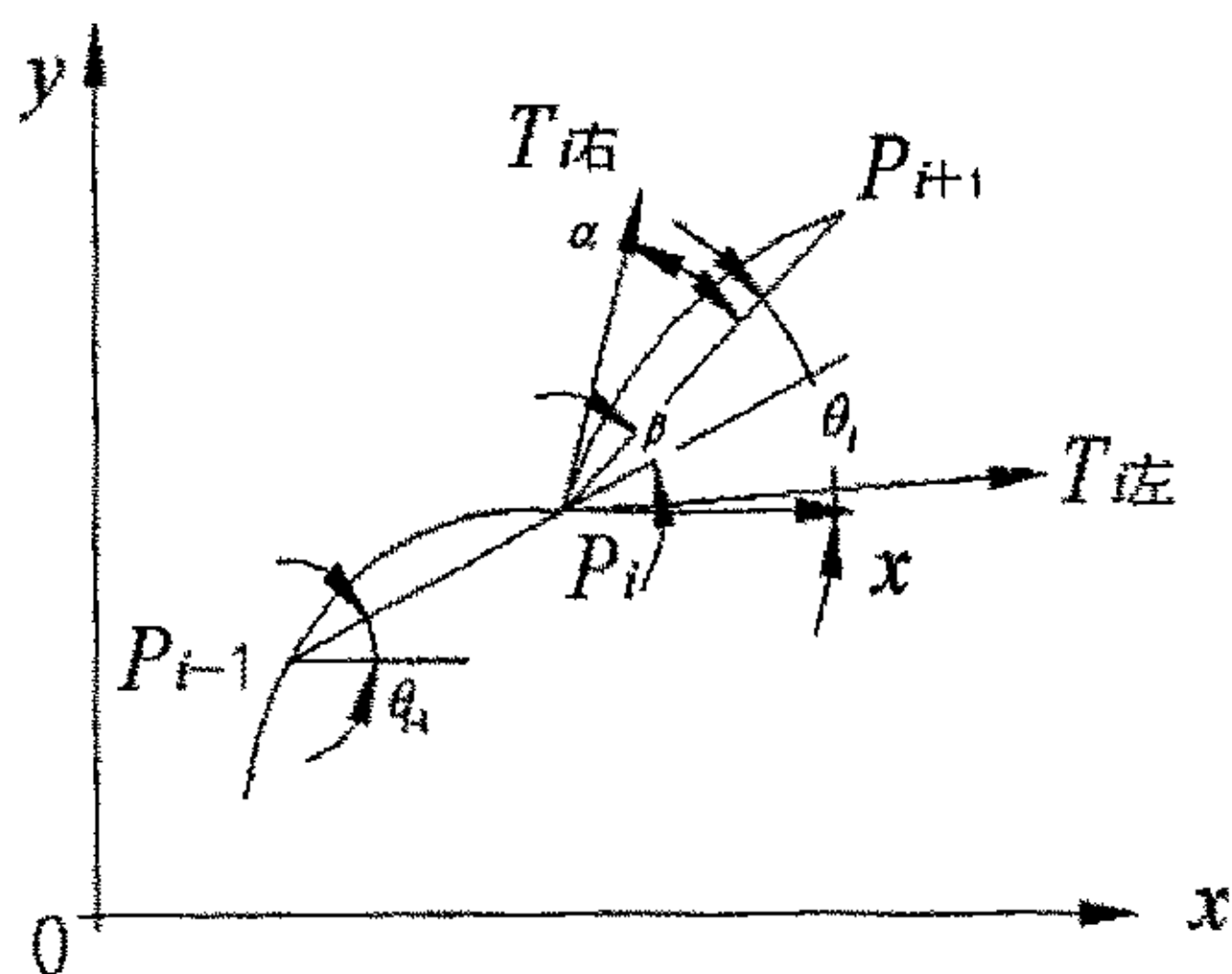


图 3-5  $P_i$  处的左、右切线

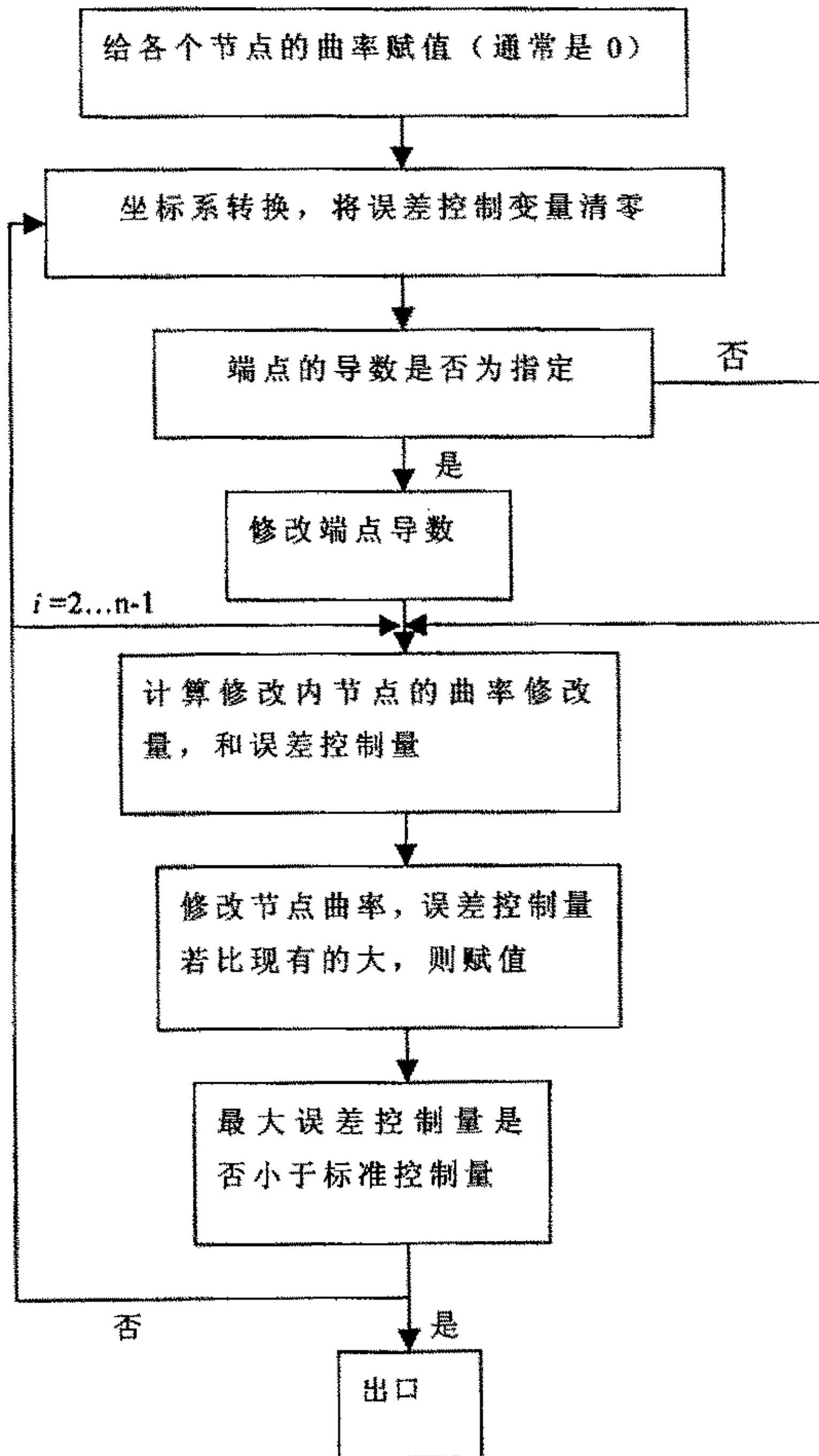
$$V'[i-1, u_2] = \frac{V'[i, u_1] + \lg(\theta_i - \theta_{i-1})}{1 - V'[i, u_1] \lg(\theta_i - \theta_{i-1})} \quad (3-4)$$

就是所谓的连接条件。

对于大多数的型线来说, 在任意选定曲率初值以后, 只需要经过几次迭代, 就能在一定精度下满足光滑连接的条件, 但是对于有的型线来说, 光滑连接的条件不是随着迭代次数的增多而稳定地接近, 而是忽高忽低的不断接近之中, 这种情况表明, 迭代过程仍然具有收敛的能力, 但是这种收敛能力显著地减弱了, 其原因可能是型值点之间矛盾得比较厉害, 这种情况, 我们

将配合后面的弯调方法来克服。

边值问题即计算每个型值点的合理曲率半径，计算流程如下：



## 3.2 单根曲线的光顺<sup>[11, 14, 19, 20, 23]</sup>

### 3.2.1 弯调信息<sup>[16]</sup>

型值表在经过小参数转坐标方法处理后, 将输出一条连续曲线, 由于进行了弧段间交界点处的连接条件的处理, 因此输出的曲线必定具有连续转向的切线, 即其一阶导数也是连续的, 又由于在求解两点边值问题时, 迭代过程中, 交接点处曲率的取值总是一致的, 因此其曲率和二阶导数也是连续的, 输出的曲线准确的通过每个型值点, 因此它在每个型值点上是没有偏离的, 而在型线上不是型值点的地方, 在计算中, 我们都尽可能地把计算精确控制在 1 毫米以内, 因此输出的曲线, 确实是足够光滑而且偏离很小的, 但是还不能说这条曲线已经是“光顺”的了, 型线的“光顺”与否, 还需按型线的设计标准来检验, 这就是要看它是否包含“多余的拐点”。

设计图纸上的船舶型线是船舶制造的主要依据, 在左单根曲线光顺时, 我们必须做到: 使光顺后的曲线具有与设计要求的弯曲形状, 为做到这一点, 就应当把设计要求的曲线形状信息化, 并使之作为输入, 作为控制单根曲线的光顺条件。

描述船型线的弯曲形状的信息是:

- (1) 拐点的个数
- (2) 各拐点横坐标的大致的位置
- (3) 第一个凸凹区间的弯曲方向。

假设设计型线上的拐点个数为  $NI$ , 那么这  $NI$  个拐点就把整个型值区间  $[x_1, x_n]$  分成  $NI+1$  个凸凹相继交替出现的区间, 又设  $m_0$  表示设计型线的最左边的那个凸凹区间的信息, 并规定: 当  $m_0 = 1$  时, 表示最左边的那个凸凹区间是凹的; 而当  $m_0 = -1$  时, 则是凸的。这样, 在知道了第一个区间的凸凹性之后, 其余区间的凸凹性也就很容易确定下来了。

在用原始型值计算样条曲线的同时, 每个型值点处的曲率也就同时计算出来了, 曲率的符号代表了样条曲线的弯曲方向, 当曲率为正时, 表示样条曲线在该点的弯曲方向是下凹的, 否则, 当曲率为负时是上凸的, 要判断每

个型值点处的弯向是否只须看这个点落在设计型线要求的第几个凹凸区间上, 并看算得的曲率符号是否与设计要求相一致, 当其一致时就是弯向正确的, 否则就是不正确的。

例如, 设第  $i$  个型值点落在设计要求的第  $j$  个凸凹区间上, 而  $m_s = \pm 1$  是设计要求的第一个凸凹区间的凸凹信息,  $m_i$  是样条曲线在第  $i$  个型值点处算得的曲率, 则当

$$(-1)^{j+1} m_s m_i > 0 \quad (3-5)$$

时, 则第  $i$  个节点的弯向就是正确的, 而当

$$(-1)^{j+1} m_s m_i < 0 \quad (3-6)$$

时, 第  $i$  个节点的弯向就是不正确的。

### 3.2.3 端点弯向的调整

当端点的弯向不正确时, 将不能使用小挠度模型进行调整, 这是因为端点的型值通常是不允许修改的, 而且在该点的一侧,  $x_{i+1}$  或  $x_{i-1}$  将没有意义。另外, 还需把端点导数值对于型线的弯曲形状的影响考虑进去。下面, 将着重讨论型线的左端点的弯向不正确的情形, 至于右端点, 这里给出其弯调的公式。

