

第三篇

矿山工程 CAD 制图技术

第一章 CAD 基础知识

第一节 基本概念

CAD 技术出现于 60 年代初期 ,并随着计算机软硬件的发展而逐渐发展成为相当成熟的设计工具 ,目前已作为商品进入市场 ,在许多行业中发挥着重要作用。

CAD 技术是利用计算机及其外围设备和图形输入输出设备来帮助人们完成工程和产品设计的。CAD 技术的最大特点是以计算机图形处理方法来进行设计 ,但其功能并不仅仅局限于此。人们可以应用 CAD 技术进行数据资料处理、工程设计优化、图形设计及经济评价等各种工作。应用 CAD 技术进行工程设计 ,不仅可以大大提高工作效率 ,同时能够显著提高设计质量。

第二节 CAD 系统的硬件设备

一、计算机

计算机是 CAD 系统的主体。计算机可以是大型机、小型机或微型机 ,可以多用户共享一台计算机 ,也可以一个用户使用一台或多台计算机。就我国目前情况而言 ,矿山

CAD 系统采用微型计算机比较切实可行。

图形处理需要进行大量计算,交互设计要求计算机响应时间短,因此 CAD 系统的计算机应该具有较快的运算速度,同时应该有较大容量的内存和硬盘。基于这些考虑,矿山 CAD 系统选用 586 以上档次的微机比较理想。

二、图形输入设备

常用的图形输入设备有键盘、鼠标、光笔、数字化仪等。

(1)键盘。键盘主要用于输入数值信息和字符串信息,在相应软件的支持下,也可用键盘控制图形光标,模拟数字化仪,输入图形坐标。

(2)鼠标。鼠标是一种点输入设备,它增强或代替了键盘的光标移动键的功能,在屏幕上更为快速准确地定位光标。

根据工作原理,鼠标可分成两种类型。一种是机械式鼠标,底部装有轮子。当轮子在桌面上移动时,轮子随之转动,连接在轮子上的电位器检测出鼠标移动的方向和距离,并传递给计算机。另一种是电子式鼠标,底部装有光电管。当鼠标在特制的刻有网格线的光滑铝板上移动时,光电器检测出移动方向和距离,同时传送给计算机。

鼠标一般装有 1~4 个按钮,按钮的作用可由软件进行定义。鼠标的位置对应屏幕上的图形光标,移动鼠标,并按一个特定的按钮,就可把光标的屏幕位置送入计算机。

(3)光笔。光笔是一种直接用于输入点标点的设备。光笔的外形象一支圆珠笔,其中有电子器件和光电纤维,用来检测显示图形的光强,并把相应的光信号转换放大成脉冲信号。光笔不仅可输入点坐标,也可用它来改变显示点的位置和选择在屏幕上的图形或菜单。

(4)数字化仪。数字化仪是图形输入的主要设备。数字化仪由感应板和指示笔或定标器组成。当游标在感应板上移动时,定标器中的电信号与网状导线相互作用,使网格十字叉线处生成感应电磁场,从而确定了定标器在感应板上的位置。

数字化仪使用的定标器一般有 4 个或 16 个按钮,其作用可由程序定义。与鼠标器不同的是,数字化仪向计算机提供的是定标器在感应板上的绝对坐标位置,而鼠标器则只能提供它与桌面的相对位移量。因此,数字化仪除了可以象鼠标器一样作为交互式定位器使用之外,它的另一重要作用就是把原始图件的几何数据输入到计算机里。

数字化仪的性能参数主要有操作方式、输出格式、分辨率和数据转换率等。

数字化仪主要有以下几种操作方式:

①点方式。点方式(Point)是每当笔尖轻触感应板或者定标器的键被按下时,感应板

就发送一对数据到计算机。

②开关流方式。在开关流(Switch Stream)方式下 ,每按下一次键 ,就将一组坐标数据发送到计算机。

③连续流方式。在连续流(Stream)方式下 ,无论定标器的键或者笔尖的状态如何 ,感应板连续不断地发送坐标数据对。也就是说 ,无论定标器的键是否按下 ,数字化仪都在向计算机发送坐标数据对。

④ δ 方式。在 δ 方式(Delta)下 ,数字化仪向计算机发送的是相对于上一对坐标的位移量 ,而不是象其它方式那样 ,发送的是绝对坐标值。

⑤步进方式。在步进方式(Increment)下 ,当定标器在感应板上移动某个距离 ,或者移动某个最低极限距离 ,数字化仪就发送一个绝对坐标值。距离大小由送入步进方式的控制命令来决定。

⑥鼠标方式。鼠标方式(Mouse)是用来模拟鼠标系统的 ,当定标器在感应板上移动时 ,就连续不断地发送坐标数据。

⑦触发方式。这种方式是每触发一次 ,数字化仪就向计算机发送一个坐标数据 ,触发的操作是由命令控制的。

⑧机控方式。在机控方式下 ,可由计算机发送命令给数字化仪 ,来改变数字化仪的状态和参数。

⑨提示方式。提示方式(Prompt)是在数据被传送之前 ,在输出的信息中带有注释信息。在通常情况下 ,提示方式应和其他方式连用。

数字化仪的输出格式指数字化仪向计算机发送的数据串的格式规定。共有两种基本格式 ,其一是 ASCII 格式 ,第二是二进制格式。

数字化仪的数据转换率是指感应板每秒向计算机发送的坐标对数。

分辨率指感应板输出的距离增量 ,以每英寸行数或每毫米行数表示。

(5)其它输入设备。除了上述的输入设备外 ,还有扫描仪和语音输入设备等。这些设备有的尚处于研制阶段 ,有的不太常用 ,此处介绍从略。

三、图形显示设备

目前的图形显示设备主要采用标准的阴极射线管(CRT)结构的显示器。

(一)随机扫描式图形显示器

从显示文件存储器中取出显示指令、方式指令送到显示控制器 ,由显示控制器控制电子束的偏移 ,在屏幕上产生一条发亮的图形轨迹。随机扫描式显示器中电子束的定位

和偏转具有随机性,在某一时刻只有一点发亮,因此可以画出很细的线条,故又称之为画矢量显示器。

(二)存储管式显示器

存储管类似 CRT,不同的是电子束不直接打在荧光屏上,而是用电子枪将图形信息写在一个细栅格上,再由电子枪把存储在网格上的图形重写在屏幕上。

(三)光栅扫描式显示器

这种显示器的 CRT 中装有水平偏转线圈和垂直偏转线圈。电子束在水平偏转线圈产生的水平磁场的作用下,沿着水平方向扫描,称为行扫描;电子束在垂直偏转线圈产生的垂直磁场的作用下,沿着垂直方向扫描,称为场扫描。电子束自上而下、自左向右进行扫描,就形成了光栅。光栅扫描显示器控制电子束依次扫描整个屏幕,屏幕上每个象素(扫描点)的亮度和颜色都可控制,因此这种显示器可以输出区域彩色图和具有明暗度差别的真实图形。

衡量监视器性能的指标主要有可显示的颜色数目,显示存贮区容量和屏幕的分辨率等。

(四)IBM PC 图形显示器

自 IBM PC 个人计算机问世以来,随着 PC/XT,PC/AT,PS/2 及各种 PC286、PC386 及 PC486 兼容机的出现,与 PC 机配套的各种性能的图形显示器也相继开发成功并进入市场。下面简单介绍几种主要的 IBM PC 图形显示器。

(1)CGA 彩色图形显示器(Color Graphics Adapter) CGA 是最早与 IBM PC 机配套使用的彩色图形显示器,也是 IBM 公司个人计算机的第一个图形显示器标准。CGA 显示器工作在单色显示模式时,分辨率为 640×200 ;彩色显示时,分辨率为 320×200 ,4 种颜色,显示存贮区 16kB。

(2)EGA 增强型彩色显示器(Enhanced Graphics Adapter) EGA 是 IBM 公司推出的第二种个人计算机图形显示器标准。它与 VGA 兼容。彩色图形显示最高分辨率为 640×350 ,可同时显示 16 种颜色,颜色总数为 64 种。显存容量为 256kB,分成 4 个位平面,并有一张 16×6 位的彩色表。

(3)VGA 彩色图形显示器(Video Graphics Array) VGA 是与 PS/2 一起推出的性能更好的彩色图形显示器。它与 CGA、EGA 保持兼容,同时增加了若干新的显示模式。显存容量为 256—512kB。彩色图形显示最大分辨率为 640×480 ,可同时显示 16 种颜色,但颜色总数达 2^{18} 种。另一种彩色显示模式的分辨为 320×200 ,可同时显示 256 种颜色。

(4)TVGA 彩色图形显示器。TVGA 是美国 Trident Microsystems 公司开发的 VGA

兼容显示器。显示存储器容量为 256 ~ 512kB ,分辨率有多种选择 :640 × 350(256 色) , 640 × 400(256 色) ,640 × 480(256 色) ,800 × 600(16 色) ,1024 × 768(16 色) ,所有不同颜色的总数为 2^{18} 。

(5)ARTIST 图形显示器。ARTIST 图形显示器系列产品具有屏幕大、分辨率高的特点。例如 ARTIST1 型分辨率为 1024 × 1024(隔行扫描)16 色 ,ARTIST Plus 的分辨率为 1024 × 768(逐行扫描)16 色。显存容量为 512kB。这些显示器不与 CGA、EGA、VGA 等兼容 ,但配有图形子程序库 ,可以支持许多图形应用软件。

四、绘图输出设备

(1)打印机。打印机是最廉价的绘图输出设备 ,常用的有行式打印机、点阵式打印机、激光打印机和喷墨打印机等。

(2)滚筒式绘图仪。绘图纸卷在滚筒上 ,由电机转动滚筒 ,从而带动图纸运动。绘图笔架在滚筒上方由另一个电机带动 ,沿垂直于图纸运动的方向运动 ,绘制图形。

(3)平板式绘图仪。图纸平铺在绘图平板上。画笔的驱动方式有两种 :一种是由两个电机以机械传动方式驱动横梁和笔架的运动 ;另一种是用平面电机驱动 ,电机的转子就是笔架 ,无需机械传动。

第三节 CAD 系统的软件

CAD 系统的软件一般可分为系统软件、支撑软件和应用软件三大类。系统软件是 CAD 系统的核心 ,是软件系统与硬件设备的内层界面 ;支撑软件是提供图形处理和编辑、数据库管理等功能的通用基础软件 ;应用软件是用户根据自己的需要、在系统软件和支撑软件的支持下 ,开发的解决各种实际问题的 CAD 软件。

系统软件通常指操作系统以及 BASIC、C、FORTRAN、Prolog 等高级语言。

支撑软件一般有以下几种 :

一、标准图形软件

图形支撑软件通常由一组公用的图形子程序组成 ,它扩展了系统中原有高级语言和操作系统的图形处理功能 ,其作用是减少应用软件的图形开发工作。图形软件标准化一

直是计算机图形字及 CAD 技术的重要研究领域。标准图形软件具备以下特点。

(1)使应用软件独立于硬件设备。在标准图形软件上开发的各种图形应用软件,能够在不同类型的计算机及图形输入输出设备上运行。

(2)与具体应用无关。标准图形软件的各种图形处理功能,综合考虑了多种应用的不同要求,具有广泛的适用性。

(3)具有较高性能。标准图形软件能够提供多种图形输出原语,如线段、圆弧、曲线、填充等,能支持各种图形输入设备的操作,可以对图形进行各种编辑。因此,可以在较高的起点上开发图形应用软件。

国际上于 70 年代中期开始着手图形标准化工作,已推出的软件有:计算机图形接口(Computer Graphics Interface)、计算机图形文件标准 CGM(Computer Graphics Metafile)、计算机核心系统 GKS(Graphics Kernel System)面向程序员的层次交互式图形标准 PHIGS(Programmer's Hierarchical Interactive Graphics Standard)以及基本图形转换规范 IGES(Initial Graphics Exchange Specification)等。这里的标准有些是面向图形设备的驱动程序包、有些是面向用户的图形程序包,有些是面向不同 CAD 系统的文档规范。

二、通用图形处理软件包

就目前情况而言,更为常用的图形支撑软件是一些通用的图形处理软件包,例如 AutoCAD、Intergraph 等。通用图形处理软件包通常具有丰富的图形处理和编辑功能,能够支持和驱动常用的图形输入输出设备,设计有与应用软件发生联系的接口。

三、数据库管理系统

数据库是 CAD 系统的重要组成部份。在矿山设计过程中,经常需要使用大量的地质和采矿生产方面各种各样的数据。为提高工作效率,这些数据应该由数据库系统进行管理。

数据库可以理解为存储在一起的相互有关的数据集合。这些数据中没有不必要的冗余,数据的存储可以独立于应用程序,对数据库插入新数据,修改和检索原有数据均能按一种公用的可控的方法进行。数据库管理系统是实现有组织地、动态地存储大量有关联的数据、方便用户访问的软件系统。

目前在 PC 机上使用的数据库管理软件主要有 DBASE 和 FOXBASE 等。

四、网络通讯系统

网络通讯系统能使数据资源尽可能共享。在 PC 机上使用的网络通讯系统有 Net-

ware 等。

五、汉字管理系统

菜单显示汉字、图形标注汉字时 ,需要汉字管理系统和矢量汉字库。

第四节 CAD 技术在矿山的应用

从地质勘探到采矿生产 ,矿山工程的各个发展阶段都需要处理大量的基础数据 ,绘制大量的地测和工程图纸 ,进行各种工程设计、制定生产计划和选择开采方案。CAD 技术能够在矿山众多的领域得到应用。

(1)地质方面。处理地质勘探和生产勘探的各种资料数据 ,建立矿床地质模型 ,绘制各种地测图件。

(2)矿山设计。进行露天矿和地下矿井的矿山基建设计、评价矿山开采方案。设计矿山开采境界、生产工艺系统和各种采矿工程。

(3)矿山生产设计与管理。设计采矿方法、制定开采方案、优化生产计划。

CAD 技术主要通过计算机图形处理方法进行工程设计 ,因此 ,计算机图形是 CAD 技术的基础。本节简要介绍计算机图形的组织结构、图形几何变换和图形裁剪等方面的基础知识。