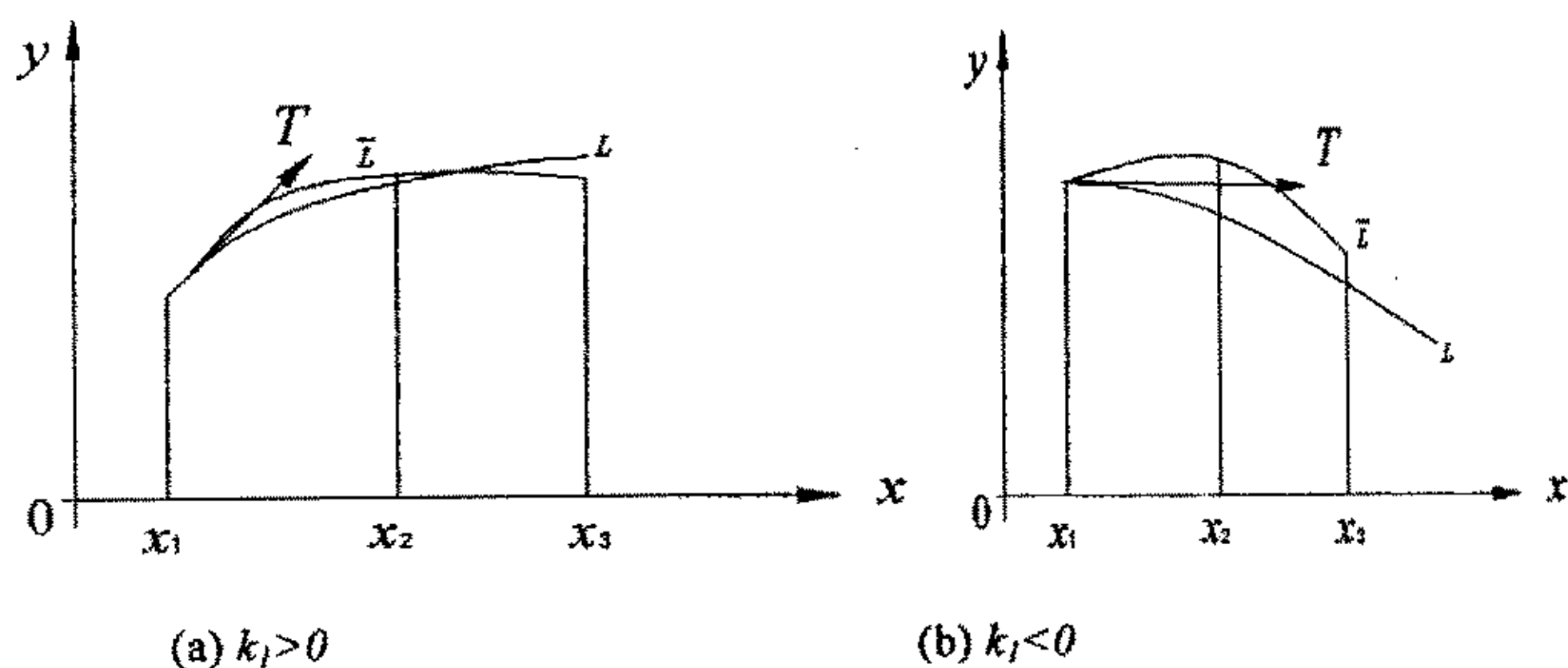


图 3-6  $k_1$  值不同型线比较

如图 3-6 中 a, 根据设计型线的要求, 型线在  $[x_1, x_2]$  上没有拐点, 且设型线的第一个凸凹区间上的信息是凸的, 即  $m_1 = -1$ , 当计算出的样条曲线在左端点  $x = x_1$  处弯向不正确时, 设  $L$  是设计型线的形状,  $\bar{L}$  是计算出的样条曲线的形状, 则此时形状分布的示意情况略如图中(a), 此时  $m_1 > 0$ , 且设在  $x_2$  处的弯向是正确的。在这种情况下, 调整  $x_1$  的弯向的措施主要有两个, 其一是旋转切线, 使样条曲线在  $x_1$  处的切线下侧与切线相切, 例如, 设  $k_1$  是在计算样条前, 现在左端点的给定的导数值, 则在弯调后, 这个新的导数值可以取为:

$$k_1 (1 + \eta) \quad (3-7)$$

这里, 取  $\eta = 0.01$  或  $0.02$ , 对于端点导数值的这样的微小的修改, 自然不会显著地影响船型, 但是对于端点弯调的效果确是明显的。

在调整端点导数值时,  $k_1$  本身的符号也应该顾及, 实际上, 当  $x_1$  的弯向不正确时, 调整端点导数值的统一公式是

$$k_1 [1 + \text{sign}(k_1 m_1) \eta] \quad (3-8)$$

其中  $\eta = 0.01$  或  $0.02$ 。

调整左端点弯向不正确的第二个措施是调整  $y_2$  和  $y_3$  的型值。综合各种根据型值点  $x_1, x_2$  的弯向情况以及拐点的位置, 修改公式如下:

$$\begin{aligned} y_2 &\leftarrow y_2 - \eta_2 (x_2 - x_1) \times \text{sign}(m_1) \\ y_3 &\leftarrow y_3 - \eta_3 (x_2 - x_1) \times \text{sign}(m_1) \end{aligned} \quad (3-9)$$

其中,  $\eta_2$  和  $\eta_3$  的取值, 可依照以下的情况选取:

当样条的右端点处出现弯向不正确时, 在靠近右端点的两个型值点  $x_{n-2}, x_{n-1}$  处的型值  $y_{n-2}, y_{n-1}$  的统一修改公式如下:

$$y_{n-2} \leftarrow y_{n-2} - \eta_{n-2} (x_2 - x_1) \times \text{sign}(m_n)$$



$$y_{n-1} \leftarrow y_{n-1} - \eta_{n-1} (x_2 - x_1) \times \text{sign}(m_n) \quad (3-10)$$

序号	在 $(x_1, x_2)$ 有无拐点	$x_1$ 处弯向是否正确	$x_2$ 处弯向是否正确	$\eta_1$	$\eta_2$
1	无	否	是	0.002	0.0005
2	无	否	否	0.0005	0.0005
3	有	否	是	0.002	0.0015
4	有	否	否	0.0025	0.0015
5	无	是	否	-0.001	0
6	有	是	否	-0.001	0

表 3-1

$\eta_{n-2}$  和  $\eta_{n-1}$  的取值, 可依照以下的情况选取:

序号	在 $(x_{n-2}, x_{n-1})$ 有无拐点	$x_n$ 处弯向是否正确	$x_{n-1}$ 处弯向是否正确	$\eta_{n-2}$	$\eta_{n-1}$
1	无	否	是	0.002	0.0005
2	无	否	否	0.0005	0.0005
3	有	否	是	0.002	0.0015
4	有	否	否	0.0025	0.0015
5	无	是	否	-0.0005	0
6	有	是	否	-0.0005	0

表 3-2

相应导数修改公式为:

$$k_2[1 - \text{sign}(k_2 m_n) \eta] \quad (3-11)$$

其中  $\eta = 0.01$  或  $0.02$ 。

下面是给定型值和拐点的大致位置时, 型线的弯向调整函数部分代码:

```
void JustWX(AcGePoint3dArray& OrenArr, int NI, double *GY, int m0)
{
    const int n=OrenArr.length();
```

```

int i, j, j0, k, nS3, n1, n2;
double k1, ms, d, dy, u1, u2, k2, dm;
double m[20];
    n1=1;n2=1;
    k1=0;k2=-1000000;
    dch(n1, k1, n2, k2, OrenArr, m);
    GY[0]=OrenArr[0].x;
    GY[NI+1]=OrenArr[n-1].x;
    ms=m0;
//调整拐点区间
    for(i=1;i<NI;i++)
    {
        ms=-ms;
        d=1000000;
        for(j=0;j<n-1;j++)
        {
            if(m[j]*ms>0)
                if(hysabs(OrenArr[j].x-GY[j])<d)
                {
                    j0=j;
                    d=hysabs(OrenArr[j].x-GY[i]);
                }
        }
    }
//求解边值问题，得到合理的节点曲率
loop8: dch(n1, k1, n2, k2, OrenArr, m);
    k=0;
    ms=m0;
    for(j=1;j<=NI+1;j++)//对拐点区间循环
    {
        ms=-ms;
        for(i=0;i<n;i++)//在该拐点区间内遍历所有节点
        { //该点是否在区间内
    
```

```

        if((GY[j-1]<=OrenArr[i].x)&&(OrenArr[i].x<=GY[j]))
        {
            if(m[i]*ms<0)//是否需要调整
            {
                nS3=1;
                if(i<=1)//首端有问题
                {
                    // 针对尾部的情况，调整首部
                    .....
                }
                else if(i>=n-2)//尾端有问题
                {
                    //针对尾部的情况，调整首部
                }
            }
            ////内节点有问题
            dy=(OrenArr[i+1].x-OrenArr[i].x)*(OrenArr[i].x-OrenArr[i-1].x)*m[i]/2.7;
            if(hysabs(dy)<0.5)
                dy=0.5*sign(dy);
            while(hysabs(dy)>10)
                {dy=dy/5; }
            OrenArr[i].y+=dy;
            k=1;
        }
    }
}

if(k==0)//符合光顺条件
{ return; }
if(k==1)
    goto loop8;
}

```

### 3.2.4 型线的弯势调整<sup>[16,20]</sup>

型线在经过弯向调整以后,其每一个决定处的弯曲方向就已经符合设计船型的要求了,但是型线的弯曲程度变化的缓急,还未毕是均匀的,当型线的某些地方过于平直,而相邻部位又弯曲得比较厉害时,这种船型还是会给人以不太光顺的感觉,所谓弯势调整,就是要控制曲率变化的均匀性。

弯势调整是在弯向调整的基础上进行的,因此,在作弯势调整时,介于设计型线的两个拐点直接的曲率必定是同号的,设 $(x_i, y_i)$ 和 $(x_{i+1}, y_{i+1})$ 是同属于某一个凹区间上的节点,则其相应的曲率比: $m_i / m_{i+1}$ 或 $m_{i+1} / m_i$ 必定是正数;当 $(x_i, y_i)$ 靠近拐点时, $|m_i|$ 通常更接近于零一些,这时,相邻节点的曲率比 $m_i / m_{i+1}$ 或 $m_{i+1} / m_i$ 的变化范围将比较大,因此,我们把两邻均无拐点和左邻或右邻有拐点的节点,分开来讨论。

设 $x_{i-1}, x_i, x_{i+1}$ 是同属于设计型线要求的某一凸凹区间上的三点,此时, $m_{i-1}, m_i, m_{i+1}$ 是同符号的,在讨论第 $i$ 点的弯势是否合适的时候,我们主要将把 $|m_i|$ 控制在一个一定的范围内,即看 $|m_i|$ 是否比 $|m_{i-1}|$ 和 $|m_{i+1}|$ 之中最小者的 $N$ 分之一还小,如果是这样,就对 $y_i$ 的值进行调整,使能把 $m_i$ 的值控制在一定的变化范围之中,否则就不调整 $y_i$ 的值, $y_i$ 的调整量 $dy$ 可依下面的公式进行计算:

$$dy_i = -(x_{i+1} - x_i)(x_i - x_{i-1})dm/3 \quad (3-12)$$

其中 $dm$ 是 $m_i$ 的调整量,可以按照下列方式计算选取:

当 $|m_i| \geq N \times \max(|m_{i-1}|, |m_{i+1}|)$ 时, 取

$$dm = N \times \max(|m_{i-1}|, |m_{i+1}|) \times \text{sign}(m_i) - m_i \quad (3-13)$$

当 $|m_i| \leq \frac{1}{N} \times \min(|m_{i-1}|, |m_{i+1}|)$ 时, 取

$$dm = \frac{1}{N} \times \min(|m_{i-1}|, |m_{i+1}|) \times \text{sign}(m_i) - m_i \quad (3-14)$$

式中控制系数  $N$  的选取, 视船型型线曲率变化的可能情况而定, 对于小船,  $N$  可以取得稍大一些, 如取 4 或 5 直间, 对于大船, 因型线变化稍平稳, 则  $N$  可以取的小一些, 如取为 2~3 之间。

当  $x_{i-1}$ ,  $x_i$ ,  $x_{i+1}$  不同属于设计型线要求的一个凸凹区间时, 则  $x_i$  或与  $x_{i-1}$  或与  $x_{i+1}$  同属于一个凸凹区间,  $m_i$  的大小将按与其同侧的  $m_{i-1}$  或  $m_{i+1}$  的大小来控制; 当  $x_i$  或与  $x_{i-1}$  或与  $x_{i+1}$  同属于一个凸凹区间,  $x_i$  是比  $x_{i-1}$  或  $x_{i+1}$  比更接近拐点一些, 而拐点处的曲率是零, 因此,  $|m_i|$  应控制在一个不大的范围内, 例如, 当  $x_i$  与  $x_{i-1}$  同侧, 且当

$$|m_i| \geq k |m_{i-1}|$$

时, 就对  $y_i$  的值进行调整, 调整量计算公式仍用 (3-12), 而  $dm$  则可取为

$$dm = k |m_{i-1}| - m_i \quad (3-15)$$

式中  $k$  的值, 小船取 2.5, 大船取 1.5~2。

当  $x_i$  与  $x_{i+1}$  同侧时, 而当需要调整  $y_i$  时, 其相应的  $dm$  亦可为:

$$dm = k |m_{i+1}| - m_i \quad (3-16)$$

弯势调整的更合理更精细的控制方式可以这样来控制: 设计型线的两个相邻拐点之间, 曲率值可能有一个极值利用两个拐点处的曲率值去做它们的线性插值多项式, 设为  $P(x)$ , 选择适当的正数  $\varepsilon$ , 使当

$$P(x) - \varepsilon < m_i < P(x) + \varepsilon$$

时,  $m_i$  的相应的型值就不修改; 否则就修改。这样做自然会使计算程序更复杂一些, 通常在弯向调整以后, 弯势调整的矛盾总是处于相对次要地位, 所以, 在没有特殊需要时, 时无需这样做的, 但有时也有这样的情形, 即型