第五节 计算机图形的组织结构

一、基本图形元素的描述

计算机图形 ,可以分解为一系列点、线、曲线以及字符等基本的图形元素。换句话说 ,计算机图形是由基本图形元素组成的。

基本图形元素是以各种形式的数据进行定义和描述的。在描述基本图形元素时 ,通常要说明以下情况。

(1)类型。说明它是哪一类基本图形元素。

(2)几何参数。用以确定元素的几何位置 ,例如 ,元素是一个点时 ,几何参数就是该点的坐标值。

(3)属性。用以表示亮度、线型、是否闪烁和色彩等。

(4)分析数据。用来给图形元素赋以某种数据 ,例如某曲线表示煤层等高线 ,那么等高线的高程值就可以作为分析数据。

(5)名字。供程序查找或调用。

以上所述的是描述图形基本元素应遵循的一般原则。对图形元素乃至整个图形信息的具体描述 ,要通过建立合适的图形数据结构来实现。

二、几何信息和拓扑信息

仅对图形基本元素进行描述还不能完整地反映一幅图的信息 ,还必须说明图形元素之间的相互关系。图形元素只有通过有序的结合。才能构成一幅有意义的图形。

计算机图形信息可分为几何信息和拓扑信息两大类。几何信息表示图形中各个基本元素的几何性质和度量关系 ,元素的几何特征等 ,拓扑信息用来说明图形元素——点、线、面等相互间的连接关系。

图 3-1-1a 为分布在平面的 7 个点 ,点的编号依次为 1,2...7。图 3-1-1b 和图 3-1-1c 分别表示以不同的顺序连接这 7 个点后形成的多边形。在图 3-1-1b 中顶点构成边的顺序为 14,43,36,67,75,52,21,而在图 3-1-1c 中 ,顶点连接成边的顺序为 13,36,67,74,45,52,21。图 3-1-1b 和图 3-1-1c 中顶点的坐标是相同的 ,由于连接顺序的不同 ,就形成了相异的两个多边形。

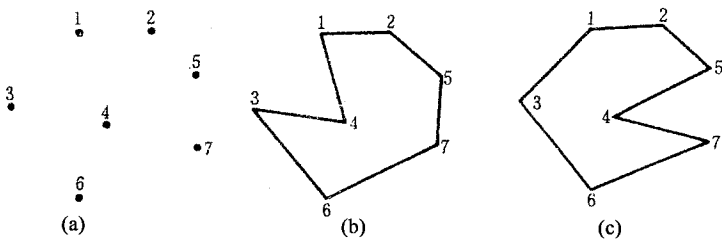


图 3-1-1 多边形

在图 3-1-1b 和图 3-1-1c 中 ,各顶点的坐标就表示了图形的几何信息 ,顶点与边的连接关系就是图形的拓扑信息。在描述上述图形时 ,不仅要记录顶点的坐标值 ,同时

还要记录顶点与边的对应关系。

三、图形数据结构

(一) 数据结构的基本概念

计算机科学中的数据是指描述客观事物的数、字符以及所有能输入到计算机中并被计算机程序加工处理的符号集合。数据元素是数据的基本单位。有时一个数据元素由若干个数据项组成,此时数据元素称为记录,数据项是数据的最小单元。

被计算机处理加工的数据彼此间一般存在着某些联系,通常将数据元素间的这种联系称为结构。数据结构就是带有结构特性的数据元素的集合。计算机科学及其各种应用领域都要用到各种数据结构。数据结构是计算机科学表示各种复杂事物的基本手段。

数据结构中的数据或记录称为该数据结构的结点。结点一般由数据域和指针域组成(图 3-1-2)。其中数据域存放该记录本身的各项指标和属性,而指针域则存储后续元素的地址。

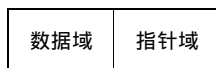


图 3-1-2 记录结构

数据结构在计算机内部主要有顺序存储和链式存储两种方式。顺序存储方式主要用于线性的数据结构,它把逻辑上相邻的数据元素存储在计算机存储器相邻的存储单元里,结点之间的关系由存储单元的邻接关系来实现,因此在这种结构里,结点的指针域是不需要的。链式存储方式就是在每个结点中至少包括一个指针域,逻辑上相邻的数据元素不一定存储在相邻的存储单元里,而是用指针来体现数据元素之间逻辑上的联系。

(二) 线性表结构

线性表是最简单也是最常用的一种数据结构。线性表是 n ($n \geq 0$) 个数据元素的有限序列,其逻辑上的结构可表示为 (d_1, d_2, \dots, d_n) ,线性表中所包含元素的个数称为线性表的长度。

(1) 向量和一维数组。用顺序存储结构存储的线性表称为向量表,可以用高级语言中的一维数组来表示。往数组中插入一个新的数据元素时,插入点以后的所有元素都要向后移一个位置;类似地,从数组中删除一个结点时,其后的所有元素都要前移一个位置。

(2) 链表。用链式存储结构存储的线性表称为链表。链表又分为线性链和循环链表。线性链表的结点只含有一个指针域,用来指出其后续结点的位置。线性链表的最后

一个结点没有后续点,它的指针域为空(记为 NULL)此外还要设置一个头指针 head,指向链表的第一个结点(图 3-1-3)。循环链表是指链表的最后一个结点的指针指向第一个结点,即最后一个结点的指针值不是空,而是第一个结点的地址(图 3-1-4)。

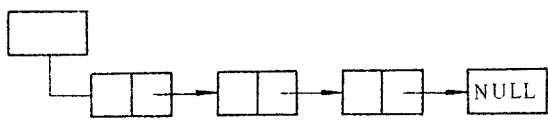


图 3-1-3 单链表

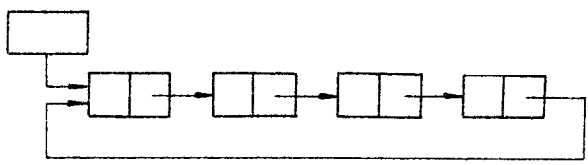


图 3-1-4 循环链表

链表的元素插入和删除运算很方便,不需移动结点,只需改变结点中指针域的值即可。图 3-1-5 和图 3-1-6 分别表示了从链表中插入和删除一个结点后的指针变化情况,虚线表示变化后的指针。

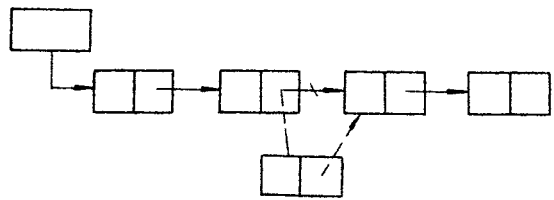


图 3-1-5 单链表的插入

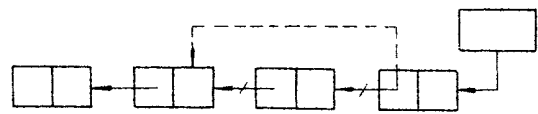


图 3-1-6 单链表的删除

(3)栈和队列。栈和队列都是线性表的特殊形式。栈是限定仅在表尾(表的一端)进行插入和删除运算的线性表,表尾称为栈顶,表头称为栈底。队列是限定所有插入都在表的一端进行,所有删除都在另一端进行的线性表。进行删除的一端叫队列的头,进行插入的一端叫队列的尾。

(三) 树与二叉树结构

树形结构是非线性结构,其中最常用的两种形式是树和二叉树。

(1) 树。树是由一个或多个结点组成的有限集合 T ,其中有一个特定的结点称为根,其余的结点分为 n ($n \geq 0$) 个不相交的集合 T_1, T_2, \dots, T_n ,每个集合又是一棵树,称为根的子树。图 3-1-7 表示了一棵树。

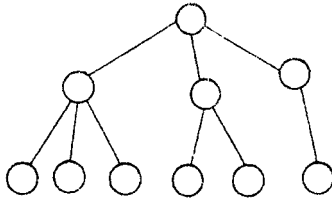


图 3-1-7 树

树形结构描述了数据之间的分层关系。树的存储结构采用多重链表形式,即每一个结点除了有数据域外,还有多个链指针域,分别存放指向该结点的各个子树(子结点)的指针。

(2) 二叉树。二叉树是结点的有限集合,这个集合或者是空的,或者是由一个称为根的结点以及该根的左子树和右子树组成(图 3-1-8)。二叉树不是树的特殊情况。树和二叉树的主要差别是二叉树的结点的子树要区分为左子树和右子树。在通常情况下,二叉树仍采用多重链表的存储形式。表中的每个结点都设有左、右指针域和数据域,左右指针域分别指向该结点的左子树和右子树(图 3-1-8)。

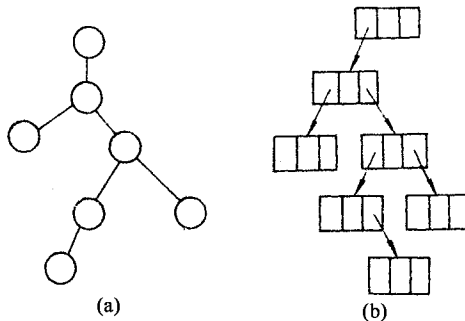


图 3-1-8 二叉树

四、图形数据结构

图形数据结构就是存储了图形元素的几何信息、拓扑信息及有关属性的数据结构。

计算机通过图形数据结构来表示、生成、处理、存储、检索和管理图形。

合理的图形数据结构应满足以下几点要求：

- (1)能够正确描述图形元素的几何位置及有关属性；
- (2)能够表示图形元素之间的相互关系即拓扑关系；
- (3)唯一定义一个图形；
- (4)节省存储容量；
- (5)便于数据查找和修改。

图形数据结构一般采用线性表结构或树结构。下面举例说明怎样通过建立数据结构来表示计算机图形。

例 1 对于图 3-1-9 所给出的三维几何体,可用图 3-1-10 所示的单链三表结构来表示其几何及拓扑信息。其中第一张表给出顶点的坐标值,第二张表给出顶点连成边的关系,第三张表表示了边形成面的关系。上述单链三表结构唯一定义了图 3-1-9 所示的三维几何形体。

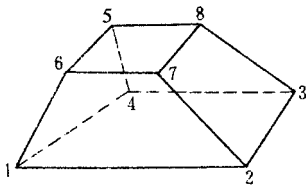


图 3-1-9 棱柱体

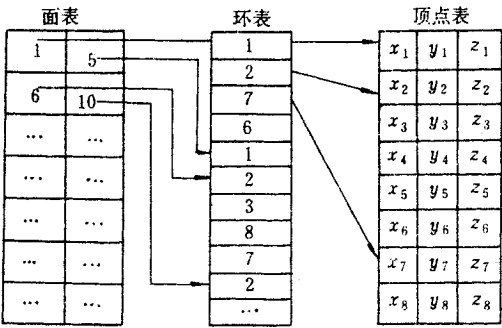


图 3-1-10 棱柱体数据结构

例 2 图 3-1-11 为露天矿采场局部示意图,其中实线表示开采台阶坡顶线,虚线表示台阶坡底线。可设计图 3-1-12 所示的两表结构表示采场的几何状态。其中第一张表存储测量点的坐标值,第二张表存储测量点组成台阶线的关系。第二张表中的 LV

表示台阶水平 ,LT 表示是坡顶线还是坡底线。

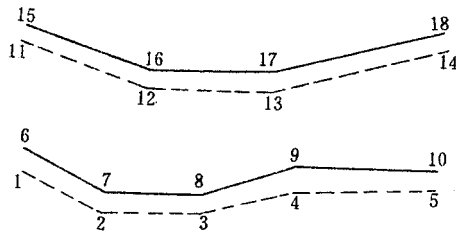


图 3-1-11 露天采场局部

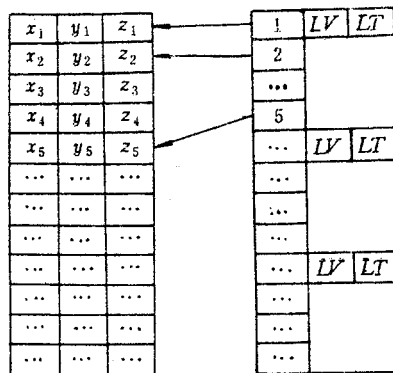


图 3-1-12 采场几何模型数据结构

第六节 图形几何变换

在计算机显示或绘图输出时 ,经常需要对图形进行平移、旋转以及缩放等几何变换。如前所述 ,计算机图形都可以看成是由点、线、面等基本图形元素构成的 ,而线、面等都可以用具有一定关系的点来表示 ,因此 ,计算机图形的几何信息 ,最终可用一个点集来表示。对图形进行几何变换 ,实质上是改变点集的坐标值。

矿山工程图件 绝大部分是二维的。在二维空间里 ,我们可以用一个行向量 $(x \ y)$ 或一个列向量 $\begin{pmatrix} x \\ y \end{pmatrix}$ 表示一个点的坐标。图形几何变换的基本方法是利用一个变换矩阵作用于点的向量 ,得到点的几何变换坐标。

设有一个二阶矩阵 $T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, 由矩阵乘法运算规则可得

$$\begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ax + cy & bx + dy \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix} \quad (3-1-1)$$

其中 $\begin{bmatrix} x^* & y^* \end{bmatrix}$ 是变换后的坐标, 显然有

$$\begin{cases} x^* = ax + cy \\ y^* = bx + dy \end{cases} \quad (3-1-2)$$

由此可见, 变换后的坐标是由矩阵 T 中各个元素决定的, 称矩阵 T 为几何变换矩阵。下面可以看出, 当变换矩阵的元素按照一定的规定取值时, 就可以按照要求实现相应的基本几何变换。

一、比例变换

当 $b = c = 0, a \neq 0, d \neq 0$ 时, 变换矩阵为 $\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$, 由 (3-1-2) 得到变换后的坐标为:

$$\begin{cases} x^* = ax \\ y^* = dy \end{cases}$$

由此可见, a 为 x 方向的缩放因子, d 为 y 方向的缩放因子。当 $a = d > 1$ 时, 图形沿 x, y 方向等比例放大; 当 $a = d < 1$ 时, 图形沿 x, y 方向等比例缩小; 当 $a = d = 1$ 时, 变换前后的点相同, 即发生恒等变换; 当 $a \neq d$ 时, 图形产生畸变, 即沿 x, y 方向的变化比例不同。