线上的曲率的变化范围比较大,而在某些局部,曲率的变化又比较剧烈,这时,但靠把式(3-13)~(3-15)中的 $N$ 和 $k$ 选大一些,实际上对某些型线将失去控制意义,否则又不能达到本来控制的目的,这时,利用上述方法则是可取的。

弯调程序出口时,应该使修改过型值的样条对于弯调程序是自然通过的,即修改过型值的样条不再经过修改型值,其弯向和弯势都是正确的。

```
void Smooth(AcGePoint3dArray& OrenArr,int NI,double *GY,int m0)
{
    .....
    //弯向调整部分
    .....
    //进行弯势调整
    if(k==0)
    {
        for(i=1;i<n-1;i++)
        {
            if((m[i-1]*m[i]<0)&&(m[i]*m[i+1]>0))
            {
                if(hysabs(m[i+1])>hysabs(m[i-1]))
                {    u1=m[i+1]; u2=m[i-1]; }
                else
                {    u1=m[i-1]; u2=m[i+1]; }
                if(hysabs(m[i])<0.2*hysabs(u2))
                {    dm=0.2*u2-m[i]; }
                else
                {
                    if(hysabs(m[i])>(5.0*hysabs(u1)))
                        dm=5*u1-m[i];
                    else continue;
                }
            }
        }
    }
```



```

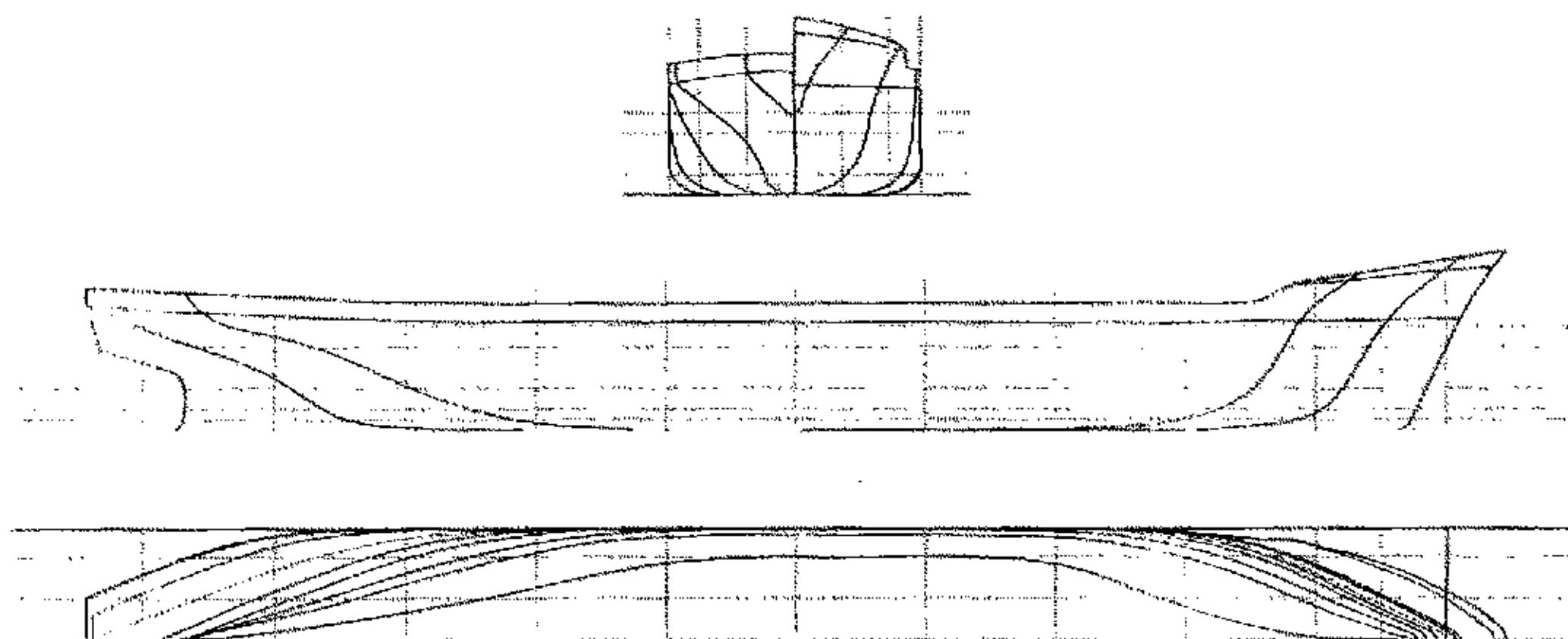
else
{
    if(m[i]*m[i-1]>0)
    {
        if(hysabs(m[i])<2.5*hysabs(m[i-1]))
            continue;
        else
            dm=2.5*m[i-1]-m[i];
    }
    else
    {
        if(hysabs(m[i])<2.5*hysabs(m[i+1]))
            continue;
        else
            dm=2.5*m[i+1]-m[i];
    }
}
dy=-(OrenArr[i+1].x-OrenArr[i].x)*(OrenArr[i].x-OrenArr[i-1].x)*dm/3;
k=1;
if(abs(dy)<0.2)
    dy=sign(dy)*0.2;
while(hysabs(dy)>5)
{
    dy=dy/5;
    OrenArr[i].y=OrenArr[i].y+dy;
}
}
else
    goto loop8;
if(k==1)
    goto loop8;
if(k==0)
{
    return ;
}
}

```

### 3.3 船型总体光顺<sup>[21, 23, 24]</sup>

上面探讨了在给定型值和边界条件下单根型线的光顺过程,所有光顺的曲线最终构成了船舶的光顺曲面,所以单根型线不是孤立的,它的端部条件取决于它在船舶中所处的位置。所以我们必须首先从船舶的总体光顺性出发,对单根型线的端部条件比较集中的部位如平底线、平边线以及水线图中水线首切点构成的首柱线等首先进行光顺,以便能够给后续的单条型线的光顺提供合理的端部条件。至于平底线和平边线的光顺等光顺方法,这里就不在赘述。在它们的光顺之后,对单根型线进行端部条件赋值并进行光顺,整个型线图就完成了。

本文按照上述思路对给定船舶进行了光顺,得到如下的型线图:



3-7 图 型线图

从效果上来看,大致还是不错的。由于目前首尾端部等完全的计算机自动光顺算法还不是很成熟,在水线的首尾部分还是不够完美。不过型线图是在 AutoCAD 中自动生成的,可以对首尾进行必要的手工调整,以达到完全的光顺。

## 第 4 章 在 AutoCAD 中绘制型线图

### 4.1 ObjectARX 及开发环境介绍<sup>[36, 38, 39, 55, 57]</sup>

#### 4.1.1 AutoCAD 二次开发语言及工具介绍

众所周知，作为 CAD 工业旗帜产品的 AutoCAD，伴随着近年来整个 PC 基础工业的突飞猛进，正在迅速而深刻地影响着人们从事设计和绘图的基本方式。从始至今，AutoCAD 均是一种定位于全球各大领域和各类专业的通用微机 CAD 平台软件，其强大生命力在于它的通用性、多种工业标准和开放体系结构，但是，要想使它真正适用于某一领域，如船舶设计，必须对它进行二次开发。目前 Autodesk 公司提供了以下几种二次开发手段：

1) AutoLISP AutoLISP 的全名是 LIST Processing Language，她出现于 1985 年推出的 AutoCAD R2.18 中，是一种嵌入在 AutoCAD 内部的编程语言，是 LISP 原版的一个子集，一直是低版本 AutoCAD 的首选编程语言。它是一种表处理语言，是被解释执行的，任何一个语句键入后就能马上执行，它对于交互式的程序开发非常方便。其缺点是继承了 LISP 语言的编程规则而导致繁多的括号。

2) ADS ADS 的全名是 AutoCAD Development System，它是 AutoCAD 的 C 语言开发系统，ADS 本质上是一组可以用 C 语言编写 AutoCAD 应用程序的头文件和目标库，它直接利用用户熟悉的各种流行的 C 语言编译器，将应用程序编译成可执行的文件在 AutoCAD 环境下运行，这种可以在 AutoCAD 环境中直接运行的可执行文件叫做 ADS 应用程序。ADS 由于其速度快，又采用结构化的编程体系，因而很适合于高强度的数据处理，如二次开发的机械设计 CAD、工程分析 CAD、建筑结构 CAD、土木工程 CAD、化学工程 CAD、电气工程 CAD 等。

3) Visual Java Java 是最早由 Sun 公司创建的一种颇具魅力的程序设计语言，她是针对嵌入系统而设计的。像许多开发语言一样，Java 是一组实时库的集合，可为软件开发者提供多种工具来创建软件，管理用户接口，

进行网络通讯、发布应用程序等。对 AutoCAD 用户和开发者而言, Java 代表着新一代的编程语言, 它主要用于开发出全新的优秀产品。

4) VisualLISP VisualLISP 已经被完整地集成到 AutoCAD 2000 中, 她为开发者提供了崭新的、增强的集成开发环境, 一改过去在 AutoCAD 中内嵌 AutoLISP 运行引擎的机制, 这样开发者可以直接使用 AutoCAD 中的对象和反应器, 进行更底层的开发。其特点为自身是 AutoCAD 2000 中默认的代码编辑工具; 用它开发 AutoLISP 程序的时间被大大地缩短, 原始代码能被保密, 以防盗版和被更改; 能帮助大家使用 ActiveX 对象及其事件; 使用了流行的有色代码编辑器和完善的调试工具, 使大家很容易创建和分析 LISP 程序的运行情况。在 VisualLISP 中新增了一些函数: 如基于 AutoLISP 的 ActiveX/COM 自动化操作接口; 用于执行基于 AutoCAD 内部事件的 LISP 程序的对象反应器; 新增了能够对操作系统文件进行操作的函数。

5) VBA VBA 即 Microsoft office 中的 Visual Basic for Applications, 它被集成到 AutoCAD 2000 中。VBA 为开发者提供了一种新的选择, 也为用户访问 AutoCAD 2000 中丰富的技术框架打开一条新的通道。VBA 和 AutoCAD 2000 中强大的 ActiveX 自动化对象模型的结合, 代表了一种新型的定制 AutoCAD 的模式构架。通过 VBA, 我们可以操作 AutoCAD, 控制 ActiveX 和其它一些应用程序, 使之相互之间发生互易活动。

6) ObjectARX ObjectARX 是一种崭新的开发 AutoCAD 应用程序的工具, 她以 C++ 为编程语言, 采用先进的面向对象的编程原理, 提供可与 AutoCAD 直接交互的开发环境, 能使用户方便快捷地开发出高效简洁的 AutoCAD 应用程序。ObjectARX 并没有包含在 AutoCAD 中, 可在 Autodesk 公司网站中去下载, 其最新版本是 ObjectARX for AutoCAD 2000, 它能够对 AutoCAD 的所有事务进行完整的、先进的、面向对象的设计与开发, 并且开发的应用程序速度更快、集成度更高、稳定性更强。ObjectARX 从本质上讲, 是一种特定的 C++ 编程环境, 她包括一组动态链接库 (DLL), 这些库与 AutoCAD 在同一地址空间运行并能直接利用 AutoCAD 核心数据结构和代码, 库中包含一组通用工具, 使得二次开发者可以充分利用 AutoCAD 的开放结构, 直接访问 AutoCAD 数据库结构、图形系统以及 CAD 几何造型核心, 以便能在运行期间实时扩展 AutoCAD 的功能, 创建能全面享受 AutoCAD 固有命令的新命令。ObjectARX 的核心是两组关键的 API, 即 AcDb (AutoCAD 数据库) 和 AcEd (Auto

CAD 编译器), 另外还有其它的一些重要库组件, 如 AcRX(Auto CAD 实时扩展)、AcGi(Auto CAD 图形接口)、AcGe(Auto CAD 几何库)、ADSRX(Auto CAD 开发系统实时扩展)。ObjectARX 还可以按需要加载应用程序; 使用 ObjectARX 进行应用开发还可以在同一水平上与 Windows 系统集成, 并与其它 Windows 应用程序实现交互操作。

Auto CAD 为我们提供了的多种开发工具为用户和开发者提供了多种新的选择, 使得对 AutoCAD 二次开发和定制变得轻松而容易。比较而言, Autolisp 是一种解释型语言, 运行速度较慢, 同时功能相对有限; 而 ADS 是开发 AutoCAD 的 C 语言设计环境; VBA 的特点是使用简单, 但存在的问题和 Autolisp 类似, 效率不高且功能有限。而 ObjectARX 是全新的面向对象的开发手段, 功能非常强大。ARX 是自 AutoCAD R13 版开始特有的开发环境, 功能强大、运算效率高。通过在 CAD 应用系统的应用表明, 它的许多新特性能实现许多原来开发环境难以实现的功能, 同时还提高了开发效率。因此就综合性能而言, ARX 代表着 AutoCAD 的发展方向, 将成为新一代应用程序开发的首选工具。

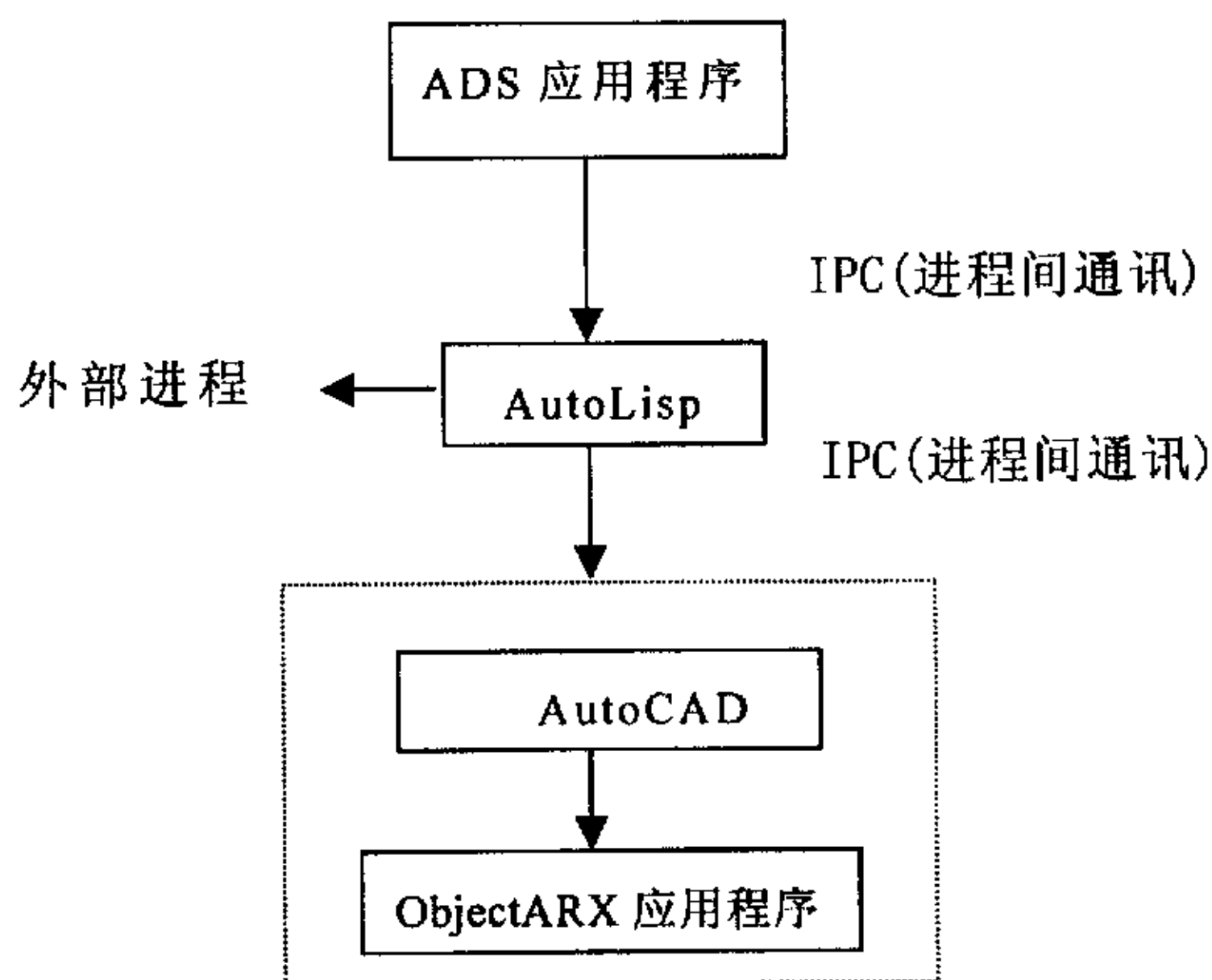


图 3-1 ObjectARX、ADS、AutoLISP 和 AutoCAD 之间的关系

ObjectARX 包含一组动态链接库(DLL), 这些库与 AutocAD 在同一地址空间内运行, 直接利用 Aut0CAD 的核心数据库结构和代码。库中包含一组通用工具, 利用这些工具可以充分利用 AutoCAD 的开放结构, 直接



访问 AutoCAD 的数据库结构、图形系统，以及几何造型核心，以便能够在运行期间实时扩展

AutoCAD 的功能，建立与 AutoCAD 本身的固有命令操作方式相同的新命令。ObjectARX 程序的本质是 windows DLL 程序。AutoCAD 本身是一个典型的 Windows 程序，ObjectARX 程序与 AutoCAD、Windows 之间采用 Windows 消息传递机制直接通信。它以 C++ 语言为基本开发语言，具有 OOP(Object Oriented Program)技术的许多特点，如数据封装、继承、多态性等。ObjectARX 支持 MFC，也就是说，能够利用 MFC 编写功能强大、具有 windows 风格的用户图形界面程序。ObjectARX 还可实现许多其他功能，如支持多文档界面，创建自定义类，与其他编程环境进行交互等。

#### 4.1.2 ObjectARX 程序的基本结构和功能

ObjectARX 程序通常由初始化程序和用户函数两部分组成。

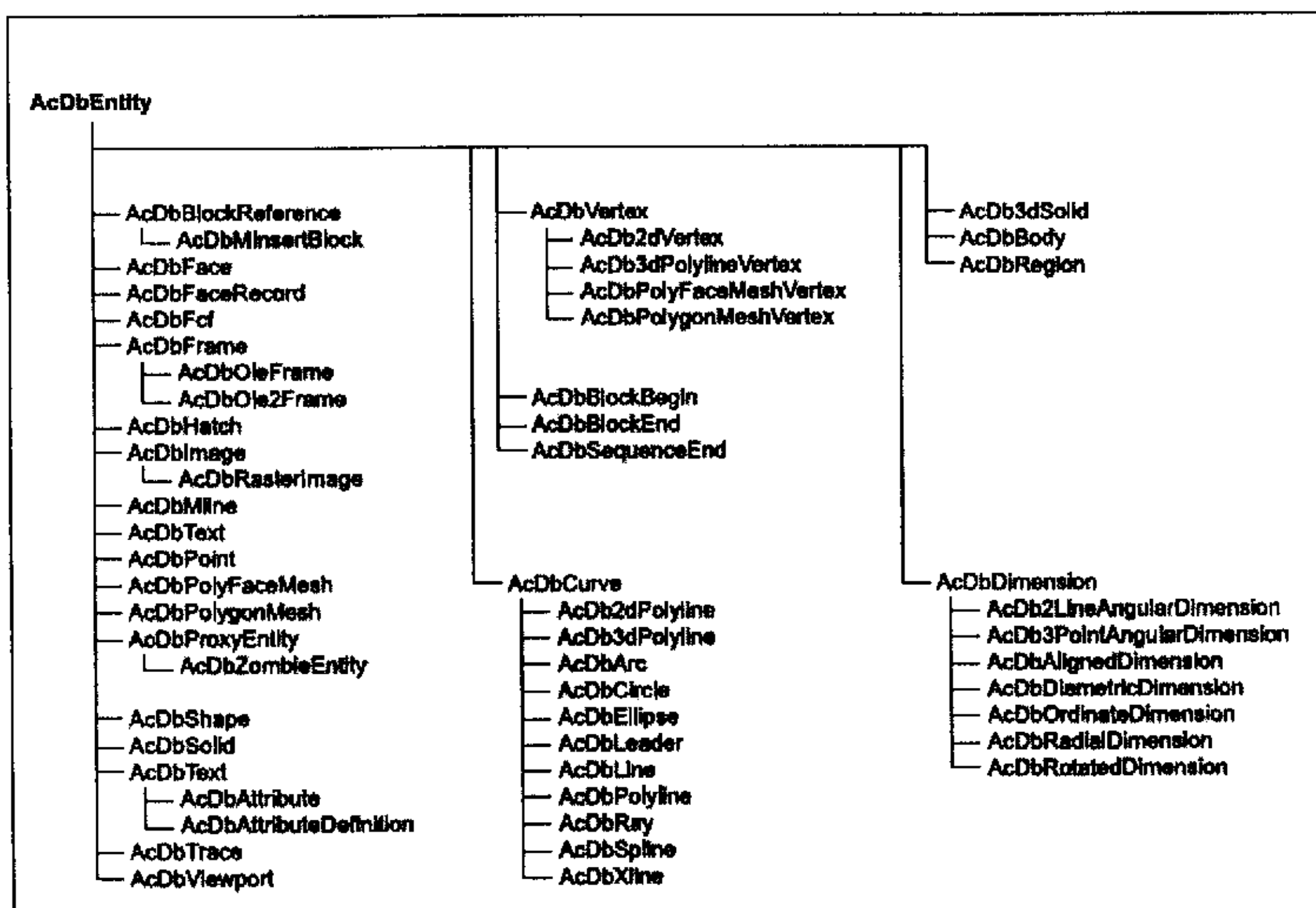
初始化部分主要由头文件(包含语句和外部函数说明及变量定义)、入口函数 `acrxEntryPoint()` 定义、初始化函数定义和卸载 ObjectARX 程序的函数定义四部分组成。由于设计对话框用到了 MFC 类库，因而，在初始化函数定义部分，除了要调用 `addCommand()` 函数注册新的命令以外，还要调用 MFC 类库的入口函数 `DllMain()`。卸载函数 `unloadAPP()` 仍调用 `DllMain()` 释放 MFC 库所占的内存，只是函数参数与加载不同。卸载函数的第二个功能是调用 `removeGroup()` 函数释放在 `addCommand()` 函数定义的命令组。ObjectARX 的初始化程序可作为一种相对固定的模式，在实际应用中仅需要修该 `addCommand()` 函数的调用参数和编写用户函数。

用户函数是程序的主体部分。包括了对话框的定义和激活、及用户设计的其它功能。在先进的 VisualC++ 编程语言的支持下，极大地减小开发人员编程的复杂性，使得编程更加简单、快捷，并能方便、高效的设计具有典型 Windows 风格的 CAD 应用程序。因此，根据各个行业的行业习惯，把不同实体进行归类，编写所需的菜单条和对话框去触发各个实体的属性、函数和方法。

#### 4.1.3 AutoCAD 数据库概述

AutoCAD 数据库(AutoCAD Database，简称 AcDb)用来存储组成





其中，块表(AcDbBlockTable)中存储实体的记录称为块表记录，即所有的实体均存储在块表记录中，所谓实体是指带有图形表现的数据库对象，包括直线、圆、弧线、文字、实心体、区域、样条曲线和椭圆等，它们是从AcDbEntity类派生而来的。尺寸标注样式表、层表、线型表和文字样式表等均用来存储相应的表记录。字典(AcDbDictionary)对象可以包含任何类型的AcDbObject对象，包括其它字典。一个字典对象一般不检查表项的类型，而

MLINE 线型字典只包括 AcDbMlineStyle 对象, 组字典也只包括 AcDbGroup 对象。

### 4.1.3 AutoCAD 数据库对象创建与编辑

#### 4.1.3.1 对象的创建

##### 1. 生成实体对象的方法

在 AutoCAD 数据库中生成一个实体对象有下述四种方法:

- (1) 用 AutoCAD 命令生成一个对象。
- (2) 用 AutoLISP 的 `command()` 函数生成一个对象。
- (3) 用 ADS 的 `ads_command()` 函数生成一个对象。
- (4) 用 ARX 数据库操作函数生成一个对象。

无论采用何种途径生成一个对象, 其结果都是在相应的数据库中写入有关记录。在前三种方式中, 向数据库写记录的操作由 AutoCAD 系统完成。后一种方式在 ARX 程序中直接调用有关函数实现写记录

##### 2. 创建实体对象的过程

在 AutoCAD 数据库中, 一个实体实际上是该数据库中的一条记录, 其存储结构为链表形式。在块表中存有分别指向模型空间和纸空间链表的指针, 在模型空间链表中存有指向存储实体的链表指针。由此看出, 生成一个实体对象的实质是向 AutoCAD 数据库的块表记录中增加一条记录。在 ARX 应用程序中, 创建实体对象的具体过程如下:

- (1) 通过调用 `acdbCurDwg` 类的成员函数 `getBlocklockTable()`, 获得当前图形的块表指针。
- (2) 调用 `AcDbBlockTable` 类的成员函数 `getAt()`, 获得当前图形的块表记录指针。
- (3) 调用 `AcDb` 对象类创建对象, 然后调用 `AcDbBlockTableRecord` 类的成员函数 `appendAcDbEntity()` 将对象写入块表记录中。

#### 4.1.3.2 修改对象属性

在 AutoCAD 中要编辑对象必须找到该对象, 获取对象主要有三种途径: 通过对象句柄、对象标识符和指向对象的指针。对象句柄是 AutoCAD 图形文件中各对象之间的唯一区别, 存在于整个图形文件的生命期; 即只要图形

文件没有被删除,无论该文件是否被打开对象句柄总是存在。在图形文件打开后,数据库中的每一个对象都具有一个区别于其它对象的对象标识符,直到图形文件的关闭或数据库删除。通过数据库打开函数可以获得对象标识符,并返回指向该对象的指针。对象编辑的主要操作内容是对其属性的查询和修改。数据库对象的属性包括共有属性和专有属性两类:以图形形式表示的实体类对象的共有属性主要指颜色、层和线型等;专有属性与特定的对象有关,如直线的起点和端点、圆的圆心和半径等。