二、对称变换

(1) 相对 x 轴的对称变换。当变换矩阵 $T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ 时, 由式 (3-1-2) 式得到

$$\begin{cases} x^* = x \\ y^* = -y \end{cases}$$

此时图形发生相对 x 轴的对称变换。

(2) 相对 y 轴的对称变换。当变换矩阵 $T = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ 时, 由式 (3-1-2) 得到

$$\begin{cases} x^* = -x \\ y^* = y \end{cases}$$

由此可见,此时图形发生相对 y 轴的对称变换。

(3) 相对 45° 线的对称变换。当变换矩阵 $T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ 时,由式(3-1-2)得到

$$\begin{cases} x^* = y \\ y^* = x \end{cases}$$

此时图形发生相对坐标轴上穿过原点的 45° 线的对称变换。

(4) 相对 -45° 线的对称变换。当变换矩阵为 $T = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$ 时,由式(3-1-2)得到

$$\begin{cases} x^* = -y \\ y^* = -x \end{cases}$$

此时图形发生相对坐标轴上穿过原点的 -45° 线的对称变换。

(5) 相对坐标原点的对称变换。当变换矩阵 $T = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$ 时,由式(3-1-2)可得

$$\begin{cases} x^* = -x \\ y^* = -y \end{cases}$$

此时图形发生相对坐标原点的对称变换。

图 3-1-13、图 3-1-14、图 3-1-15、图 3-1-6 和图 3-1-17 分别表示了 $\triangle ABC$ 相对 x 轴、 y 轴、 45° 线、 -45° 线和坐标原点发生对称变换后形成 $\triangle A_1 B_1 C_1$ 的情况。

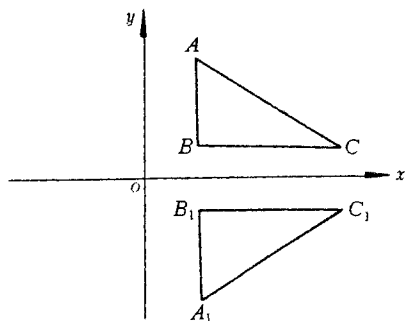


图 3-1-13 相对 x 轴对称变换

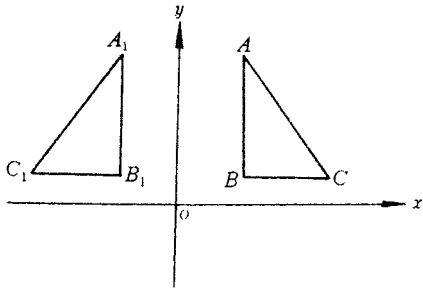


图 3-1-14 相对 y 轴对称变换

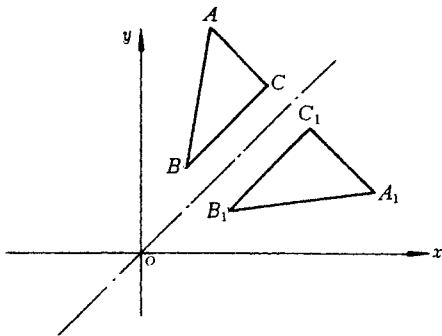


图 3-1-15 相对 45° 线对称变换

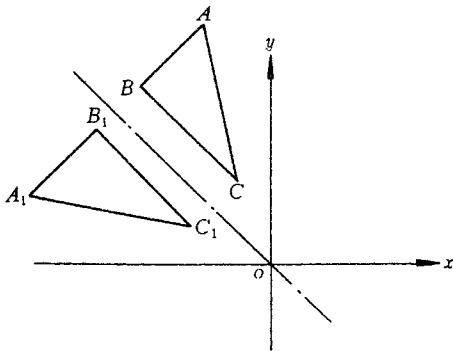


图 3-1-16 相对 -45° 线对称变换

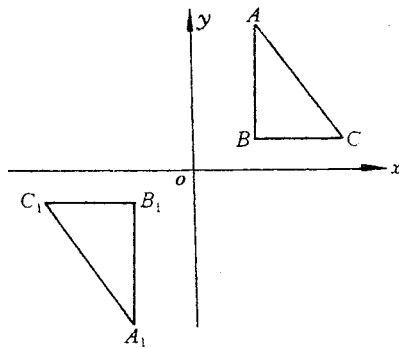


图 3-1-17 相对坐标原点对称变换

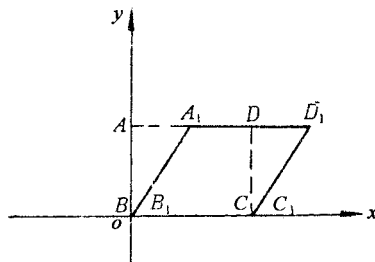
三、错切变换

(1) 沿 x 方向错切。当变换矩阵为

$$T = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} (c \neq 0) \text{ 时, 由式 (3-1-2) 有}$$

$$\begin{cases} x^* = x + cy \\ y^* = y \end{cases}$$

此时发生变换的特点是变换后 y 坐标保持不变, 而 x 坐标则变为 $x^* = x + cy$, 即坐标点沿 x 方向平移了 cy , 显然当 y 为 0 时, 平移量为 0, 当 y 取不同值时, 平移量也不同。这种变换称为相对 x 方向的错切变换。如图 3-1-18 所示, 矩形 $ABCD$ 沿 x 方向错切变换后, 形成棱形 $A_1B_1C_1D_1$ 。

图 3-1-18 沿 x 方向错切变换

(2) 沿 y 方向错切。当变换矩阵为 $T = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} (b \neq 0)$ 时, 由式 (3-1-2) 有

$$\begin{cases} x^* = x \\ y^* = bx + y \end{cases}$$

此时发生变换的特点是变换后 x 坐标保持不变,而 y 坐标则变为 $y^* = bx + y$,即坐标点沿 y 方向移动了 bx ,显然,当 $x = 0$ 时,移动量为 0。这种变换称为相对 y 方向的错切变换。如图 3-1-19 所示,矩形 $ABCD$ 沿 y 方向错切变换后,变为菱形 $A_1B_1C_1D_1$ 。

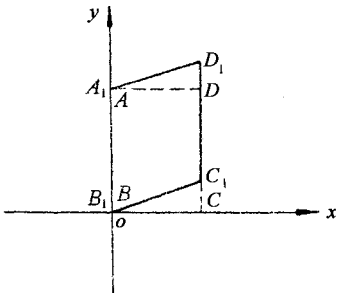


图 3-1-19 沿 y 方向错切变换

四、旋转变换

对一点 $(x \ y)$ 进行旋转变换是指绕坐标原点旋转 α 角,并且规定逆时针方向旋转时 α 取正值,顺时针方向旋转时 α 取负值。由解析几何学可知,点 $(x \ y)$ 旋转后的新坐标值是

$$\begin{cases} x^* = x \cos\alpha - y \sin\alpha \\ y^* = x \sin\alpha + y \cos\alpha \end{cases} \tag{3-1-3}$$

将式(3-1-3)与式(3-1-2)进行比较不难发现,旋转变换的变换矩阵应为

$$T = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

图 3-1-20 表示了 $\triangle ABC$ 经旋转变换后成为 $\triangle A_1B_1C_1$ 的情况。

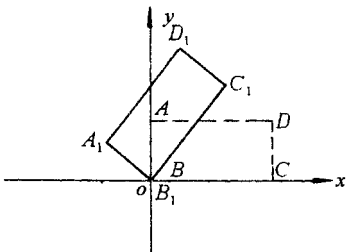


图 3-1-20 旋转变换

五、齐次坐标与齐次变换矩阵

上述的变换矩阵不能实现图形的平移变换,因为无论 a 、 b 、 c 、 d 为何值,都不能使图形平移。为了使图形产生平移,应该向变换矩阵中引入平移参数,这显然需要扩充变换矩阵。要使点向量能够与扩充了的变换矩阵进行矩阵乘法运算,同时也应扩充点向量,这就需要应用点的齐次坐标。

(一) 齐次坐标

在二维直角平面坐标系中,任意一点都可用一个二维矩阵 $\begin{bmatrix} x & y \end{bmatrix}$ 表示,如果将该点用一个三维坐标矩阵 $\begin{bmatrix} x_H & y_H & H \end{bmatrix}$ 表示,则称其为该点的齐坐标,其中 H 称为比例因子。依次类推, n 维空间坐标系中,点的齐次坐标用 $n+1$ 维空间坐标表示。

二维直角坐标与其齐次坐标的关系是

$$\begin{cases} x = x_H/H \\ y = y_H/H \end{cases}$$

即

$$x_H : y_H : H = x : y : 1$$

H 的取值是任意的,所以一点的齐次坐标也有多种表示形式。为保持坐标一致性,通常总是将 H 取为 1。

(二) 齐次变换矩阵

二维直角坐标点的齐次变换矩阵的一般形式为:

$$T = \begin{bmatrix} a & b & p \\ c & d & q \\ l & m & s \end{bmatrix}$$

变换矩阵可分为四个部份,其中 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 的作用是对被变换点的坐标进行缩放、对

称、错切和旋转变换, $\begin{bmatrix} l & m \end{bmatrix}$ 的作用是进行平移变换, $\begin{bmatrix} p \\ q \end{bmatrix}$ 是进行投影变换, s 是整个变换中比例系数。

例 3 若 $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l & m & 1 \end{bmatrix}$ 时,有

$$\begin{bmatrix} x_H & y_H & H \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l & m & 1 \end{bmatrix} = \begin{bmatrix} (x + l) & (y + m) & 1 \end{bmatrix}$$

上式相当于

$$\begin{cases} x_H = x + l \\ y_H = y + m \\ H = 1 \end{cases}$$

对应的直角坐标为

$$\begin{cases} x^* = x + l \\ y^* = y + m \end{cases}$$

此时发生的变换为平移变换,其中 l, m 分别为点在 x 和 y 方向的平移量。

例 4 若变换矩阵为 $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$, 有

$$\begin{bmatrix} x_H & y_H & H \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix} = \begin{bmatrix} x & y & s \end{bmatrix}$$

将变换后的齐次坐标转换为直角坐标,有

$$\begin{cases} x^* = x/s \\ y^* = y/s \end{cases}$$

显然此时发生的变换为缩放变换,其中 s 为比例因子。

下面列出二维图形的各种基本齐次变换矩阵。

(1) 恒等变换矩阵 $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(2) 比例变换矩阵 $T = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(3) 全比例变换矩阵 $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$

(4) 对 x 轴对称变换 $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$(5) \text{对 } y \text{ 轴对称变换 } T = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(6) \text{对 } 45^\circ \text{线对称变换 } T = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(7) \text{对 } -45^\circ \text{线对称变换 } T = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(8) \text{对坐标原点对称变换 } T = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(9) \text{沿 } x \text{ 向错切变换 } T = \begin{bmatrix} 1 & 0 & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(10) \text{沿 } y \text{ 向错切变换 } T = \begin{bmatrix} 1 & b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(11) \text{旋转变换矩阵 } T = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(12) \text{平移变换 } T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l & m & 1 \end{bmatrix}$$

六、基本变换矩阵的级联

对图形连续进行多个基本变换后,可以得到更为复杂的图形变换。换句话说,复杂的图形变换可以分解为一系列基本的图形变换。这种由多个基本变换组成复杂变换的方法叫做基本变换的级联或组合变换,相应的矩阵称为基本变换矩阵的级联矩阵或组合变换矩阵。

例 5 如图 3-1-21 所示,矩形 $ABCD$ 绕坐标原点以外的任意一点 $P(x_c, y_c)$ 旋转 α

角,求其变换矩阵。

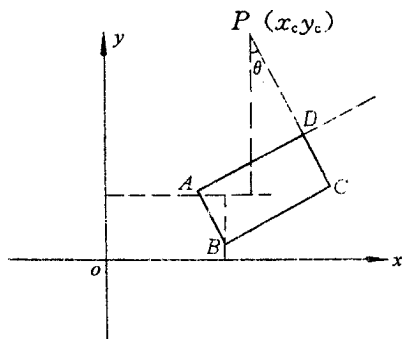


图 3-1-21 绕任意点旋转变换

解:该变换可以分解为下列基本变换:

(1)将旋转中心平移到坐标原点,平移量为 $-x_c$ 、 $-y_c$,平移变换矩阵为

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(2)使矩形 ABCD 绕坐标原点旋转 α 角,其变换矩阵为

$$T_2 = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3)将旋转中心平移回原来位置,变换矩阵为

$$T_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_c & y_c & 1 \end{bmatrix}$$

上述三个基本变换过程可表示为

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \cdot T = \begin{bmatrix} x & y & 1 \end{bmatrix} T_1 T_2 T_3$$

因此可知矩形 ABCD 绕 $P(x_c, y_c)$ 旋转的变换矩阵为:

$$\begin{aligned} T = T_1 T_2 T_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_c & y_c & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ x_c(1 - \cos\alpha) + y_c \sin\alpha & x_c \sin\alpha + y_c(1 - \cos\alpha) & 1 \end{bmatrix} \end{aligned}$$

例 6 如图 3-1-22 所示,求 $\triangle ABC$ 相对任意直线的对称变换矩阵,设直线的方程为 $Ax + By + C = 0$ 。

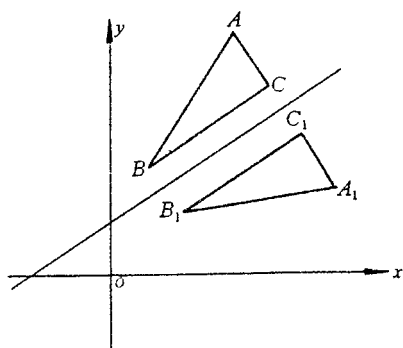


图 3-1-22 相对任意直线的对称变换

解:由直线的方程可以求出直线在 x 轴和 y 轴上的截距分别为 $-C/A$ 和 $-C/B$, 直线与 x 轴的夹角为 $\alpha = \arctg(-A/B)$ 。

该变换可以分解为下列基本变换:

(1)沿 x 轴方向平移 C/A , 使对称轴通过坐标原点, 变换矩阵为

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ C/A & 0 & 1 \end{bmatrix}$$

(2)绕坐标原点旋转 $-\alpha$ 角, 使对称轴和 x 轴重合, 变换矩阵为

$$T_2 = \begin{bmatrix} \cos(-\alpha) & \sin(-\alpha) & 0 \\ -\sin(-\alpha) & \cos(-\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3)对 x 轴进行对称变换, 变换矩阵为

$$T_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(4)绕坐标原点旋转 α 角, 变换矩阵为

$$T_4 = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(5)沿 x 轴方向平移 $-C/A$, 使对称轴回到原来位置, 变换矩阵为:

$$T_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -C/A & 0 & 1 \end{bmatrix}$$

综上所述,对任意直线的对称变换为

$$\begin{bmatrix} x & y & 1 \end{bmatrix} T = \begin{bmatrix} X & Y & 1 \end{bmatrix} T_1 T_2 T_3 T_4 T_5$$

由此得出变换矩阵为

$$T = T_1 T_2 T_3 T_4 T_5 = \begin{bmatrix} \cos 2\alpha & \sin 2\alpha & 0 \\ \sin 2\alpha & -\cos 2\alpha & 0 \\ \frac{C}{A}(\cos 2\alpha - 1) & \frac{C}{A}\sin 2\alpha & 1 \end{bmatrix}$$