对象共有属性的查询和修改比较简单,首先打开指定的对象,获得指向该对象的 `AcDbEntity` 类指针,并利用 `AcDbEntity` 类的成员函数对属性进行查询和修改,最后关闭对象。下面是改变对象颜色和线型的一段代码:

```
AcDbEntity *pEnt;//定义指向对象的指针
AcdbOpenObject(pEnt, eld, AcDb::kForWrite);//打开对象,并获得指向对象的指针
Pent->SetCorlorIndex(newCorlor);//调用成员函数改变颜色
Pent->SetLinetype(“center”);//调用成员函数改变线型为中心线
Pent->close();//关闭对象
```

对象专有属性的查询和修改,要比共有属性的查询和修改复杂些,其基本步骤是:

- 1)打开指定的对象,获得指向该对象的 `AcDbEntity` 类指针;
- 2)利用类型转换函数得到指向该对象某一具体类指针;

3)使用该类的成员函数查询和修改对象属性。因此,只要获得指向某一特定对象的指针,就可以通过“指针变量—>成员函数”的形式,查询和修改对象的属性。正是由于 ARX 应用程序可以直接访问 AutoCAD 数据库对象,因此,其运行速度比 ADS 应用程序快得多。

#### 4.1.4 使用 MFC AppWizard 创建 ObjectARX 应用程序

在 AutoCAD 2000 环境下开发 ObjectARX 应用程序,需要 Visual C++6.0 编译环境和 ObjectARX 2000 SDK 的支持,应用 Visual C++ 6.0 的集成开发环境和 ObjectARX 2000 应用程序向导,能够轻松,方便地创建具有 Windows 风格的 CAD 应用程序。

为了方便 ObjectARX 开发人员,Autodesk 公司提供了 ObjectARX 2000

应用程序向导—ObjectARX 2000 Wizard。ObjectARX SDK for AutoCAD 2000 的 Wizard 是为 Visual C++ 6.0 定做的, 而且集成到 Visual C++ 6.0 的编程环境之中, 应用 ObjectARX 应用程序向导, 可使开发人员避免直接进行“设置编译器选项”、“设置连接选项”、“添加 C++ 源文件”、“添加建立 ObjectARX 程序的框架代码”、“创建 DEF 文件”等烦杂的步骤, 只需很简单的几步, 就可完成 ObjectARX 程序框架。在下面所述的创建应用程序框架的步骤中, 可看出 ObjectARX 2000 应用程序向导给开发人员带来的方便。

ObjectARX 嵌入工具(即 ObjectARX addIn)是与 ObjectARX 应用程序向导配合使用的。在 Visual C++ 6.0 安装 ObjectARX add.In 后就会出现 ObjectARX 嵌入工具条, ObjectARX 嵌入工具条提供了很多功能, 如选择 ObjectARX 包含文件、定义新的 AutoCAD 命令、选择入口点消息、创建新的自定义类、ObjectARX MFC 支持、ObjectARX 在线帮助等。应用 ObjectARX 嵌入工具可减轻开发人员的编程工作量。

利用 ObjectARX 2000 应用程序向导建立 ObjectARX 应用程序的步骤如下:

1) 选择 File/New 弹出“New”对话框, 选择 ObjectARX AppWizard 2000, 再输入项目名, 单击 OK 按钮。

2) ObjectARX 应用程序向导第 1 步的 ProjectType 中选“ObjectARX(AutoCAD extension)”, Additional 中选中“Use WC”复选框, 再单击“Finish”

#### 4.1.5 类的简单介绍<sup>[27,28]</sup>

C 语言中的结构体可以把相关联的数据元素组成一个单独的统一体, 例如可以设置一个结构体方桌 Table, 包括桌子的高 h, 宽 w, 长 l 若干属性, 结构体如下:

```
struct Table{
    float h;
    float w;
    float l;
};
```

可以如下定义 A, B 两个桌子:

```
struct Table tableA, tableB;
```

然后就可以通过分别设置对象 tableA 、tableB 的数据成员 h, w, l 来“定制”这两个桌子，而对于某个圆桌，就不能用结构体 Table 的对象来定义，而必须重新定义结构体 CirTable 来表示圆桌。

C++中的类的概念与 C 语言中的结构体的概念有相似的地方，差别在于类有封装性和继承性，同时类有成员函数，这样类的运用比结构体要灵活的多，这也是面向对象编程和过程编程的本质区别。对于上面的桌子的例子，这里可以加一个求桌面面积的成员函数 GetArea()用类来表示如下：

```
class CTable{
    float h;
    float w;
    float l;
    float GetArea()
};
```

看起来除了把关键字 struct 改成了 class，变化似乎不大，但是对于有抽屉的方桌，除了抽屉，普通方桌的性质它都有，我们设计带抽屉方桌时就不用从头开始设计，只要从普通方桌类继承那些共有的性质，然后加上自己的性质即可。抽屉方桌可以如下设计：

```
class CdrawTable: public Ctable
{
    float dl;
    float dw;
}
```

数据成员 dl, dw 分别代表抽屉的长和宽，是普通方桌所不具备的性质，但是它的类对象也可以象普通方桌对象那样求桌面的面积，例如有如下定义：

```
CdrawTable drawtable;
```

在设置了对象 drawtable 的 h, w, 和 l（这 3 个属性继承于普通方桌类 CTable），就可以用语句 drawtable.GetArea()来求抽屉方桌的桌面面积。当然，抽屉方桌也可以对抽屉进行普通方桌所不具备的操作。由于 CdrawTable 是从 Ctable 继承而来，我们称 CdrawTable 是 Ctable 的子类，而 Ctable 是 CdrawTable 的父类。



## 4.2 型线图相关类的定制<sup>[36, 38, 42, 43, 56]</sup>

上面介绍了介绍了开发工具和环境, 这里就开始绘制具体的船舶型线了. 在 Visual C++ 的开发环境是以类为中心的, 程序所有的功能和实现都在相关类的成员函数中实现, 所以型线的绘制实际上是类的定制过程。

型线图是由一条条型线构成的, 先从当单条型线的绘制说起。根据前面提供的 B 样条理论, 给定一组型值和曲线两端的切矢条件, 就可以用 B 样条模拟一条曲线, 有的型线比较规则, 如纵剖线图图中的主甲板线, 整条型线可以用一根 B 样条曲线来拟合; 而有的型线就不那么规则, 如方尾船的水线图, 最后端如果是方尾则是一条直线段, 最前端是一段圆弧线, 如果整条曲线都用一根 B 样条来拟合的话, 利用前面的节点控制手段, 也可以拟合, 但是控制和计算都很复杂, 效果也难以保证, 这里就用一种更灵活的方法, 对于方尾船, 中间的一段连续曲线, 就用 B 样条拟合, 尾部的直线就直接用直线段来表示, 首端的圆弧也直接用圆弧., 这样尽管在插指处理上多了一些控制, 但是计算比较简单, 精度也更好控制一些。

### 4.2.1 绘图基类<sup>[46,47]</sup>

对于型线图中为数众多的曲线, 按照各自的特点分为好几类以便于分开处理, 但这些曲线都有一些相同点, 如创建时对图形数据库的操作几乎都是相同的, 还有插值等运算等也都有通用性。这样我们就可以把这些相同的部分抽象出来, 总结和定制出一个基类 CDraw, 它最重要的功能就是操作图形数据库, 具体说来, 就是根据某线条的定义数据先创建一条 AcDbSpline 或其他线条对象, 然后找到并打开图形数据库的块表, 接着打开模型空间记录, 将设置好属性的线条对象加入模型空间记录, 最后关闭相应的对象。CDraw 的派生类需要绘制时, 只要调用从 CDraw 继承来的成员函数即可, 这样通过充分利用类的继承特性使程序得到了简化。当然, 基类 CDraw 还有设置图层等功能。CDraw 的类说明如下(部分):

```
class CDraw
{
public:
    CDraw();
```



```

virtual ~CDraw();
AcGePoint3d GetY(AcDbObjectId entId,double x);
protected:
void createNewLayer(char *LayerName);
AcDbObjectId createLine(AcGePoint3d startPt, AcGePoint3d endPt);
void changColor(AcDbObjectId entId, int newColor);
AcDbObjectId createSpline(AcGePoint3dArray &PtArray);
AcDbObjectId drawShipSpline(AcGePoint3dArray pt_array);
.....
};

```

函数 GetY() 可以根据横坐标 x, 插值得到曲线上的纵坐标 y; createNewLayer() 可以创建图层, 图层的名字由输入的字符串来确定; 几个重载函数 createSpline() 用来创建样条, 根据指定条件的不同分别选用相应的函数; 以下是其中一个 createSpline() 的实现:

```

AcDbObjectId CDraw::createSpline(AcGePoint3dArray &PtArray)
{
    int order = 4;
    float fitTolerance = 0.0; //拟合精度
    AcDbSpline *pSpline=new AcDbSpline(PtArray,order,fitTolerance);
    //获得当前图形数据库的块表
    AcDbBlockTable *pBlockTable;
    acdbHostApplicationServices()->workingDatabase()->getSymbolTable(
pBlockTable,AcDb::kForRead);
    //获得符号表中的模型空间块表记录指针, 用于添加对象
    AcDbBlockTableRecord *pBlockTableRecord;
    pBlockTable->getAt(ACDB_MODEL_SPACE,pBlockTableRecord,AcD
b::kForWrite);
    pBlockTable->close();
    //将样条添加到模型空间块表记录中
    AcDbObjectId splineId;
    pBlockTableRecord->appendAcDbEntity(splineId,pSpline);
    //关闭块表记录指针和样条对象指针
    pBlockTableRecord->close();
    pSpline->close();
}

```

```

//返回样条 Id 号
return splineId;
}

```

#### 4.2.2 线型类<sup>[24,35,44]</sup>

有了上面的曲线基类 CDraw,下面就可以根据各种线型的特点由基类派生出需要的各种型线类。

对于普通船型船舶的型线图,水线在半宽水线图中表现为曲线,在另两副图中表现为直线,而纵剖线在纵剖线图中表现为曲线,在横剖线图和半宽水线图中为直线。类似的,横剖线在横剖线图中表现为曲线,却在半宽水线图和纵剖线图中是直线。与水线和纵剖线不同,主甲板线、首尾楼甲板线、舷墙顶线在纵剖线图、半宽水线图、横剖线图中都表现为曲线。在这里利用面向对象的原则,定义 CMaindeck 代表主甲板的类,由于首尾楼甲板线和主甲板线有较多的相同之处,这里就不再为他们定义新的类,也用 CMaindeck 来代表他们。定义 CWaterLine 代表半宽水线图上的水线,定义 CBodyline 代表横剖线图上的横剖线等。这些类的对象都实际的代表一条相应类型的曲线,在三副图中,这些类代表的曲线都要实际的绘出,而对于纵剖线图上的水线,半宽水线图上的纵剖线以及横剖线图上的水线和纵剖线,由于都是直线,就可以采取比较灵活的手段,经过简单的计算可以确定位置并以直线的形式方便的绘出,因此不为这些“直线”定制专门的类来对应,也不用上面的类来代表,这样可以简化计算,提高绘图速度。

这些线型类都是 CDraw 的派生类,根据其数据成员之一的 ptArray (是一个三维点的集合对象)提供的数据,调用从其父类继承来的成员函数 Draw() 即可完成自身的绘制。下面详细介绍其中代表水线图中水线的水线类。

##### 1) 水线类 CWaterLine.

```

class CWaterLine :    public CDraw
{
public:
    AcGePoint3d      Origin;// 基点
    double Height;//水线高度
    double aCircle;//尾端圆弧半径
    double fCircle;//首端圆弧半径

```

```

AcGeVector3d vecF;//首部切点的矢量
AcGePoint3d qdF;//首部圆弧切点
AcGeVector3d vecA;//尾部切点的矢量
AcGePoint3d qdA;//尾部圆弧部切点
    AcGePoint3d pt1,pt2; //曲线端点
    AcGePoint3d pt[30]; //站点
BOOL IsDesignT; //设计水线的标志
double Section_X[5][2];
double SectionHalfWide[5];
AcGePoint3dArray pt_array;//型值点集合
AcDbObjectId m_nWaterLineSplineId;//曲线句柄
void Clear();
void SetMainData(AcGePoint3d origin, double height);
void GetSectionX(AcGePoint3dArray pt_array);
void Draw();
CWaterLine();
virtual ~CWaterLine();
}

```

创建水线时，首先设置水线的首尾端点数值 pt1, pt2 和水线上的型值点数组 pt，首尾端圆弧半径 fCircle, aCircle 然后就可以通过计算首尾切点的位置来创建水线。其中最重要的是函数 Draw()，它的调用会在图纸上绘制出所需要的型线。下面是它的原型函数：

```

void CWaterLine::Draw()
{
    int n=pt_array.length();
    if(n>0)pt_array.removeSubArray(0,n);
    AcGePoint3d tempPt;
    AcGePoint3d tempPt1;
    AcGePoint3d tempPt2;
    AcGePoint3d qpt;
    tempPt.set(0,0,Origin.z);
    tempPt1.set(0,0,Origin.z);
    tempPt2.set(0,0,Origin.z);
}