有一点应该特别注意的是,基本变换的级联顺序是不能任意颠倒的。也就是说,级联的顺序不同,所得变换结果也可能不同。

例如,先平移后旋转的级联矩阵为

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l & m & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ l \cos \alpha - m \sin \alpha & l \sin \alpha + m \cos \alpha & 1 \end{bmatrix}$$

而先旋转后平移的级联矩阵为

$$T = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l & m & 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ l & m & 1 \end{bmatrix}$$

由此可见,二者的结果是不相同的。

第七节 图形裁剪

一、窗口与视区

(一) 用户坐标系与设备坐标系

在计算机图形系统中,用户坐标系是用户建立自己的图形时所采用的坐标系,其单位由用户自己确定。用户在建立或定义图形时,就是通过用户坐标系来描述图形数据的,即图形数据结构中的几何信息是以用户坐标系为基础的。

设备坐标系是与一个图形输入输出设备相关联的坐标系。例如,显示屏常以分辨率确定坐标单位,以左下角(或左上角)为坐标原点;绘图机绘图平面上也有自己的坐标系,即以某一角点为原点,以绘图精度(例如 0.1mm)为单位;同样地,数字化仪也有自己的坐标系。图形显示或绘图输出是以设备坐标系为基础的。

(二)窗口与裁剪

实际应用中经常会有一些范围大而且结构复杂的图形。在图形显示或绘图输出时,有时只需输出图形的局部内容,这时就需要指定图形的输出区域。

窗口是用户坐标系中的一个矩形区域,它可以用其左下角点和左上角点的坐标来表示,也可以用左下角点坐标和矩形的长宽来表示。窗口是用户根据需要任意确定的,其作用是限定图形输出区域。处于窗口内的图形在图形设备上显示或绘图输出,而对于窗口外的图形部分,将被去掉而不在图形设备上输出,这种处理方法就称为图形的裁剪。

(三)视区与视见变换

视区是设备坐标系中的一个矩形区域,同样地,可以用左下角点坐标和右上角点坐标表示一视区,也可用左下角点坐标和视区的长宽表示一个视区。视区的作用是在显示屏幕或绘图机的绘图平台上指定图形输出的区域。

通过变换可以把窗口中的图形与视区中的图形一一对应起来,也就是说把图形从用户坐标系输出到图形设备时,可以适当定义窗口和视区,使得窗口内的图形在视区内输出。这种变换过程称为视见变换。

如图 3-1-23 所示,设用户坐标系 xoy 中定义的窗口其左下角和右上角的坐标分别为 (x_l, y_l) 和 (x_r, y_r) , 在图形设备坐标系 UOV 中定义的视区的左下角和右上角的顶点坐标分别为 (U_l, V_l) 和 (U_r, V_r) , 窗口内的一点 $W(W_x, W_y)$ 在视区内的映象点为 $S(S_u, S_v)$, 则有以下关系式成立。

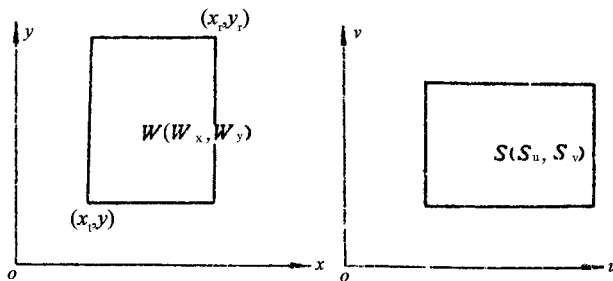


图 3-1-23 窗口与视区

$$\begin{cases} \frac{S_u - U_t}{U_r - U_t} = \frac{W_x - x_t}{X_r - X_t} \\ \frac{S_v - V_t}{U_r - V_t} = \frac{W_y - y_t}{y_r - y_t} \end{cases}$$

由此可推得

$$\begin{cases} S_u = \left(\frac{U_r - U_t}{X_r - X_t} \right) \cdot (W_x - X_t) + u_t \\ S_v = \left(\frac{V_r - V_t}{y_r - y_t} \right) \cdot (W_y - y_t) + V_t \end{cases} \quad (3-1-4)$$

式(3-1-4)即为窗口与视区的变换关系。

二、线段的裁剪

窗口把一幅图形分为窗口内和窗口外两部分。窗门内的图形是应该输出的,而窗口外的图形信息是应该去掉的。图形裁剪,就是要将图形在窗口内和窗口外的部分区分出来。裁剪的原理并不复杂,但工作量极大,因此对处理速度有较高的要求。

线段是组成一切其它图形的基础,所以线段的裁剪是最基本也是最重要的裁剪处理。下面介绍线段的两种裁剪算法。

(一) 矢量裁剪法

图 3-1-24 给出了任意一条线段与窗口可能有的相对位置关系。由图 3-1-24 可看出,线段要么完全位于窗口之外(图 3-1-24 中 a、b),要么完全位于窗口内或在窗口内唯一留下一个可见线段(图 3-1-24 中 c、d、e)。裁剪算法要找出可见线段,只需求出可见线段的两个端点就可以了。设窗口左下角和右上角顶点的坐标为 (x_l, y_l) 和 (x_r, y_r) ,并设窗口的四条边界把坐标平面分成九个区域。将这九个区域分别标上号码,其中窗口区号码为 0(图 3-1-25)。下面以图 3-1-25 中线段 EF 为例,说明矢量裁剪法的运算步骤。

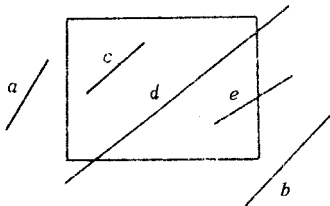


图 3-1-24 线段与窗口的关系

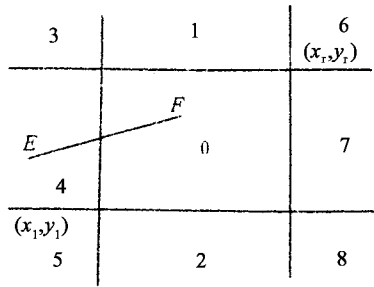


图 3-1-25 矢量裁剪法

设线段 EF 的两个端点的坐标分别为 (x_1, y_1) 和 (x_2, y_2) , 裁剪算法的运算步骤如下

(1) 若下列条件之一成立, 则 EF 不在窗口之内, 运算结束

$$\max(x_1, x_2) < x_l; \min(x_1, x_2) > x_r;$$

$$\max(y_1, y_2) < y_l; \min(y_1, y_2) > y_r;$$

(2) 如果 $x_l \leq x_1 \leq x_r$ 且 $y_l \leq y_1 \leq y_r$, 则 EF 的始点在 0 区内, 窗口内可见线段的新始点的坐标为 $x_s = x_1, y_s = y_1$; 否则, 线段 EF 与窗口的关系及其新始点坐标 (x_s, y_s) 的求解过程如下。

(3) 若 $x_1 < x_l$, 则有

$$\begin{cases} x_s = x_l \\ y_s = y_1 + (x_l - x_1)(y_2 - y_1)(x_2 - x_1) \end{cases} \quad (3-1-5)$$

若求得的 y_s 满足 $y_l \leq y_s \leq y_r$, 则 x_s, y_s 求解有效; 否则:

① 若 (x_1, y_1) 在 4 区, 则线段 EF 与窗口无交点, 运算结束;

② 若 (x_1, y_1) 在 5 区, 且 $y_s > y_r$, 或者当 (x_1, y_1) 在 3 区, 且 $y_s < y_l$ 时, 线段 EF 与窗口无交点。运算结束;

③ 若 $y_1 < y_r$, 有

$$\begin{cases} x_s = x_1 + (y_r - y_1)(x_2 - x_1)(y_2 - y_1) \\ y_s = y_r \end{cases} \quad (3-1-6)$$

若 $y_1 > y_r$, 则有

$$\begin{cases} x_s = x_1 + (y_l - y_1)(x_2 - x_1)(y_2 - y_1) \\ y_s = y_l \end{cases} \quad (3-1-7)$$

式 (3-1-6) 和式 (3-1-7) 中求出的 x_s 如果满足 $x_l \leq x_s \leq x_r$, 则交点有效; 否则, 线段 EF 与窗口无交点, 位于窗口之外, 运算结束。

(4)当 $x_1 > x_r$ 时,可用以上类似过程求出线段 EF 与窗口右边框的交点。

(5)当 (x_1, y_1) 在 1、2 区时,即可用式(3-1-6)和式(3-1-7)求出线段 EF 与窗口上、下两边框的交点。如求得的 x_s 满足 $x_l \leq x_s \leq x_r$,则求解有效,否则,线段 EF 与窗口无交点,即 EF 位于窗口之外。

(二)中点再分法

在前面介绍的矢量裁剪算法中,确定可见线段的端点,需要进行乘除运算,而大量的乘除运算会降低程序的执行速度。中点再分法的基本思想是以求线段的中点来代替求线段与窗口边框的交点,通过不断对分,排除线段在窗口外的部份,分别寻找线段两个端点各自对应的最远的可见点。求线段的中点只须用到加法和除以“2”的运算,而一个数除以“2”的运算,在计算机中可通过对该数存储码简单地右移一位来完成,运算速度大为提高。

线段两个可见端点的求出过程是相同的,下面以图 3-1-26 所示的线段 P_1P_2 为例,说明求离 P_1 最远的可见点的算法步骤。

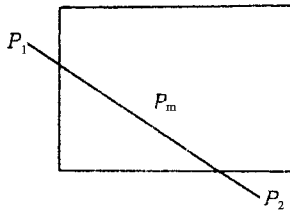


图 3-1-26 中点再分法

(1)测试 P_2 是否在窗口内,如是,则 P_2 就是离 P_1 最远的可见点,运算结束。否则,进行下一步。

(2)测试 P_1P_2 是否在窗口同侧的外面,若是, P_1P_2 全部不可见,运算结束。否则,进行下一步。

(3)取 P_1P_2 中点 P_m ,若 P_mP_2 在窗口同侧的外面,则以 P_m 代替 P_2 重复步骤(2)。否则,以 P_m 代替 P_1 重复步骤(2)。

三、多边形的裁剪

平面多边形是由若干线段围成的平面封闭图形。对多边形的裁剪,不能简单地用重复对线段的裁剪来完成,主要原因是对平面多边形的裁剪结果也应是多边形。因此,裁剪多边形要解决的问题有:一个完整的封闭多边形经裁剪后一般不再是封闭的,这时需

要用窗口边界的适当部分来封闭它,需要判断窗口的四个交点在裁剪中是否要与其它交点连线。图 3-1-27 给出了多边形裁剪的几种情况。

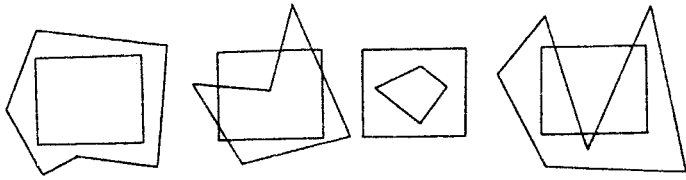


图 3-1-27 多边形与窗口的关系

多边形裁剪已有多重算法,下面介绍其中的 Suther Land-Hodgman 算法。

该算法的基本思想是依次用窗口的四条边框对多边形进行分步裁剪。先用一条边框直线对整个多边形进行裁剪,得到一个或者若干个新的多边形;再用第二条边框裁剪这些新产生的多边形,依次类推,直到用第四条边框裁剪完毕,整个多边形的裁剪也就完成了。

如图 3-1-28 所示,设有多边形 $A_1 A_2 \dots A_n$,对每个顶点 $A_i (i = 1, 2, \dots, n)$ 测试 $A_{i-1} A_i$ (令 $A_0 = A_n$) 与窗口 L_j 是否相交。若相交则求出交点 B_{i-1} 作为输出顶点,再判断 A_i 本身是否在窗口区域内,在区域内的也作为输出顶点,否则舍弃。对 L_j 裁剪的输出又作为对 L_{j+1} 边框裁剪的输入。经过四次裁剪,就得到所要的结果。图 3-1-28 表示了裁剪过程。多边形 $A_1 A_2 A_3 A_4 A_5$ 经过图 3-1-28a、图 3-1-28b、图 3-1-28c、图 3-1-28d 所示的四次裁剪,最后得到多边形 $A_0 A_1 A_2 A_3 B_3 B_5$ 。

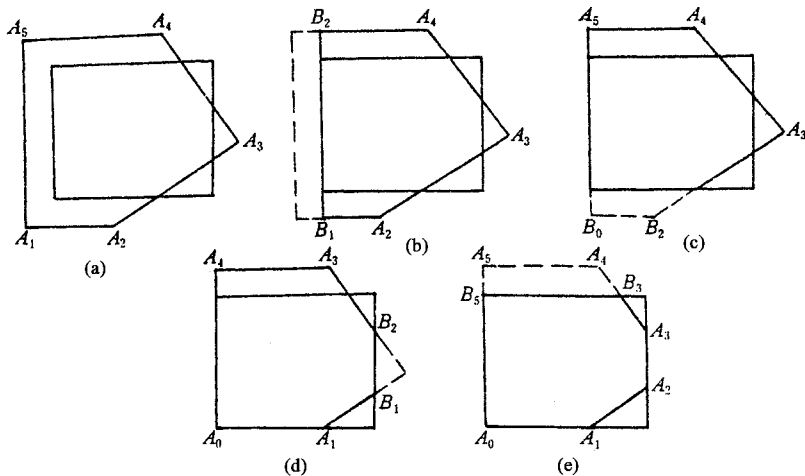


图 3-1-28 多边形裁剪

该算法首先把待裁剪多边形的各顶点按照其构成多边形的顺序,组成“顶点序列”,并且用数组存储起来,这个数组称为待处理的输入数组。如图 3-1-28a 所示,输入数组存储的内容为顶点 $A_1 A_2 A_3 A_4 A_5$,相继连接 $A_1 A_2$ 、 $A_2 A_3$ 、 $A_3 A_4$ 、 $A_4 A_5$ 、 $A_5 A_1$,就构成了待裁剪的多边形。程序对输入数组的顶点序列进行处理,经过一次处理后产生一组新的顶点序列,并用另一个数组存储,称为输出数组。新的顶点序列就表示了经一次裁剪后形成的新的多边形。如图 3-1-28b 所示,经第一次裁剪后,输出的顶点序列为 $B_1 A_2 A_3 A_4 B_4$,新的多边形为 $B_1 A_2 A_3 A_4 B_4 B_1$,经裁剪产生的输出数组,又成为下一次裁剪的输入数组,直到第四次裁剪完成,其输出数组表示了最后形成的多边形。