```

```

qpt.set(0,0,Origin.z);
//尾端点
tempPt.x = pt1.x + Origin.x;
tempPt.y = pt1.y + Origin.y;
tempPt.z = pt1.z + Origin.z;
pt_array.append(tempPt);
//站点
for(int n=5; n<30 ; n++)
    if(pt[n].x == -999)break;
n--;//n 指向最后一个有效型值点
for(int i=0; i<n; i++)
{
    if(pt[i].y == -999)continue;
    tempPt.x = pt[i].x + Origin.x;
    tempPt.y = pt[i].y + Origin.y;
    tempPt.z = pt[i].z + Origin.z;////hys
    pt_array.append( tempPt );
}
pt_array.append(qdF);
//首端点
bool Optmal(AcGePoint3dArray& OrenArr,double ss,double ee);
//绘制之前进行光顺
double endxl;
endl=vecF.y/vecF.x;
m_nWaterLineSplineId=createSpline(pt_array,AcGeVector3d(0,0,0),vec
F,4,0);
if(design_Flag)
    changColor(m_nWaterLineSplineId,1);
if(pt_array.first().y == Origin.y)return;
////绘制前面的一段圆弧
AcDbObjectId arcId;
tempPt1.x=pt2.x-fCircle;
tempPt1.y=Origin.y;
arcId=createArc(tempPt1,fCircle,0,qdF.z);

```

```
.....,
}
```

绘制之前，先创建一个型值点集合 `pt_array` 并清空，这个集合用来存储经过过滤的型值点。然后就是对 `pt1`, `pt2` 和数组 `pt` 中的型值点进行过滤，如果是有效的型值点，则加入集合以便绘制。加入完毕，调用光顺函数进行光顺，然后根据首尾的圆弧是否有圆弧，若有圆弧则求出切点和该点的切矢，有了这些条件就可以调用 `createSpline()` 将这一段曲线绘制出来，然后再绘制首圆弧和尾部一小段直线端，这样就构成了整条水线并在图纸上显示出来。

2) 甲板线类 `CmaindeckLine`. 甲板类代表着型线图的非平面曲线，包括所有的主甲板、艏艉楼甲板边线、舷墙顶线和折角线。由于这几类型线都在纵剖线图，半宽水线图以及横剖线图中都存在而绘制起来有相当的差异，比如水线图中可能需要绘制首尾圆弧而在纵剖线和横剖线图中则无需绘制，因此甲板类的图形绘制函数需要有参数来进行识别该曲线是属于哪一个型线图以便绘制。

3) 站线类 `CstationLine`. 代表横剖线图站线。

4) 轮廓线类 `CkeelLine`. 代表纵剖线图轮廓线，它的赋值函数检索每一条水线、甲板线、舷墙顶线以及艏艉楼甲板线的端点，按照顺序连接起来构成。

### 4.2.3 图形类<sup>[24]</sup>

有了上面线型类定制的基础上，现在定制分别代表纵剖线图、半宽水线图、横剖线图类 `CProfilePlane`、`CWaterPlane` 和 `CBodyPlane`，每个类包含其特定的线型对象，前面提到过由于主甲板、首位楼甲板线和舷墙顶线在每个图中都曲线形式，每个图形类都包含这 3 个数据成员，此外，`CWaterPlane` 还包含有代表水线 `CWaterLine` 的数据成员，而 `CBodyPlane` 还包含有代表横剖线 `CBodyLine` 的数据成员 `CProfilePlane` 包含代表纵剖线 `CStationLine` 的数据成员等。

每个图形类的重要功能就是对该图形中的各条型线设置数值，调用每条曲线的绘制函数完成相应型线图的绘制。对各个线条对象设置型值的功能是成员函数 `SetValue()` 来完成的，该函数操作型值表，找到每条曲线在型值



表中的位置，依次给每条曲线赋值。下面介绍最重要的图形类——代表水线图的 CWaterPlane。

```
class CWaterPlane : public CDraw, public CData:
{
//数据成员
public:
    AcGePoint3d    Origin;//原点
    int waterNum;
    AcGePoint3dArray hysInsarr;
    AcGePoint3dArray hysWLarr[20];
    BOOL m_nDrawFlag;
    CWaterLine waterline[20];//各条水线
    CMainDeckLine maindeckline;//主甲板线
    CMainDeckLine afterSheerline;//尾舷樯顶线
    CMainDeckLine forwardSheerline;//首舷樯顶线
    void GetSection_WL_X();
    double Section_WL_FX[5][20];
    double Section_WL_AX[5][20];
    AcGePoint3dArray SectoinLinePt_array[5];
    void SetSheerValue();//设置舷樯数值
    void SetMainDeckValue();//设置主甲板数值
    void SetWaterLineValue();//设置水线数值
    void SetValue();
//插指函数
public:
    void InsWL(double wh);//插指并绘制高度为 wh 的水线
    void InsZPCrs(double wth);//绘制纵剖线
    void InsCrs(AcGePoint3dArray& Isrray,double x);
    void DrawMainDeck();
    void DrawWaterLine();
    void DrawSheer();
    void DrawingGridLine();
    void Lable();
```



```

void BuildSectionLinePt_Array();
void Draw();
CWaterPlane();
virtual ~CWaterPlane();
}
    
```

上面的注释标明了相关数据成员和成员函数的作用。绘图时 CWaterPlane 的对象先对各条型线设置型值,然后调用 draw()就可以绘制整个水线图,由于有了前面线型类的支撑,这里 draw()函数就变的很简单,只是依次调用每条水线和主甲板线、首尾甲板线的 draw()函数即可。下面是函数 SetWaterLineValue()。

```

void CWaterPlane::SetWaterLineValue()
{
    AcDbSpline *pSpline;
    double R,Cx,Cy,x,y,alpa,beta;
    AcGePoint3dArray cirarry,Carry,qdarr;
    Carry.setLogicalLength(0);
    cirarry.setLogicalLength(0);
    qdarr.setLogicalLength(0);
    AcGeDoubleArray xarray;
    xarray.setLogicalLength(0);
    for(int i=FirstWL; i<=EndWL; i++)
    {
        if(i == DesignWL)
            waterline[i-FirstWL].design_Flag = TRUE;
        waterline[i-FirstWL].SetMainData(Origin,m_pTable->value[i][0]);
        //起始点
        waterline[i-FirstWL].pt1.x = m_pTable->value[i][1];
        waterline[i-FirstWL].pt1.y = m_pTable->value[i][2];
        waterline[i-FirstWL].pt1.z = m_pTable->value[i][0];
        //站上的点
        for(int j=FirstStation; j<=EndStation; j++)
        {
            double tempValue;
        }
    }
}
    
```

```

        tempValue = m_pTable->value[i][j];
        waterline[i-FirstWL].pt[j-FirstStation].x=
            m_pTable->value[StationX][j];
        waterline[i-FirstWL].pt[j-FirstStation].y = tempValue;
    }
    //终止点
        waterline[i-FirstWL].pt2.x = m_pTable->value[i][EndStation+2];
        waterline[i-FirstWL].pt2.y = m_pTable->value[i][EndStation+1];
    //首尾圆弧半径
        waterline[i-FirstWL].fCircle = m_pTable->value[i][EndStation+3];
        waterline[i-FirstWL].aCircle = m_pTable->value[i][EndStation+4];
    }

```

水线图类是最重要的图形类，除了绘制自身的水线及其他线条外，实际上纵剖线图上的纵剖线是由水线图进行插值得来。

#### 4.2.4 管理类 CShip

类 CShip 担负上面各个类的管理工作，包括对数据文件的读取与保存，提供船舶的主要参数，它的三个数据成员 m\_WaterPlane， m\_BodyPlane 和 m\_ProfilePlane，分别代表型线图上的水线图，横剖线图和纵剖线图。其类原型如下。

```

class CShip :    public CData,    public CDraw
{public:
    void Connect(CTable* pTable);
    CWaterPlane m_WaterPlane;
    CProfilePlane m_ProfilePlane;
    CBodyPlane m_BodyPlane;
private:
    double Hr;
    double WL_Length;
    int WaterLineNum;//不含船底线
    int StationNum;
    double B;

```

```

double T;
double Lpp;
double **value;
void clear();
CArray< double ,double&> arrWLHeight;
CArray< double ,double&> arrPLWidth;
void Label();
void BuildValueTable();
void CheckShipData();
BOOL SaveFile();
BOOL Disconnect();
BOOL DrawGrid();
BOOL DrawShip();
public:
    Cship();
    virtual ~CShip();
.....
}

```

其绘图函数

```

void CShip::DrawShip()
{
    m_BodyPlane.Draw();
    m_ProfilePlane.Draw();
    m_WaterPlane.Draw();
}

```

显得非常“简洁”，它分别调用这三个图形各自的绘图函数，由它们分别完成自身的绘制，这就充分体现了面向对象程序设计的优势。调用完毕，整条船的型线图就绘制出来了。

#### 4.3 型线图的插值<sup>[31, 33]</sup>

型线图绘制出来后，就可以根据需要进行插值了。目前为止，我们绘制的型线图称之为原始型线图，说它原始是由于它是根据型值表中现有的数据

绘制出来的，而实际生产中需要大量的非原始数据，例如对船舶进行总布置设计，需要知道某个甲板的情况而型线图中并没有提供这个甲板的情况，这就需要对水线进行插值得到该甲板位置的水线轮廓；肋骨的加工需要知道每根肋骨的型值，而型线图中只有有限个站的横剖线图，这就需要就每个肋位对横剖线图进行插值。在前面第二章中详细探讨了单条型线的插值算法及过程，这里就用上述算法对型线图进行插值。

前面介绍型线类时提到过，在每根线条绘制之前会对赋予它的型值进行筛选，把合理的型值点保留在一个样点集合里，然后依据这个集合绘制线条。这些有效样点集合也是我们进行插值的依据。

### 1. 插值横剖线

插值横剖线比较简单，对于给定的船中距，对水线图上的每一条水线包括船底线进行插值，然后对主甲板、艏楼甲板或艉楼甲板、折角线、舷墙顶线分别进行插值得到一系列的点，这些点就是插值横剖线的型值点。对于有折角线的船舶，折角线以上为直线段，光顺连接只限于折角线以下的部位。另外插值横剖线的下端点注意设置切线方向以与船底相切。部分代码如下：

```
void CWaterPlane::InsCrs(double x)
{
    //x 为船的纵向位置
    AcGePoint3dArray crsarray;
    for(i=0;i<nWL;i++)
        pSpline[i]=new cDbSpline(hysWLarr[i],order,fitTolerance);
    AcGePoint3d pa,pb;
    int n= crsarray.length();
    if(n>0) crsarray.removeSubArray(0,n);
    for(i=0;i< nWL; i++)
    {
        if(hysWLarr[i].first().x>x||hysWLarr[i].last().x<x)//越界
            continue;
        else
        {
            ArrInsX2Y (hysWLarr[i],pSpline[i],x,pa);
            if(x<0)//船尾部，在横剖线图的左边
```

```

    pb.x=-pa.y-B;
    else//船尾部，在横剖线图的右边
    pb.x=pa.y+B;
    pb.y=hysWLarr[i][0].z+2*H;
    crsarray.append(pb);
    }
    }.....
    }
    
```

代码中，crsarray 是用于存放型值点的集合，插值函数 ArrInsX2YPR 可以对型值点集合进行插值。下面是对某船进行步长为 1m 的横剖线插值：

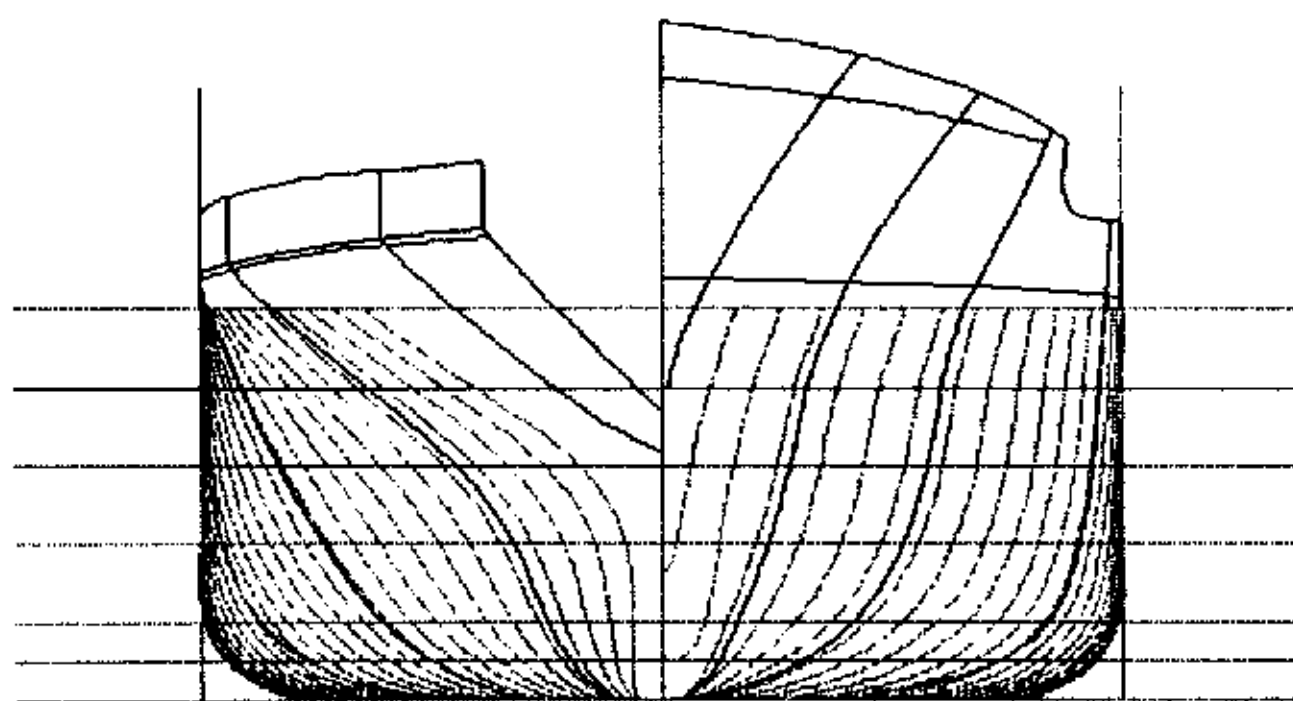


图 4-3 插值横剖线

其中深色线条是原始型线，在首尾端点由于作为插值依据的水线数量较少，导致插值偏差比较大，需要进行一些人工调整。

## 2. 插值水线

1). 主体型值点的确定. 对于给定的某个吃水  $d$ ，欲得到该吃水下的水线，还是归结到要先插值求出这个水线上的各个点，这里的思路是，沿船长方向取某一个纵向位置  $x$ ，按上述插值得到该位置的一条横剖线，然后对这条横剖线按照吃水  $d$  进行插值，得到纵向位置为  $x$ ，吃水为  $d$  的点，也就是我们想求的插值水线上的一个点，然后进行一系列的船长方向的插值，得到一系列的点，它们是插值水线的主要型值点。

上述插值在船体中部大部分区间插值都很正常，但是越往船体的端部，某个纵向位置的所能插得的横剖线型值点就越少，这些部位用 B 样条拟合已经意义不大，用普通的多项式插值就可以了。

2) 首尾端点的确定。插值水线的首尾端点的纵向位置可以根据吃水高度

从纵剖线图中插值轮廓线得到。半宽水线图有一个保存首部圆弧切点的集合和圆弧半径的集合，对圆弧半径根据吃水高度进行插值得到插值水线的首部半径，对圆弧切点根据纵向位置插值得到了插值水线的切点。尾部的出来与此类似。

将首尾端点和前面大量插值得到的型值点光顺连接，便得到了所需要的水线。

### 3. 插值纵剖线

对半宽水线图中水线，它与某一宽度纵剖线通常有前后两个交点。欲求插值某宽度下的纵剖线，首先对水线图中的水线、甲板线及舷墙顶线等根据宽度反向插值得到前后两组型值点，这便是该宽度下的纵剖线的型值点，按要求连接便得到所求的纵剖线，这里不再赘述。



## 第 5 章 绘图界面及数据处理

前面几章介绍了相关的理论、计算过程以及实现的途径，它们构成了本程序的计算内核。由于本程序是绘图程序，绘图时必然涉及到绘图的界面的问题。本章就绘图的操作界面简单介绍，并展示了船舶型线图的绘制过程。当然绘图本身不是本程序的最终目标，还需要将插值得到的大量数据进行存储，然后可以以这些数据为依据，进行数据的查询。

### 5.1 型线图的绘制<sup>[37, 41]</sup>

#### 5.1.1 绘图程序的加载

在 AutoCAD 界面下，输入命令（这是 AutoCAD 的系统命令）arx，然后在命令提示符下键入字母“L”以加载 arx 文件，通过浏览器找到本程序生成的 ship.arx，按照提示加入即可。这样就可以使用船舶型线绘制的专用命令了。

实际上还能实现 arx 程序的自动加载，在 AutoCAD2002 程序目录下有一个 ACAD.RX 文件，用文本方式把 arx 程序的全路径和文件名输入这个文件即可。这样每次打开 AutoCAD，这些专用命令就直接可以使用了。

#### 5.1.2 型值的录入

程序加载完毕，然后在 AutoCAD 中输入命令“input”，会弹出如图 5-1 所示对话框：

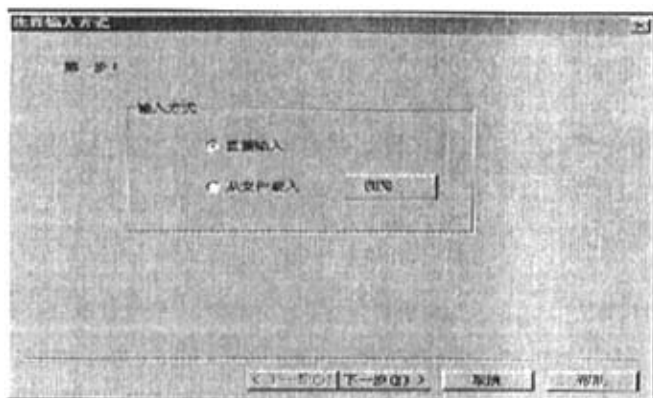


图 5-1

型值数据的录入有两种方式，首先就是直接录入，在对话框中选中“直接输入”，得到下图所示对话框：

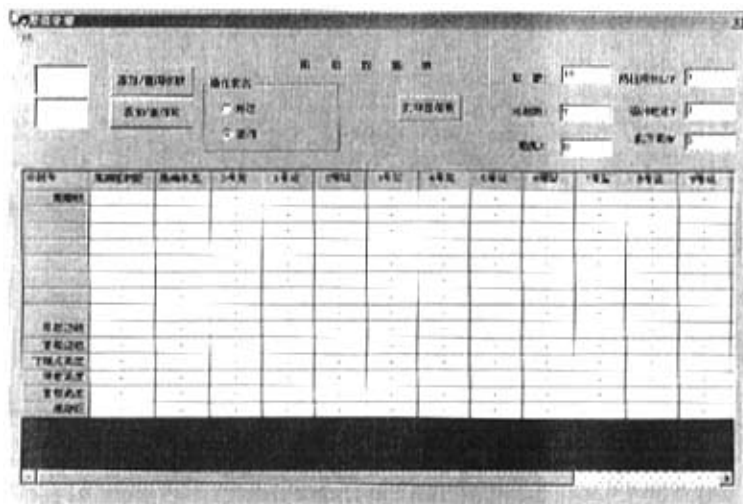


图 5-2

在操作状态中选中“修改”，就可以输入型值了，注意每输入一条水线，需要在左上角的文本框中先输入该水线的高度，然后点击“添加/查询水线”按钮，在相应的行输入该型线的型值。同样，表中原来没有的站号，比方说  $8\frac{1}{2}$ ，也需要在左上角的文本框中输入站号“8.5”，然后点击“添加/查询站线”按钮，然后就可以输入这个站的数据了。型值表的输入是个费力的工作，

输入完毕，可以通过左上角的系统菜单进行保存。

这个对话框实际上是一个独立的数据处理程序，后面还会用它进行型值查询。

在选择输入方式中，如果选择“从文件载入”，点击“浏览”按钮，如图所示提示选择输入数据文件。这里就需要事先把计算所得的型值表数值和相关数据按照指定的格式存放到数据文件中，在这里就以直接载入进行绘图了。

展示程序里为了方便，是从数据文件里直接载入型值的方式输入型值的。

### 5.1.3 绘制型线图

型值录入完毕，就可以绘制型线图了。在 AutoCAD 命令行里输入“draw”便得到初始的型线图，如图 5-3 所示。

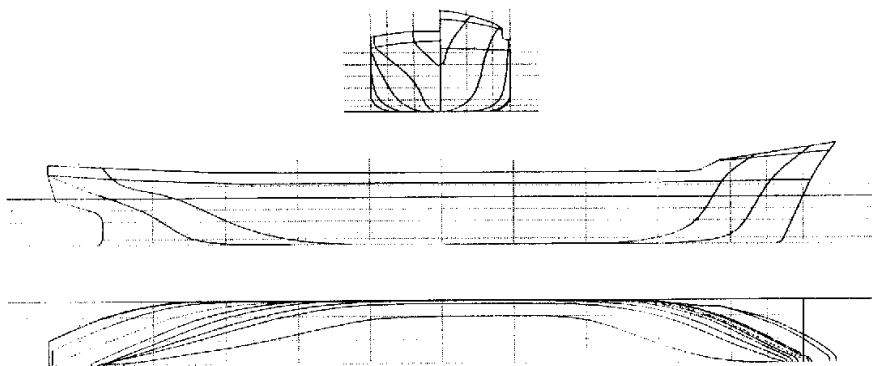


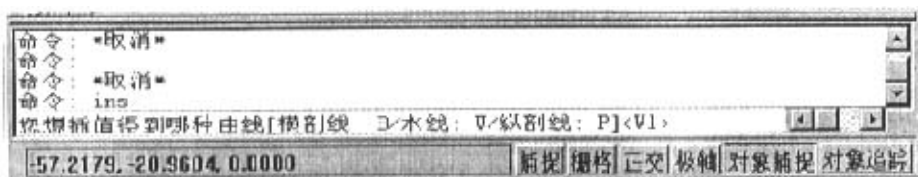
图 5-3 绘制的某船型线图

绘图之前，型线已经进行了光顺，但有些部位可能不是太理想，利用 AutoCAD 绘图工具，可以手工适当调整。

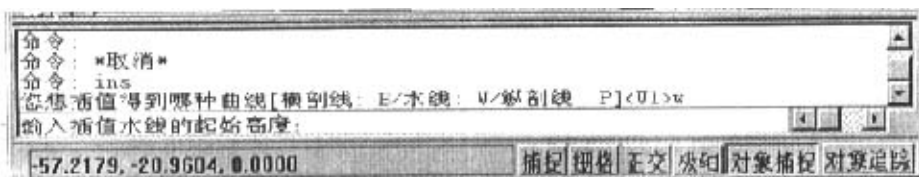
### 5.1.4 型线图的插值操作

下面可以对绘制出来的型线图进行插值，通常插值有 3 种方式，一是在船舶纵向插值得到所需位置的横剖线，二是沿着垂向插值得到所需位置的水线，还有一种就是在船宽方向插值得到纵剖线。这三种方式对应着三种插值命令。

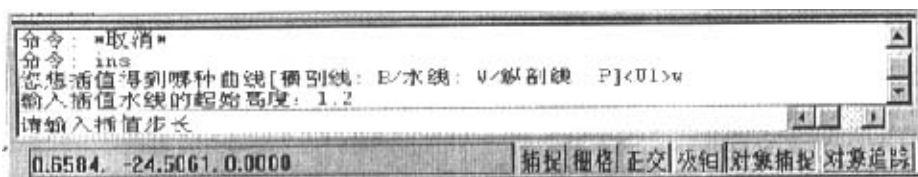
插值操作的命令是由自定义命令 INS 来完成的。在命令行里输入“INS”，则系统提示如下：



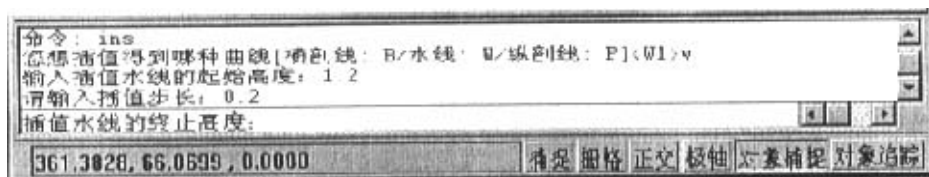
选择“W”，以绘制水线，则接着提示如下：



假设输入插值水线高度“1.2”，则接着提示如下：



输入步长 0.2，则提示如下：



输入终止高度 5.0，则得到如下所示的插值水线：



图 5-4 插值水线图

其中深色的是原始型线，浅色的是插值水线。

上面是水线的插值过程，在第一步插值型线类型选择中选择“B”会插值横剖线，选择“P”则会插值纵剖线，后面的提示与插值水线类似，都是依次要求输入起始位置，步长和终止位置，如果在输入步长的时候直接回车跳过，则会得到起始点位置的一条插值曲线。

## 5.2 数据的处理<sup>[31, 35, 42]</sup>

在前面的绘图和插值操作中，比方说一次插值若干条插值水线，插值完毕，系统会提示你是否保存将刚才进行的插值保存到性质表，需要保存的话，系统会弹出“保存文件”对话框进行保存，本程序产生的数据文件以“.ship”结尾。

绘制型线图并进行插值最终是为了方便地查询船舶任意位置的型值，为肋骨的冷弯等加工设备提供所需的数据。通过型线图的插值可以得到比原始型值表详细的多的新型值，把这些扩充过的型值保存到数据文件里，然后型线型值的查询就可以独立于绘图程序，虽然在 AutoCAD 里可以边插值型线图边获得所需的型值，但是每次查询每次都要绘图，这样查询速度比较低，效率不高，况且有时候查询并非一定要绘出相应的图，而数据的查看也不方便。所以有必要用专用的数据处理程序来处理这些数据。这里提供了数据接口和数据查询的人工操作界面。