下面说明输入数组的处理过程。设输入数组存储的顶点序列为 $A_i (i = 1 \sim n)$,裁剪窗口的边界为 l 。在处理顶点序列时,以裁剪边 l 依次检验顶点序列中的每个顶点的位置:处于 l 边可见侧的顶点被列入新产生的顶点序列中,以便于最后输出;而处于 l 边不可见侧的顶点则被删除掉。除了检验每个顶点位于 l 边的哪一侧外,还要检测 A_i 点和它的前一点 A_{i-1} 点是否处于 l 边的同侧。如果不处于同侧,那么就必须求出 l 边和直线段 $A_i A_{i-1}$ 的交点,这个交点作为新的多边形顶点输入到新产生的顶点序列中。当最后一点 A_n 检验完毕后,再检验 $A_n A_1$ 是否与 l 边相交。如果相交,则同样应求出交点,并把交点作为新的多边形顶点列入新的顶点序列中。

第二章 常用图形算法

本章针对矿山工程图形的特点,介绍一些常用的图形计算方法。

第一节 交点的计算

一、两条线段的交点

设有直线段 AB 和 CD ,其端点坐标分别为 $A(x_A, y_A)$ 、 $B(x_B, y_B)$ 、 $C(x_C, y_C)$ 、 $D(x_D, y_D)$,由解析几何学可知线段 AB 的参数方程为

$$\begin{cases} x = x_A + (x_B - x_A)t \\ y = y_A + (y_B - y_A)t \end{cases} \quad (0 \leq t \leq 1)$$

线段 CD 的参数方程为

$$\begin{cases} x = x_C + (x_D - x_C)\tau \\ y = y_C + (y_D - y_C)\tau \end{cases} \quad (0 \leq \tau \leq 1)$$

如果 AB 和 CD 相交,则在交点处有

$$\begin{cases} x_A + (x_B - x_A)t = x_C + (x_D - x_C)\tau \\ y_A + (y_B - y_A)t = y_C + (y_D - y_C)\tau \end{cases} \quad (3-2-1)$$

设

$$\Delta = \begin{vmatrix} x_B - x_A & -(x_D - x_C) \\ y_B - y_A & -(y_D - y_C) \end{vmatrix}$$

如果 $\Delta = 0$,由解析几何学知直线 AB 与 CD 平行或重合。如果 $\Delta \neq 0$,由式(3-2-1)可得

$$t = \frac{1}{\Delta} \begin{vmatrix} x_C - x_A - (x_D - x_C) \\ y_C - y_A - (y_D - y_C) \end{vmatrix}$$

$$t = \frac{1}{\Delta} \begin{vmatrix} x_B - x_A & x_C - x_A \\ y_B - y_A & y_C - y_A \end{vmatrix}$$

当 $0 \leq t \leq 1$ 和 $0 \leq \tau \leq 1$ 同时成立时, AB 和 CD 有一个交点, 交点坐标为

$$\begin{cases} x = x_A + (x_B - x_A)t \\ y = y_A + (y_B - y_A)\tau \end{cases}$$

否则, 交点只是在 AB 和 CD 的延长线上。

二、直线段与圆弧的交点

设圆弧的中心为 (x_C, y_C) ,半径为 R ,其参数方程为

$$\begin{cases} x = x_C + R \cos \alpha \\ y = y_C + R \sin \alpha \end{cases} \quad (\alpha_1 \leq \alpha \leq \alpha_2)$$

直线 AB 的参数方程为

$$\begin{cases} x = x_A + (x_B - x_A)t \\ y = y_A + (y_B - y_A)t \end{cases} \quad (0 \leq t \leq 1)$$

显然, 交点处的参数 t, α 应满足

$$\begin{cases} x_A + (x_B - x_A)t = x_C + R \cos \alpha \\ y_A + (y_B - y_A)t = y_C + R \sin \alpha \end{cases}$$

消去 α ,得

$$\left[\frac{(x_A - x_C) + (x_B - x_A)t}{R} \right]^2 + \left[\frac{(y_A - y_C) + (y_B - y_A)t}{R} \right]^2 = 1 \quad (3-2-2)$$

整理后得

$$At^2 + Bt + C = 0$$

其中 $A = (x_B - x_A)^2 + (y_B - y_A)^2$

$$B = 2(x_A - x_C)(x_B - x_A) + 2(y_A - y_C)(y_B - y_A)$$

$$C = (x_A - x_C)^2 + (y_A - y_C)^2 - R^2$$

如果 $B^2 - 4AC < 0$,则式(3-2-2)没有根, 直线段与圆弧不相交。否则, 由式(3-2-2)求出 t 的两个实根 t_1, t_2 ,此时, 若 $0 \leq t_i \leq 1$ ($i = 1, 2$) ,将 t_i 代入下式

$$\cos\alpha_i = \frac{x_A - x_C + (x_B - x_A)t_i}{R}$$

$$\sin\alpha_i = \frac{y_A - y_C + (y_B - y_A)t_i}{R}$$

可求出 α_i 、若 $\alpha_1 \leq \alpha_i \leq \alpha$ 成立,则由 $t = t_i$ 或者 $\alpha = \alpha_i$ 可得直线和圆弧的一个交点。否则,所求点不是真正的交点,而仅仅是直线段所在直线与圆弧所在圆的交点。

三、线段与拟合曲线的交点

拟合曲线是根据给定的数据样本点,建立起合适的数学模型,通过计算机求得一系列逼近的插值点的坐标,再用一系列微小的直线段连接这些插值点而形成的曲线。拟合曲线。在工程设计中有着广泛的用处。常见的拟合曲线有最小二乘法拟合曲线、三次样条曲线、Bézier 曲线、B 样条曲线等。

直线段与拟合曲线的交点通常用近似解法。一种适用于所有拟合曲线的计算方法是求直线段与组成拟合曲线的所有微小直线段的交点,这些有效交点就是直线段与拟合曲线的交点。

四、圆弧与圆弧的交点

设圆弧 A 的中心点为 (x_A, y_A) , 半径为 R_A , 其参数方程为:

$$\begin{cases} x = x_A + R_A \cos\theta \\ y = y_A + R_A \sin\theta \end{cases} \quad (\theta \in U) \quad (3-2-3)$$

圆弧 B 的中心点为 (x_B, y_B) , 半径为 R_B , 其参数方程为

$$\begin{cases} x = x_B + R_B \cos\alpha \\ y = y_B + R_B \sin\alpha \end{cases} \quad (\alpha \in V) \quad (3-2-4)$$

在交点处, α, θ 应满足

$$\begin{cases} x_A + R_A \cos\theta = x_B + R_B \cos\alpha \\ y_A + R_A \sin\theta = y_B + R_B \sin\alpha \end{cases} \quad (3-2-5)$$

消去 θ , 得

$$\left[\frac{(x_B - x_A) + R_B \cos\alpha}{R_A} \right]^2 + \left[\frac{(y_B - y_A) + R_B \sin\alpha}{R_A} \right]^2 = 1 \quad (3-2-6)$$

令

$$H^2 = (x_B - x_A)^2 + (y_B - y_A)^2$$

$$\cos\varphi = \frac{x_B - x_A}{H} \quad \sin\varphi = \frac{y_B - y_A}{H}$$

则式 (3-2-6) 可以表达为

$$H^2 + 2R_B(x_B - x_A)\cos\alpha + 2R_B(y_B - y_A)\sin\alpha + R_B^2 - R_A^2 = 0 \quad (3-2-7)$$

即

$$\frac{H^2 + (R_B^2 - R_A^2)}{2R_B H} + \cos(\varphi - \alpha) = 0$$

由此可知, 当 $\left| \frac{H^2 + (R_B^2 - R_A^2)}{2R_B H} \right| \leq 1$ 时, 令 $\delta = \arccos \frac{H^2 + (R_B^2 - R_A^2)}{2R_B H}$

则式 (3-2-7) 有解

$$\alpha_1 = \varphi + \pi + \delta \pmod{2\pi}$$

$$\alpha_2 = \varphi + \pi - \delta \pmod{2\pi}$$

而由式 (3-2-5) 可以求得 $\theta_1 \theta_2$ 。如果 $\theta_i \in U$, 同时 $\alpha_i \in V$, 则由式 (3-2-3) 或式 (3-2-4) 可求得交点 (x_i, y_i) 。否则, α_i 不是交点参数。而当 $\delta = 0$ 或 $\delta = \pi$ 时, (x_i, y_i) 是二圆弧的切点。

五、线段与平面的交点

在绘制矿山地质剖面图时, 经常要求线段与平面的交点。

设空间直线 AB 的两个端点坐标为 $A(x_A, y_A, z_A)$ 和 $B(x_B, y_B, z_B)$, 其解析式为

$$\frac{x - x_A}{l} = \frac{y - y_A}{m} = \frac{z - z_A}{n} = t \quad (3-2-8)$$

其中 $l = \frac{x_B - x_A}{H}, m = \frac{y_B - y_A}{H}, n = \frac{z_B - z_A}{H}$

$$H = [(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2]^{1/2}$$

设另有一平面过不共线的三点 $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$ 和 $P_3(x_3, y_3, z_3)$, 其解析式为

$$Ax + By + Cz + D = 0 \quad (3-2-9)$$

其中

$$A = \begin{vmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix} \quad B = \begin{vmatrix} z_1 & x_1 & 1 \\ z_2 & x_2 & 1 \\ z_3 & x_3 & 1 \end{vmatrix} \quad C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad D = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

如果 $Al + Bm + Cn = 0$, 那么 AB 与平面平行或重合。否则, 将式 (3-2-8) 改写为

$$\begin{cases} x = x_A + lt \\ y = y_A + mt \\ z = z_A + nt \end{cases} \quad (3-2-10)$$

将式(3-2-10)代入式(3-2-9),有

$$A(x_A + l_1) + B(y_A + m_1) + C(z_A + n_1) + D = 0$$

由此可得交点处的参数值为

$$t_0 = \frac{Ax_A + By_A + Cz_A + D}{Al + Bm + Cn}$$

从而可得交点处的坐标值为

$$\begin{cases} x_0 = x_A + lt_0 \\ y_0 = y_A + mt_0 \\ z_0 = z_A + nt_0 \end{cases}$$

如果 $x_A \leq x_0 \leq x_B$, $y_A \leq y_0 \leq y_B$, $z_A \leq z_0 \leq z_B$ 成立,则交点为有效点。否则,所求点为直线 AB 的延长线与平面的交点。

第二节 包含与重叠的检测

在图形处理和分析过程中,经常会涉及到判别一个点与一个区域或一个区域与另一个区域的相互关系等问题。这里只讨论区域为平面多边形时的情况。通常矿山工程图中的曲线在图形处理时可以将其离散为一系列很微小的直线段,因此可将由曲线围定的区域视为平面多边形。

一、点在区域内的判断

设有一个由平面多边形 $ABCDE$ 围成的区域,另有一个点 P 。判断 P 点是否位于区域内,称为点的包含性检测。

(一)射线法

如图 3-2-1 所示,过 P 点向任意方向作一条射线 PM ,然后求出 PM 与多边形的边相交的点数。如果交点数为偶数(包括交点数为 0),说明点 P 一定在多边形 $ABCDE$ 之外(图 3-2-1a);若交点数为奇数, P 点就一定在多边形之内(图 3-2-1b)。射线的方向可以是任意的,但为了计算方便,通常使其与某一坐标轴平行。此外在计算交点数时,如果 PM 恰好穿过多边形的某一个顶点时,如果多边形中与这个顶点相邻的两条边在 PM 的同侧,则此交点无效;在异侧时,交点有效。

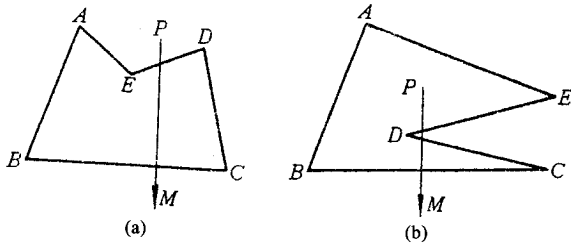


图 3-2-1 包含性射线检验法

下面给出算法的执行步骤。设有平面多边形 Q , 平面点 $P(x_p, y_p)$

(1) 从点 $P(x_p, y_p)$ 引一条与 y 轴平行的射线 PM , 射线方程为 $x = x_p$ 。令交点个数

$N_p = 0$

(2) 粗判 Q 上的边 l_i 是否与 PM 相交。设 l_i 的两端点坐标为 (x_i, y_i) 和 (x_{i+1}, y_{i+1}) 。

如果 $y_i > y_p$ 且 $y_{i+1} > y_p$

或者 $x_i < x_p$ 且 $x_{i+1} < x_p$

或者 $x_i > x_p$ 且 $x_{i+1} > x_p$

那么此时 l_i 与 PM 无交点, 转 (4)。否则, 转 (3)。

(3) 细判。 l_i 的参数方程为

$$\begin{cases} x = x_i + (x_{i+1} - x_i)t \\ y = y_i + (y_{i+1} - y_i)t \end{cases} \quad (0 \leq t \leq 1)$$

将射线 PM 的方程 $x_i = x_p$ 代入上式, 求得交点 (x_p, y_o) 。由于已经过了第 (2) 步的初判, 故交点 (x_p, y_o) 肯定包含在 l_i 的区间上。此时进一步分析。

若交点 $y_o > y_p$, 交点无效, 转 (4)。

若 $y_o < y_p$ 且 $x_i < x_p < x_{i+1}$ 或 $x_i > x_p > x_{i+1}$, 交点有效, $N_p = N_p + 1$, 转 (4)。

若 $y_o < y_p$ 且 l_i 在 P 点右侧, 即 $x_i > x_p, x_{i+1} > x_p$, 交点有效, $N_p = N_p + 1$, 转 (4)。

若 $y_o < y_p$ 且 l_i 在 P 点左侧, 即 $x_i < x_p, x_{i+1} < x_p$, 交点无效, 转 (4)。

(4) 判断 Q 上是否还有未检测的边, 若有, 转 (3)。若所有边已检查完毕, 则检查 N_p 的奇偶性, 若 N_p 为奇数, 则 P 在 Q 内。反之, P 在 Q 外。