### 5.2.1 数据接口

这里的数据接口实际上是一个函数，当数控设备需要数据时，只要提供 2 个参数，数据处理程序就可以相应这个需求，返回一组相应的型值，接口函数如下所示：

`GetData (int type, double position, AcGePoint3dArray RtArray)`

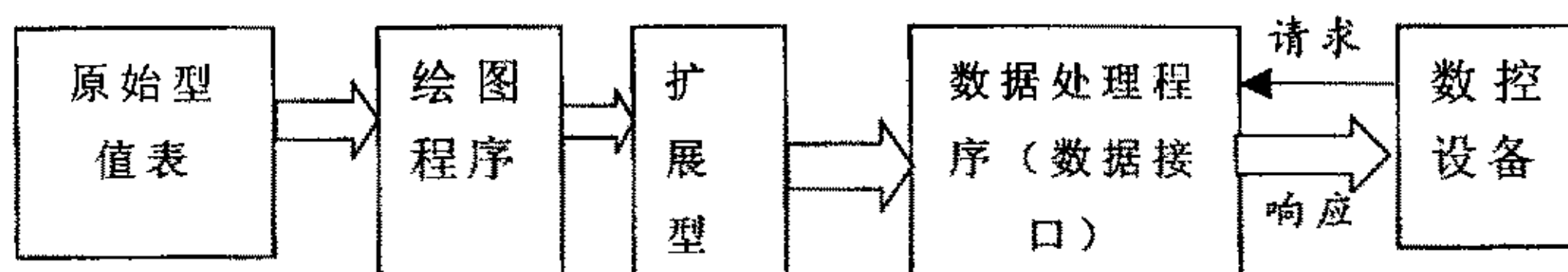
参数的含义如下：

type—所需数据的类型，以区分横剖线，纵剖线，水线。

position—设定所需水线、纵剖线、横剖线的位置

RtArray—用于存放返回的数据集合

因此，从原始型值表到数控设备，数据流示意如下：



### 5.2.1 人工操作界面

上面讲了主机和数控设备之间的数据接口，除此之外，这里还提供了人工数据操作界面。它可以用来修改型值表，也可以用来查询扩展型值表，获得相应的数据。这个界面也就是图 5.2 所示的界面。

#### 1) 型值的编辑

在“操作状态”单选按钮中选中“修改”，则型值表格处于编辑状态，可以修改型值，在“添加查询水线”按钮左边的文本框中输入水线的高度，点击“添加查询水线”，此时会在表格中自动增加一行，可以在其中输入新增水线的型值，“添加查询站”操作与此类似，可以增加某一站的型值。

#### 2) 型值的查询

在“操作状态”单选按钮中选中“修改”，则型值表格处于编辑状态，可以修改型值，在“添加查询水线”按钮左边的文本框中输入水线的高度，点击“添加查询水线”，此时会在表格中自动增加一行，可以在其中输入新增水线的型值，“添加查询站”操作与此类似，可以查询某一站的型值。



## 第 6 章 总结与展望

### 6.1 主要研究工作总结

本文以 Visual C++ 为平台对 AutoCAD 进行了一些二次开发工作, 以实现船舶型线的自动绘制功能, 基本上达到了预定目标。这个过程表明, 通过对通用绘图软件进行二次开发, 针对特定领域的应用情况, 可以大大提高通用软件的利用效率。例如船舶设计中大量的肋位线的绘制费时费力, 而用本文提供的命令, 可以轻而易举完成上百条肋位线的绘制。同时由于本文程序是属于二次开发, 可以充分利用通用软件现有的许多功能, 使程序的编制相对简单, 缩短了开发周期。本文通过对某船的模拟绘制过程, 得到以下结论:

1) 曲线的拟合形式的多样化。本文在对大部分型线插值是基于 B 样条曲线, 而对型线进行光顺却是基于小参数模型, 实际上无论基于哪一种方式, 实质上都是一样的, 都是对客观事物的数学描述。通常说来, B 样条有很好的手工光顺可操作性, 但是本文的目标是尽可能实现自动光顺, 处于次要地位的手工光顺 AutoCAD 提供的工具就足够使用了。

2) 开发工具选择的重要性。合适的开发工具对程序的开发意义重大, 本文用 Visual C++ 直接操作 AutoCAD 内核, 许多绘图功能的实现都很方便; 并且程序完成之后, c++ 语言还能带来执行上的高效率。

本文绘图程序的开发与常规绘图程序的开发相比具有以下优点:

1) 通过 objectarx 开发工具深入操作 AutoCAD, 程序的执行效率很高, 并且充分利用了 AutoCAD 的现有绘图资源。

2) 用准均匀三次 B 样条拟合船舶型线进行插值, 精度比较理想。从最终插值得到的曲线来看, 总体效果还是不错的。

### 6.2 研究展望

本文在船舶的型线绘制方面对 AutoCAD 的二次开发方面作了一些探索性的工作, 初步实现了程序的基本功能, 但整体功能还不够丰富, 还存在着许多不足, 有待进一步展开研究。

1) 程序的参数化程度不够高, 船型多种多样, 而本程序只能对一些特定的不太复杂的船型进行绘制。

2) 对船舶型线的处理还不够全面, 有些部位特别是船舶首部水线的光顺解决的不够理想, 还需要人工做必要的修改才能符合要求。

3) 数据的处理从发展趋势来看用数据库来存储比较好, 以后有必要进行这方面的改进。

## 致 谢

本文是在导师董元胜教授的悉心指导和关怀下顺利完成的。在本人硕士学习期间，导师不仅在学业方面给予了详尽的指导，而且在生活上给予了无微不至的关怀，以及教导我做人的道理。导师严谨求实、锐意创新的治学态度、渊博的知识、敏捷的思维、开阔的视野、对事业的执着追求和忘我的工作精神、始终饱满的工作热情及正直善良的人品使本人深受感悟，受益匪浅。在此对导师致以深深的敬意和诚挚的感谢！

特别感谢陈宾康教授在本人课题研究和撰写论文期间给予的帮助，陈宾康教授对科学的一丝不苟和严格要求、分析问题敏捷的思维和宽广渊博的知识及和蔼可亲、平易近人的待人风格，令本人终身难忘，在此对陈宾康教授致以衷心的感谢！

感谢石强、杨帅、陈红梅、李毓江、熊文海的关心和帮助，同时还要感谢王宇和刘小建两位师弟的热心帮助和支持！

感谢我深爱的父母！您们不仅赐予我宝贵的生命，还言传身教让我懂得做人的道理！感谢你们二十多年来在精神和物质上给予的巨大支持！

## 参考文献

- [1] 陈宾康、董元胜 计算机辅助船舶设计, 国防工业出版社
- [2] 王宇成. 迈向 21 世纪的世界船舶工业与船舶科技. 船舶工程, 1998.1
- [3] 薛炎华. 计算机辅助设计发展动向及其对策. 电子与自动化 1993.22
- [4] 曾隆杰. 船舶 CAD. 人民交通出版社
- [5] 刘宁、王其昌. 浅谈船舶 CAD 软件开发的几个问题. 软件开发与应用, 1996. Vol1.1
- [6] 郭曼君, 国际船舶 CAD/CAM 软件发展与应用 1999.5
- [7] 陈宾康. 用 B 样条拟合与绘制船体型线. 武汉水运工程学院学报 1989.13
- [8] 施法中. 计算机辅助几何设计与非均匀有理 B 样条(CAGD&NURBS). 北京航空航天大学出版社
- [9] 林焰、纪卓上、戴寅生. 船体 B 样条曲面的数学描述及计算方法. 中国造船. 1996. Vol.4, No.7
- [10] 李水进、周艳红. B 样条曲线的局部自动光顺算法. 中国机械工程 1998.9
- [11] 陈宾康. 函数合成法生成型线与光顺. 中国造船. 1992 .1
- [12] 施法中. 均匀 B 样条曲线的几何意义. 数值计算与计算机应用. 1985.
- [13] Farin G. Curves and surfaces for computer aided geometric design: A practical guide. Academic press. 1988
- [14] Choi B K. Surface Modeling for CAD/CAM. Elsevier. 1991
- [15] 蒋大为、李安平, B 样条曲线的最小二乘保形光顺逼近. 工程数学学报. 2000.17
- [16] 熊西文. 船体型线光顺. 国防工业出版社
- [17] 樊勇兵、陈宾康 关于曲线光顺性的讨论 . 船舶工程. 1999 .3
- [18] 程明道. 船型生成. 船舶工程. 1997.2
- [19] 冯恩德、席龙飞 . 船舶设计原理. 大连海运学院出版社. 1989
- [20] 林焰、纪卓尚、戴寅生. 船舶型线设计软件系统. 软件开发与应用. 1995. Vol. 1, No. 6
- [21] 陈建兰、洪涛. 基于函数样条的船舶型线的三向光顺的研究 杭州电子工业学院学报 2002.22
- [22] 李俊华、应文烨等. 基于 CADD5 的船体外板加工样板软件开发 中国造船. 2002 .3
- [23] 蒋大为、王方明. 保形光顺的样条曲线. 航空计算技术. 1998 28
- [24] 汪希龄. 用计算机作船体型线设计. 中国造船. 1978. Vol.3, No.4
- [25] [美] Davis Chapman 著. 骆长乐译. 学用 VC++6. 0. 清华大学出版社. 1999
- [26] 张金铭. 数学船型设计及其三向光顺性. 上海交通大学学报. 1980. Vol14, No.3
- [27] 谭浩强. C 程序设计. 清华大学出版社 1988

- [28]钱能. C++程序设计教程.清华大学出版社.1999
- [29]Pedersen.Einar、 Hatling, John Fredrik.Computer integrated ship production Journal of Ship Production.1997,13
- [30] Sapidis.N, Farin.G .Automatic Fairing Algorithm for B-spline Curves[J] .Computer Aided Design.1990,22
- [31] James V.Valentino. CAD with Computer Vision .CBS HRW U.S.A 1987
- [32] Computer Applications in the Automation of Shipyard Operation and Ship Design. 1988, 1991, (6) (7)
- [33]Abrams, S.L.、 Bardis, L.、 Chrysostomidis C.Geometric modeling and interrogation system praxiteles.Journal of Ship Production. 1995.11
- [34] Rogers D F. Fog N G. "Constrained B-Spline Curve and Surface Fitting".ICCAS.1998
- [35]陈宾康.船舶静力学现代计算法.大连海事大学出版社.1995
- [36] Tan. K、 T.Bligh. New approach to an integrated CAD method for surface ship design. Naval Engineers Journal. 1998,110(1)
- [37]黄永红等.WINDOWS 环境下 CAD 用户界面的设计与实现.计算机辅助工程。1999 (5)
- [38]陈彬等.用 ObjectARX 开发 AutoCAD 2000 应用程序.人民邮电出版社, 2000
- [39]李世国.AutoCAD 高级开发技术 ARX 编程及应用. 机械工业出版社,1999
- [40] 何义、石德新、邹劲.船体型线三向光顺系统研究.哈尔滨工程大学学报.Vol.21,No.2
- [41]李于剑.VisualC++与提高一形图像编程篇.中国铁道出版社.2001
- [42]二代龙震工作室, AutoCAD&数据库整合运用实作.北京:机械工业出版社,2002
- [43]李世国、潘建忠译.AutoCAD2000 ObjectARX 编程指南.机械工业出版社,2000
- [44] 王勇毅, 船舶建造工艺学.人民交通出版社,1994
- [45] 杨永祥、茆文玉、翁士纲.哈尔滨工程大学出版社
- [46] 张永生. 图形图象软件工程.解放军出版社
- [47] 王润生. 图形理解 国防科技出版社
- [48] Aldefeld B. Variation of geometries based on a geometric reasoning method. Computer\_Aided Design. 1978,10(3)
- [49] Hagen H. Roller D. eds. Geometrics Modeling. Methods and Applications. Springer-Verlag. N. Y. 1991
- [50]Choi B k. Surfice Modeling for CAD/CAM.Elsevier.1991
- [51] Bohem W. Inserting New Knots into B-Spline curves. Computer\_Aided Design. 1980,Vol.6,No.12
- [52]A Z Qanliyah. A strategy for the construction of customized design libraries for CAD Computer\_Aided Design. 1996,30.(6).
- [53] Tom.Beard. Interpolating Curves. Modern Machine Shop 1997,0(10)

- [54] Matthias Eck 、 Jan Hadenfeld Knot removal for B-spline curves Matthias Eck and Jan Hadenfeld 1995,12(3)
- [55] Carl E. Jacobson .Using AutoCAD for descriptive geometry exercises - in undergraduate structural geology. Computers & geosciences. 2001,27(1)
- [56] Feature Extraction Software Supports AutoCAD Imaging Notes. 2000,15(2)
- [57] Tony.Hotchkiss.Make AutoCAD 2000 do what CADalyst 2000,17(7)



## 硕士期间发表的论文及所做的科研工作

- [1] 黄永生、董元胜. ObjectARX2002 在型线设计中的应用 船海工程.2004.1
- [2]黄永生.船舶型线数学光顺在 AutoCAD 中的实现.中国水运（理论版）.2004.2（待发表）