(5) 分类运算结束。

(二) 夹角之和法

如图 3-2-2 所示, 依次将 P 点分别与 A 、 B 、 C 、 D 、 E 各点相连, 令 α_i 依次为

$\angle APB, \angle BPC, \dots, \angle EPA$ 。如果 $\sum_{i=1}^5 d_i = 0$ 则 P 点在多边形 $ABCDE$ 之外(图 3-2-2a); 如果 $\sum_{i=1}^5 d_i = \pm 2\pi$ 则 P 点在多边形 $ABCDE$ 之内(图 3-2-2b)。

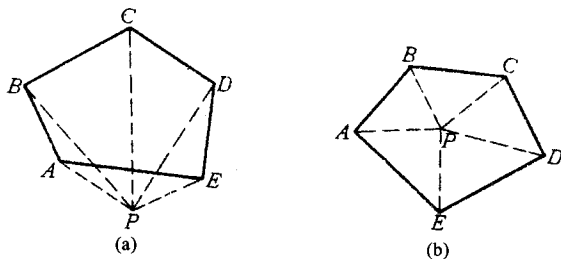


图 3-2-2 夹角之和检验法

二、线段与区域的重叠

线段与区域的重叠是指线段与区域的公共部分或线落在区域内的部分。

设区域 Q 与直线段 AB 所在直线相交于 C 和 D (图 3-2-3)。已知直线段 AB 的参数方程为

$$\begin{cases} x = x_A + (x_B - x_A)t \\ y = y_A + (y_B - y_A)t \end{cases} \quad (0 \leq t \leq 1)$$

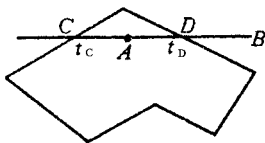


图 3-2-3 线段与区域有二个交点

则交点 C, D 的参数值为 t_C, t_D 且 $t_C \leq t_D$ 。于是 线段 AB 与区域 Q 有重叠的充分必要条件是

$$(t_C, t_D) \cap (0, 1) \neq \emptyset$$

即 $(t_C, t_D) \cap (0, 1) = [\max(t_C, 0), \min(t_D, 1)] \neq \emptyset$

亦即 $\max(t_C, 0) < \min(t_D, 1)$

于是有

$$\begin{cases} t_C < t_D \\ t_C < 1 \\ t_D > 0 \end{cases}$$

当直线与区域有多于二个交点时 (图 3-2-4),可以把交点按其 t_i 大小排序,即 $t_1 \leq t_2 \leq \dots \leq t_{2n}$ 。如果直线穿过多边形的顶点,这时按交点两侧的边线是否在直线 AB 的同一侧来确定交点个数。在同一侧时算两个交点,不在同一侧时按一个交点计算。这样可以测试落入区域中的每个线段是否同 AB 重叠,即测试每个参数区间是否满足 $(t_{2k-1}, t_{2u}) \cap (0, 1) \neq \phi$ 。

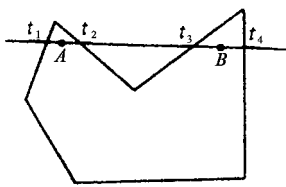


图 3-2-4 线段与区域有多个交点

三、区域与区域的重叠

区域与区域的重叠问题实际上是两个区域的边界求交点的问题。在详细测试之前可先作否定性测试,也就是利用两个区域的外接矩形进行快速测试。如图 3-2-5 所示,设区域 Q_A 的外接矩形为 $R_A: X_{1A} \leq X \leq X_{2A}, y_{1A} \leq y \leq y_{2A}$, 区域 Q_B 的外接矩形为 $R_B: X_{1B} \leq X \leq X_{2B}, y_{1B} \leq y \leq y_{2B}$ 。于是, Q_A 与 Q_B 有重叠部分的必要条件是 R_A 与 R_B 有重叠部分,而 R_A 与 R_B 有重叠部分的必要条件是 $[X_{1A}, X_{2A}]$ 与 $[X_{1B}, X_{2B}]$ 有重叠部分和 $[y_{1A}, y_{2A}]$ 与 $[y_{1B}, y_{2B}]$ 有重叠部分。即

$$\begin{cases} \max(X_{1A}, X_{1B}) \leq \min(X_{2A}, X_{2B}) \\ \max(y_{1A}, y_{1B}) \leq \min(y_{2A}, y_{2B}) \end{cases}$$

上面二式中,只要有一个不成立, Q_A 与 Q_B 就不会重叠

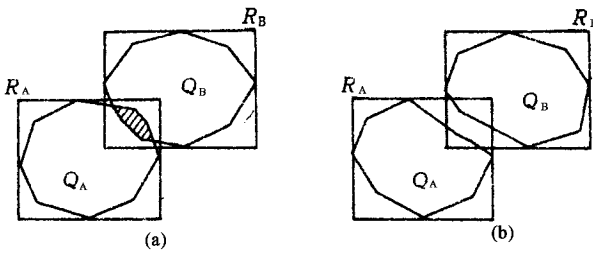


图 3-2-5 区域的相互关系

下面详细讨论两个多边形的重叠判断问题。设 A_1, A_2, \dots, A_n 与 B_1, B_2, \dots, B_m

分别是两个多边形区域 A 和 B 的顶点。判断 A 与 B 是否重叠,就要判断 A 的任一条边 $A_i A_{i+1}$ ($i = 1, 2, \dots, n, A_{n+1} = A_1$) 与 B 的任一条边 $B_j B_{j+1}$ ($j = 1, 2, \dots, m, B_{m+1} = B$) 是否相交,其计算量很大。如果 A, B 都是凸多边形,计算量将大为减少。检验方法如下。

(1) 令 $x_1 = \max\{\min(x_{A_i}), \min(x_{B_j}) \mid 1 \leq i \leq n, 1 \leq j \leq m\}$,

$x_2 = \min\{\max(x_{A_i}), \max(x_{B_j}) \mid 1 \leq i \leq n, 1 \leq j \leq m\}$ 。

如果 $x_1 > x_2$ 则 A 与 B 不重叠。

(2) 用过多边形 A 顶点的平行 y 轴的直线可以把区域 $\min(x_{A_i}) \leq x \leq \max(x_{A_i})$ ($i = 1, \dots, n$) 最多分为 $n-1$ 个直条;用过多边形 B 顶点的平行 y 轴的直线可以把区域 $\min(x_{B_j}) \leq x \leq \max(x_{B_j})$ ($j = 1, \dots, m$) 最多分为 $m-1$ 个分条(图 3-2-6)。

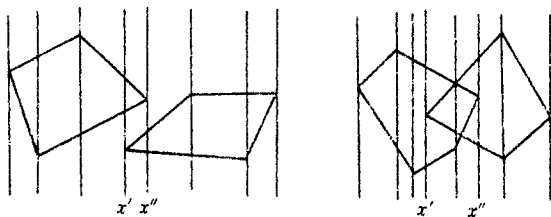


图 3-2-6 多边形重叠检验

只有在 $x_1 \leq x \leq x_2$ 区域中的每一直条包含四条边,两条属于 A ,另两条属于 B 。

(3) 取一个具有四条边的直条 $x' \leq x \leq x''$,如果四条边中, A 的边与 B 的边相交,则必与 B 必有重叠部分;如果不相交,则有 $x = x'$ 与 A 的边截出区间 $[y_{A1}, y_{A2}]$ 与 B 的边截出区间 $[y_{B1}, y_{B2}]$ 。若 $[y_{A1}, y_{A2}]$ 与 $[y_{B1}, y_{B2}]$ 重叠则 A 与 B 重叠。否则, A 与 B 不重叠。

第三节 凸包的计算

包含一个平面点集 S 的最小凸区域(区域内任意二点的连线总在区域内)称为 S 的凸包(图 3-2-7)。由定义可知凸包有以下特点(1)凸包的顶点必属于 S 。即凸包是由 S 中的若干个顶点构成的凸多边形(2)凸包的面积应是阻含点集 S 的最小区域。下面介绍凸包的两种计算方法。

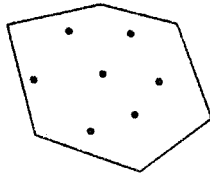


图 3-2-7 凸包

一、凸包的回溯算法

(1) 任意取二个点 P_1, P_2 。

(2) 把其它顶点 P_i 按 $P_1 P_i$ 与 $P_i P_2$ 的夹角(反时针方向为正)增加的次序排序。夹角相同时,以 $P_1 P_i$ 的距离由小到大排序,并把排序结果用双向链表链接。

(3) 依次检查链表相邻的三个顶点 P_i, P_{i+1}, P_{i+2} , 若由 $P_i P_{i+1}$ 到 $P_{i+1} P_{i+2}$ 要向左转,则顶点由 P_i 前进到 P_{i+1} , 若向右转,则删去顶点 P_{i+1} , 并退回到链表中前一个顶点 P_{i-1} 。

(4) 经历了所有顶点后,在双向链表中留下的顶点就是凸多边形。

具体过程可见图 3-2-8,最后算出的凸包是多边形 $P_1 P_2 P_3 P_5 P_7 P_1$ 。

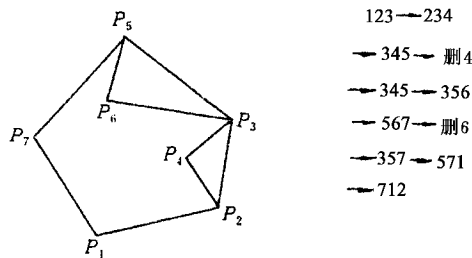


图 3-2-8 回溯法求凸包

二、非回溯算法

(1) 求点集 S 的外接矩形 $P_1 P_2 P_3 P_4$ 。若 $P_i (i = 1, 2, 3, 4)$ 皆属于 S , 则矩形 $P_1 P_2 P_3 P_4$ 就是凸包。

(2) 否则对每一个不属于 S 的角点 P_i , 在 $P_{i-1} P_i$ 与 $P_i P_{i+1} (P_0 = P_4, P_5 = P_1)$ 边上必有属于 S 的顶点, 把最靠近 P_i 的点分别记为 P'_i 和 P''_i 。

(3) 如果在 $P'_i P'_i$ 外(即 P_i 所在一侧)已没有 S 的顶点, 则 $P'_i P''_i$ 就是凸包的边。否则在 $P'_i P''_i$ 处 S 的顶点中取 P'''_i , 使 $\angle P'''_i P'_i P''_i$ 最大, 得凸包边 $P''_i P'_i$ 。以 P'''_i 代

P'_i 重复本步骤 ,直到新的凸包边之外已没有 S 的顶点为止。

具体过程见图 3-2-9。

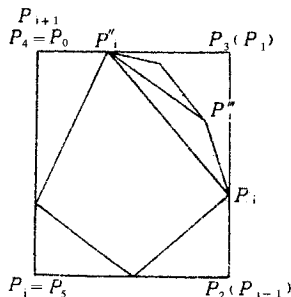


图 3-2-9 非回溯法求凸包

第四节 图形拾取算法

在图形交互设计过程中 ,用户经常需要通过移动图形光标在屏幕上选择即拾取需要删除或修改的图形 ,图形系统则在存储图形的数据结构中找到存储该图形信息的地址 ,以便对该图形进行处理。一般情况下 ,用户传递给计算机的信息是所要拾取的图形的类型(如点、线、曲线、字符等)和用于指定图形的光标坐标。

下面介绍几种常用图形拾取算法。设光标坐标值为 $P(x_g, y_g)$,由于光标和图形都是显示在屏幕上的 ,所以这里提到的坐标都是指屏幕坐标。

一、点的拾取

对于屏幕上的一点 $A(x_A, y_A)$,如果下式成立

$$(x_A - x_0)^2 + (y_A - y_0)^2 \leq r^2$$

说明 A 点即是要拾取的点 ,上式中的 r 是交互系统设定的领域精度。

二、字符串的拾取

屏幕上显示的字符串 ,通常都设有显示基点。因此拾取字符串 ,也就是拾取基点 ,这实际上仍然是点的拾取问题 ,方法同上。

三、直线段的拾取

设屏幕上一条直线段 AB 的端点坐标为 $A(x_A, y_A)$ 和 $B(x_B, y_B)$, 该线段的显示领域可近似如图 3-2-10 所示。

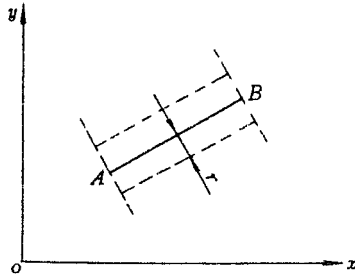


图 3-2-10 直线段的拾取领域

直线 AB 的方程为

$$(y_A - y_B)x - (x_A - x_B)y - y_A(x_A - x_B) + y_A(x_A - x_B) = 0$$

直线 M 和 N 的斜率为

$$K_B = \frac{x_A - x_B}{y_A - y_B} \quad (\text{设 } y_A - y_B \neq 0)$$

斜率为 K_B 的直线族可表示为 $y = K_B x + b$ 。若分别将 P_0, A, B 代入斜率为 K_B 的直线族方程, 得

$$\begin{cases} b_0 = y_0 - K_B x_0 \\ b_A = y_A - K_B x_A \\ b_B = y_B - K_B x_B \end{cases}$$

对于 $y_A - y_B = 0$ 的情况, $b_0 = x_0, b_A = x_A, b_B = x_B$ 。

如果 $\min(b_A, b_B) \leq b_0 \leq \max(b_A, b_B)$, 则 P_0 在 M, N 所夹的区域中。下面再判断 P_0 到直线段 AB 的距离是否小于或等于系统设定的领域精度 r , 即

$$\frac{|(y_A - y_B)x_0 - (x_A - x_B)y_0 - (y_A - y_B)x_A + (x_A - x_B)y_A|}{[(x_A - x_B)^2 + (y_A - y_B)^2]^{1/2}} \leq r$$

若上述两条件同时满足, 线段 AB 就是所要拾取的线段。

四、折线集的拾取

对于折线集的拾取方法是依次判断每条直线段的显示领域是否包含拾取点。如果

某折线集中的一条线段满足拾取条件,该折线集就是要拾取的对象。

五、曲线的拾取

曲线在显示时,已离散成折线集,所以曲线的拾取算法与折线集的拾取算法相类似。

第五节 面积的计算

设多边形的顶点序列为 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$,那么该多边形围成的区域的面积 S 为

$$S = \frac{1}{2} \sum_{i=1}^n (x_i \cdot y_{i+1} - x_{i+1} \cdot y_i)$$

式中 $x_{n+1} = x_1, y_{n+1} = y_1$ 。

第三章 矿山工程常用曲线的生成方法

在矿山地质采矿工程图件中,需要用各种曲线来表示煤层及各种地质结构的边界,需要用各种等值线来表示煤层空向几何形态、煤质指标的空间分布以及地形地貌等。这些曲线通常都是根据地质勘测等工作获得的离散数据而绘制的。这里介绍几种适用于矿山工程图形的常用曲线的生成方法。

第一节 曲线的基础知识

一、插值与逼近

由离散的数据点构造光滑连续的曲线,通常有插值和逼近两种作法。

(1)插值。即曲线函数通过离散数据点,同时在各点的一阶导数或二阶导数是连续的。插值方法通常用于离散数据比较准确的情况。

(2)逼近。曲线函数不一定通过给定的离散点,但在一定误差要求下靠近诸离散点。逼近法通常适用于数据误差较大的场合。

二、曲线的参数矢量表示

曲线的几何表示可分为参数和非参数两类。非参数表达式有显式方程 $y = f(x)$ 和隐式方程 $f(x, y) = 0$ 两种。在计算机绘图系统中,采用参数方式更为方便。

一般平面曲线的参数表达式为

$$\begin{cases} x = f(t) \\ y = g(t) \end{cases}$$

矢量表示形式为

$$P(t) = (f(t), g(t))$$

例如,对于图 3-3-1 所示的抛物线,其参数矢量表达式为 $P(t) = A_0 + A_1 t + A_2 t^2$ ($0 \leq t \leq 1$)。这条抛物线是由一系列以 A_0 位置为起点,在 A_1 方向上取 t ,在 A_2 方向上取 t^2 的点形成的曲线。

设 A_0 、 A_1 、 A_2 的 x 、 y 分量分别为 (a_0, b_0) 、 (a_1, b_1) 、 (a_2, b_2) ,则可改写出下列抛物线参数形式

$$\begin{cases} x = a_0 + a_1 t + a_2 t^2 \\ y = b_0 + b_1 t + b_2 t^2 \end{cases} (0 \leq t \leq 1)$$

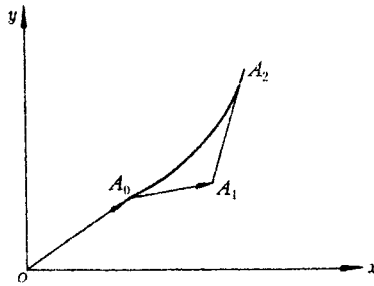


图 3-3-1 抛物线曲线段的参数矢量形式

三、曲线的光滑条件

由离散点构造曲线,不仅离散点之间的曲线段应该是光滑的,而且曲线段在连接点处也应是光滑的(图 3-3-2)。

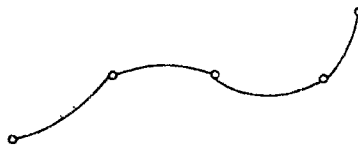


图 3-3-2 曲线的光滑连接

要使曲线段本身是光滑的,通常是采用二次曲线段或三次曲线段。要使曲线段在连

接处也保持光滑,则应该保证连接点处的两曲线段的切线斜率相等,即有 $P_i'^- = P_i'^+$,称这种连接条件为一阶导数连续或 c^1 级连续。

如果在节点处除了满足 $P_i'^- = P_i'^+$ 外,还满足 $P_i''^- = P_i''^+$,即连结点附近两侧的二阶导数也相等,称这种连续条件为 c^2 级连续或二阶导数连续。 c^2 级连续比 c^1 级连续更为光滑。

第二节 三次参数样条曲线

一、力学背景

在生产实践中常常需要用一条光滑的曲线连接一组经试验或计算得来的数据点。一种可行的办法是用一根弹性细木条,让它依次经过这些点,并在这些点处把木条压住,然后沿着木条画出一条光滑的曲线。这根木条称为样条,上述过程称为放样。

如果把样条看作是弹性细梁,压住的各点看作是简支点或集中荷载,由材料力学知道梁变形曲线的微分方程为

$$R(x) = \frac{M(x)}{EJ}$$

式中, $M(x)$ 为弯矩函数; E 为弹性模量; J 为几何惯性矩; $R(x)$ 为曲率。在梁的弯曲度较小的情况下, $R(x) \approx y'(x)$,从而有

$$y''(x) = \frac{M(x)}{EJ}$$

由于在两个压住点之间没有外力作用,故 $M(x)$ 是 x 的线性函数。由此可见,在两点之间,曲线 $y = y(x)$ 是三次多项式。如果把这样的若干条曲线段按 c^2 连续条件连接起来,就构成了整条样条曲线。

二、三次参数样条曲线

设有一个点列 $P_i(x_i, y_i), i = 1, 2, \dots, n$ (图 3-3-3)建立三次参数样条曲线方程。

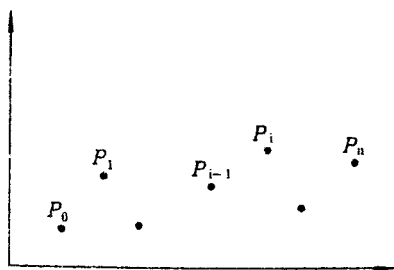


图 3-3-3 由插值点绘制三次参数样条曲线

(一) 表达式

在每一个小区间 $[P_i, P_{i+1}]$ 曲线是三次多项式, 以 t 为参变量的表达形式为

$$\begin{cases} P(t) = B_0 + B_1 t + B_2 t^2 + B_3 t & 0 \leq t \leq t_i \\ t_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2} & i = 1, 2, \dots, n-1 \end{cases} \quad (3-3-1)$$

式中 t_i 为第 i 段曲线的弦长。因此, 式 (3-2-9) 称为以弦长为参数的样条函数。系数 $B_j (j=0, 1, 2, 3)$ 取决于 P_i, P_{i+1} 以及两端点的切向量 P'_i 和 P'_{i+1} 。

将式 (3-2-9) 对 t 微分得

$$\begin{cases} P'(t) = B_1 + 2B_2 t + 3B_3 t^2 \\ P''(t) = 2B_2 + 6B_3 t \end{cases}$$

所以有

$$\begin{cases} P(0) = P_i = B_0 \\ P(t_i) = P_{i+1} = B_0 + B_1 t_i + B_2 t_i^2 + B_3 t_i^3 \\ P'(0) = P'_i = B_1 \\ P'(t_i) = P'_{i+1} = B_1 + 2B_2 t_i + 3B_3 t_i^2 \end{cases}$$

由此可得

$$\begin{cases} B_0 = P_i \\ B_1 = P'_i \\ B_2 = \frac{1}{2} (P_{i+1} - P_i) / t_i^2 - (P'_{i+1} + 2P'_i) / t_i \\ B_3 = -\frac{1}{6} (P_{i+1} - P_i) / t_i^2 + (P'_{i+1} + P'_i) / t_i \end{cases}$$

将式 (3-2-10) 代入式 (3-2-9) 得

$$P(t) = \left[1 - 3\left(\frac{t}{t_i}\right)^2 + 2\left(\frac{t}{t_i}\right)^3 \right] P_i + \left[3\left(\frac{t}{t_i}\right)^2 - 2\left(\frac{t}{t_i}\right)^3 \right] P_{i+1} + \left[\left(\frac{t}{t_i}\right) - 2\left(\frac{t}{t_i}\right)^2 + \left(\frac{t}{t_i}\right)^3 \right] \cdot$$

$$P'_{i+1} + \left[- \left(\frac{t}{t_i} \right)^2 + \left(\frac{t}{t_i} \right)^3 \right] P'_{i+1} \quad (0 \leq t \leq t_i)$$

令

$$\begin{aligned} f_1(t) &= 1 - 3 \left(\frac{t}{t_i} \right)^2 + 2 \left(\frac{t}{t_i} \right)^3 \\ f_2(t) &= 3 \left(\frac{t}{t_i} \right)^2 - 2 \left(\frac{t}{t_i} \right)^3 \\ g_1(t) &= \left(\frac{t}{t_i} \right)^2 - 2 \left(\frac{t}{t_i} \right)^2 + \left(\frac{t}{t_i} \right)^3 \\ g_2(t) &= - \left(\frac{t}{t_i} \right)^2 + \left(\frac{t}{t_i} \right)^3 \end{aligned}$$

则式(3-3-1)可整理为

$$P(t) = f_1(t)P_i + f_2(t)P_{i+1} + g_1(t)P'_i + g_2(t)P'_{i+1} \quad (3-3-2)$$

式中 $f_1(t)$ 、 $f_2(t)$ 、 $g_1(t)$ 、 $g_2(t)$ 称为权函数。

(二) 连续方程

对于三次样条曲线, 要求 $P(t)$ 在条节点处二阶导数连续, 故有

$$P''_{i-1}(t_{i-1}) = P''_i(0)$$

即:

$$\begin{aligned} t_i P'_{i+1} + 2(t_{i-1} + t_i)P'_i + t_{i-1}P'_{i+1} &= \frac{3t_i}{t_{i-1}}(P_i - P_{i-1}) + \frac{3t_{i-1}}{t_i}(P_{i+1} - P_i) \\ (i &= 2, 3, \dots, n-1) \end{aligned} \quad (3-3-3)$$

式(3-3-3)即为样条的连续方程, 共有 $n-2$ 个。

(三) 边界条件

上述的连续方程共有 $n-2$ 个线性方程组, 但共有 n 个未知数, 故对方程组的求解还得考虑边界条件。边界条件是由实际问题的要求对曲线两端 P_0 和 P_n 提出的约束条件, 主要有三种。

1. 夹持端

夹持端边界条件是直接给定两端的切线方向 P' 和 P'_n (图 3-3-4a)。

2. 抛物端

抛物端边界条件是首段和末端为抛物线, 即其二阶导数为常数。

根据此条件有

$$P''_1 = P''_2 \quad P''_{n-1} = P''_n$$

从而有

$$P'_1 + P'_2 = \frac{2}{t_1}(P_2 - P_1) \quad P'_{n-1} + P'_n = \frac{2}{t_{n-1}}(P_n - P_{n-1})$$

抛物端边界见图 3-3-4。

3. 自由端

自由端边界条件是端点处的二阶导数为零,即

$$P''_1 = 0, P''_n = 0$$

从而有

$$2P'_1 - P'_2 = \frac{3}{t_1}(P_2 - P_1)$$

$$P'_{n-1} + 2P'_n = \frac{3}{t_{n-1}}(P_n - P_{n-1})$$

自由端边界条件见图 3-3-4c。

(四) 连续方程求解

$n-2$ 个连续方程加上两个边界条件,可组成 n 阶线性方程组,由此可求出 n 个切矢的值,从而得到各段三次参数样条插值公式。

以抛物端边界条件为例,组成的线性方程组如下:

$$\begin{bmatrix} 1 & 1 & & & & & & \\ t_2 & 2(t_1 + t_2) & t_1 & & & & & \\ & t_3 & 2(t_2 + t_3) & t_2 & & & & \\ & & & & \dots & & & \\ & & & & & t_n & 2(t_{n-1} + t_n) & t_{n-1} \\ & & & & & & 1 & 1 \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ \vdots \\ P'_{n-1} \\ P'_n \end{bmatrix} = \begin{bmatrix} \frac{2}{t_1}(P_2 - P_1) \\ \frac{3t_2}{t_1}(P_2 - P_1) + \frac{3t_1}{t_2}(P_3 - P_2) \\ \frac{3t_3}{t_2}(P_3 - P_4) + \frac{3t_2}{t_3}(P_4 - P_3) \\ \vdots \\ \frac{3t_n}{t_{n-1}}(P_{n-1} - P_{n-2}) + \frac{3t_{n-1}}{t_n}(P_n - P_{n-1}) \\ \frac{2}{t_{n-1}}(P_n - P_{n-1}) \end{bmatrix} \quad (3-3-4)$$

下面用“追赶法”求解式(3-3-4)。

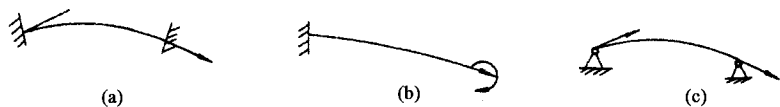


图 3-3-4 边界条件

将式 (3-3-4) 改写为下式：

$$\frac{t_i}{t_{i-1} + t_i} P'_{i-1} + 2P'_i + \frac{t_{i-1}}{t_{i-1} + t_i} P'_{i+1} = \frac{1}{t_{i-1} + t_i} \left[\frac{3t_i}{t_{i-1}} (P_i - P_{i-1}) + \frac{3t_{i-1}}{t_i} (P_{i+1} - P_i) \right] \quad (i = 2, 3, \dots, n-1, t_i = |P_{i+1} - P_i|) \quad (3-3-5)$$

令

$$a_i = \frac{t_i}{t_{i-1} + t_i} \quad c_i = \frac{t_{i-1}}{t_{i-1} + t_i} \quad P_i = \frac{1}{t_{i-1} + t_i} \left[\frac{3t_i}{t_{i-1}} (P_i - P_{i-1}) + \frac{3t_{i-1}}{t_i} (P_{i+1} - P_i) \right]$$

则将式 (3-3-5) 整理为

$$a_i P_{i-1} + 2P'_i + c_i P'_{i+1} = P_i \quad (3-3-6)$$

令 $D_1 = \frac{2}{t_1} (P_2 - P_1)$, $D_n = \frac{2}{t_{n-1}} (P_n - P_{n-1})$ 则式 (3-3-4) 可改写为以下形式

$$\begin{bmatrix} 1 & q_1 & & & \\ a_2 & 2 & c_2 & & \\ & a_3 & 2 & c_3 & \\ & & \dots & & \\ & & & a_{n-1} & 2 & c_{n-1} \\ & & & & 1 & q_n \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ \vdots \\ P'_{n-1} \\ P'_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{n-1} \\ D_n \end{bmatrix} \quad (3-3-7)$$

其中 $q_1 = 1$, $q_n = 1$ 。将上述矩阵展开, 则第一和第二个方程可写为：

$$\begin{cases} P'_1 + q_1 P'_2 = D_1 \\ a_2 P'_1 + 2P'_2 + c_3 P'_3 = D_2 \end{cases} \quad (3-3-8)$$

进一步将式 (3-3-8) 改写为

$$\begin{cases} P'_1 + q_1 P'_2 = R_1 \\ P'_2 + q_2 P'_3 = R_2 \end{cases}$$

其中 $R_1 = D_1$; $q_2 = \frac{c_2}{2 - a_2 q_1}$; $R_2 = \frac{D_2 - a_2 R_1}{2 - a_2 q_1}$

类似地, 可将式 (3-3-7) 中的第 i 个方程改写为

$$P'_i + q_i P'_{i+1} = R_i \quad (i = 1, 2, \dots, n-1)$$

其中 $q_i = \frac{c_i}{2 - a_i q_{i-1}}$, $R_i = \frac{D - a_i R_{i-1}}{2 - a_i q_{i-1}}$ 这样就可把式 (3-3-7) 改写为

$$\begin{bmatrix} 1 & q_1 & & & & \\ & 1 & q_2 & & & \\ & & 1 & q_3 & & \\ & & & \dots & & \\ & & & & 1 & q_{n-1} \\ & & & & 1 & q_n \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ \vdots \\ P'_{n-1} \\ P'_n \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_{n-1} \\ R_n \end{bmatrix} \quad (3-3-9)$$

通过求解最后两个联立方程,得

$$P'_n = (R_n - R_{n-1}) / (q_n - q_{n-1})$$

依次向上类推,有

$$P'_i = R_i - q_i P'_{i+1} \quad (i = n-1, n-2, \dots, 1)$$

最后可求出所有的切矢即一阶导数值。

(五) 计算机程序

下面给出绘制三次参数样条曲线的计算机程序,程序由 FORTRAN 语言编写。程序中的参量与变量的含义如下:

P——数组,存放数据点的坐标;

N——数据点数;

K——各曲线段插值点数;

KS——标志,始端边界条件时 KS=1,夹持端 KS=2,自由端 KS=3;

KE——末端边界条件标志,意义同 KS;

V₁₁, V₁₂——始端切矢分量,

VN₁, VN₂——末端切矢分量;

F₁、F₂、G₁、G₂——权函数;

下面为程序清单,其中的子程序 PLOT(x, y, Key)为绘图机画线子程序,当 Key=3 时表示把画笔移至(x, y)点处,当 Key=2 时,表示由当前位置画线至(x, y)处。

```
SUBROUTINE SPLINX(P,N,K,KS,KE)
  DIMENSION P(2,N),T(50),A(50),C(50),D(2,50),
    Q(50),R(2,50),V(2,50)
```

```

N1 = N - 1
DO 10 I = 1 ,N1
10  T(I)=SQR((R(1,I+1)-R(1,I))* * 2 + (R(2,I+1) -R(2,I))* * 2)
DO 20 I = 2 ,N ,
A(I)= T(I)/(T(I-1)+T(I))
Q(I)= 1.0 - A(I)
DO 20 J = 1 ,2
20  D(J,I)- 3. * A(I) * (R(J,I)-R(J,I-1))/T(I-1)+ 3. * Q(I)*(R(J,I+
1)-R(J,I))/T(I)
GO TO(30 ,40 ,50 ),KS
30  Q(I)= 0.0
READ(* ,* )V11 ,V12
R(1,I)= V
R(2,I)= V12
GOTO 60
40  Q(I)= 1 ,0
DO 45 J = 1 ,2
45  R(J,I)= 2. *(R(J,2)-R(J,I))/T(I)
GOTO 60
50  Q(I)= 0.5
DO 55 J = 1 ,2
55  R(J,I)= 1.5 * (R(J,2)-R(J,I))/T(I)
60  DO 70 I = 2 ,N1
W = 2.0 - A(I)* Q(I-1)
Q(I)= Q(I)/W
DO 70 J = 1 ,2
70  R(J,I)=(D(J,I)- A(I)* R(J,I-1))/W
GO TO(80 ,90 ,100 ),KE
80  READ(* ,* )VN1 ,VN2
V(1,N)= VN1
V(2,N)= VN2

```

```

GOTO 110
90 DO 95 J = 1 ,2
95 V(J ,N)=(2. *( R(J ,N)-R(J ,N1 ))/T(N1)-R(J ,N1 ))*(1.0-Q(N1 ))
GOTO 110
100 DO 105J= 1 ,2
105 V(J ,N)=(3. *( R(J ,N)-R(J ,N1 ))/T(N1)-R(J ,N1 ))*(2.-Q(N1 ))
110 DO 120 II = 1 ,N1
I = N - II
DO 120J= 1 ,2
120 V(J ,I)=R(J ,I)-Q(I)* V(J ,I+1)
CALL PLOT(R(1 ,I),R(2 ,I),3)
W = 1./ (K+1. )
DO 200 I= 1 ,N1
U = W
CALL PLOT(R(1 ,I),R(2 ,I),2)
DO 200J= 1 ,K
F1 = 1.0-3.0* U * U +2.0* U * U * U
F2 = 3.0* U * U -2.0* U * U * U
G1 = U -2.0* U * U + U * U * U
G2 = - U * U + U * U * U
X=R(1 ,I)* F1 +R(1 ,I+1)* F2 +V(1 ,I)* G1 * T(I)+V(1 ,I+1)* G2 * T
(I)
Y=R(2 ,I)* F1 +R(2 ,I+1)* F2 +V(2 ,I)* G1 * T(I)+V(2 ,I+1)* G2 * T
(I)
CALL PLOT(X ,Y ,2)
U = U + W
200 CONTINUE
CALL PLOT(R(1 ,N),R(2 ,N),2)
RETURN
END

```

第三节 等值线的绘制

等值线图在矿山有着广泛的用处,常用的等值线图有煤层顶底板等高线图、地形等高线图以及各种煤质指标等值线图等,等值线一般具有以下特点:

- (1) 一条等值线通常是连续曲线;
- (2) 给定值后,等值线可以不限一条;
- (3) 等值线可以是封闭的,也可能与域的边界相交;
- (4) 等值线一般互不相交。

等值线图通常也是根据一系列离散数据点绘制的。绘制等值线的一般作法是首先把绘图区域划分为矩形网格,然后根据一定的换算方法,将对象由离散点描述转换为网格节点描述,判断等值线与每一网格的各边是否相交,如果有交点,计算出交点坐标,再将交点按顺序连接起来。

下面讨论已知网格数据点,在矩形区域 R 内绘制等值线的问题。设矩形区域划分为 $(m-1) \times (n-1)$ 个网格,网格点的序号排列为横向 $j = 1, 2, \dots, n$, 纵向 $i = 1, 2, \dots, m$,

一、等值点的计算

(一) 等值线与网格边相交的条件

等值线是否与网格的某一边相交,取决于这条边的两个端点的高值是否夹有等值线的高值。对于网格边 AB ,如果满足

$$z(A) \geq z \geq z(B) \text{ 或 } z(A) \leq z \leq z(B)$$

网格边 AB 一定与等值线相交。式中 z 表示等值线高程值, $z(A)$ 和 $z(B)$ 分别表示 A 、 B 端点的高值。

(二) 交点坐标的计算

等值线与网格边的交点即是所要计算的等值点。

设网格边的端点为 A 、 B ,其高值为 $z(A)$ 和 $z(B)$,对图 3-3-5a 所示情况,有

$$\frac{z(B) - z(A)}{X_B - X_A} = \frac{z - z(A)}{X - X_A}$$

可得交点坐标

$$\begin{cases} x = x_A + \frac{z - z(A)}{z(B) - z(A)} (x_B - x_A) \\ y = y_A = y_B \end{cases}$$

对于图 3-3-5b 所示的情况,有

$$\frac{z(B) - z(A)}{y_B - y_A} = \frac{z - z(A)}{y - y_A}$$

交点坐标为

$$\begin{cases} x = x_A = x_B \\ y = y_A + \frac{z - z(A)}{z(B) - z(A)} (y_B - y_A) \end{cases}$$

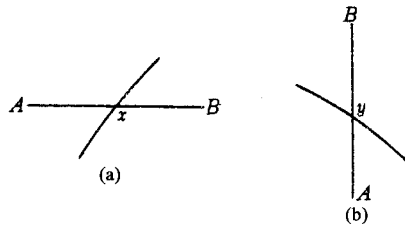


图 3-3-5 等值线交点

二、等值线的起点 终点及分枝

在及域内,等值线有两种情况,其一是等值线是封闭的,起点和终点重合(图 3-3-6 中的曲线 Q);其二是等值线是不封闭的,其特点是起自某个边界而又终止于边界,称为开曲线(图 3-3-6 中的曲线 L)。绘制封闭曲线时,保留起点,每画一点都与起点比较,当返回起点时,一条等值线就完成了;绘制开曲线时,每画一点都与边界条件比较,判断是否与边界相交。

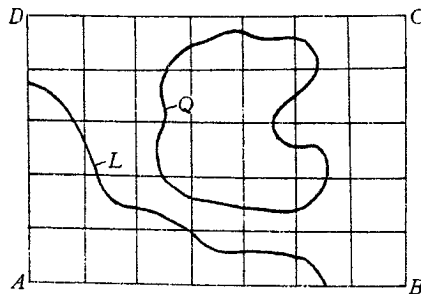


图 3-3-6 等值线图

寻找等值线的步骤是先边界,后域内。如图 3-3-6,设域 R 的边界顶点为 A, B, C, D 。当给定等值线高程值后,可先求出 AB, BC 上网格各竖边是否与等值线相交,若相交记相交边为“1”,同时求出交点,若不相交,则对网格边作标记“-2”,然后求 DC ,再一一求各水平边上是否有交点,并作“1”或“-2”的标记。

在绘制等值线时,可从边界最下边开始搜索,每过一个交点便将标记“1”改为“-2”。一条等值线绘完,重新扫描网格的水平边,看是否还有标记为“1”的边,如果有,说明等值线有另一分枝,需要继续绘制。如果所有标记均为“-2”,说明这一高程值的等高线绘制完毕。

三、等值线的追踪

找到等值线的起点后,接下来就应该从起点开始,追踪等值线的走向。

(一) 确定等值线进入网格时的走向

如图 3-3-7 所示,等值线进入网格的走向有四种可能:自下而上进入上网格;自左向右进入右网格;自上而下进入下网格;自右向左进入左网格。

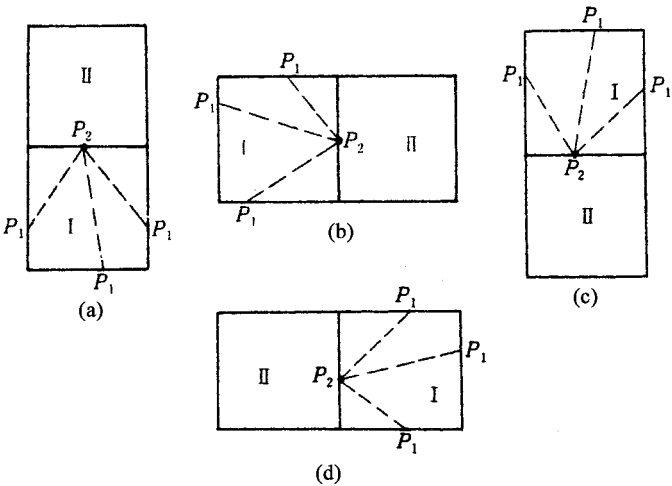


图 3-3-7 等值线的走向

设 P_2 点为等值线从网格 I 进入网格 II 的等值点, P_1 为前一等值点, 又设 (x_1, y_1) 和 (x_2, y_2) 分别为 P_1 点和 P_2 点的坐标, 那么

$$y_1 < y_2, y_2 < y_1, x_1 < x_2, x_2 < x_1$$

可分别作为自下而上, 自上而下, 自左向右和自右向左追踪等值走向的判别式。

(二) 确定等值线的出走方向

显然, 等值线进入网格后, 只会通过另外三边中的某条边走出。图 3-3-8 表明, 当网

格四边上都有等值点存在时,会出现出走方向不确定情况。因此,应根据问题的一般原则,对出走方向作适当的规定。

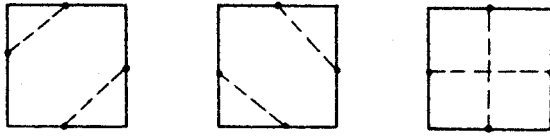


图 3-3-8 等值线走向的不确定现象

确定等值线方向遵循的原则是:首先考虑等值线原来的前进方向,后应考虑与当前等值点的远近。在图 3-3-9a 中,当前等值点 P_2 位于网格 (j, i) 的下边。设下一个等值点为 P_3 。 P_3 点的位置可能有三种情况,即 P_{31} 、 P_{32} 、 P_{33} ,而实际只能选择其中的一种。

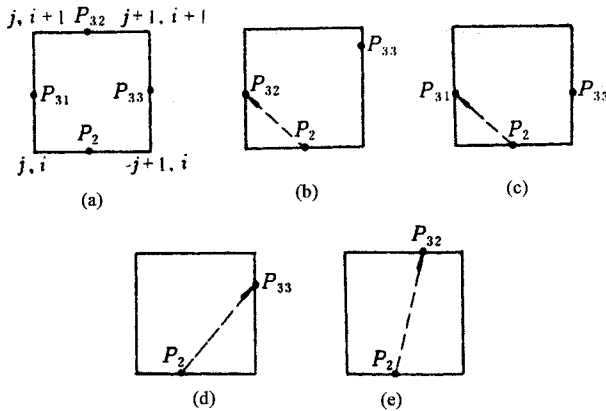


图 3-3-9 等值点的追踪

选择 P_{31} 、 P_{32} 、 P_{33} 的原则如下:

- (1) 当 P_{31} 、 P_{33} 都存在时,选择靠近网格底边的点为 P_3 (图 3-3-9b);
- (2) 若 P_{31} 、 P_{33} 靠近网格底边的距离相同,则选择与 P_2 点距离近者为 P_3 (图 3-3-9c);
- (3) 当 P_{31} 、 P_{33} 中只存在一种时,存在的点即为 P_3 (图 3-3-9d);
- (4) 当无 P_{31} 、 P_{33} 时,对边必存在 P_{32} 作为 P_3 (图 3-3-9e)。

以上讨论的是自下而上的情况,其余三种情况可按照类似方法处理。

(三) 网格点与等值点相同时的处理方法

如果网格点的数据与高程值相同,此时等高线通过网格点。而该网格点同时又是四个相邻网格的公点,在绘制等高线时会发生重复使用和方向错乱问题。解决这一问题的方法

是对网格点加上一个足够小的数值予以修正,使等值点与网格点错开。

(四) 计算机程序

下面给出网格法绘制等值线的计算机程序 程序由 FORTRAN 语言编写。

图 3-3-10 为程序框图。程序中的子程序 WF 的作用是搜索开曲线、子程序 FY 的作用是追踪一个等值点,子程序 FZ 的功能是计算等值点绘图坐标,子程序 CONTOU 的功能是把等值点连接成等值线。

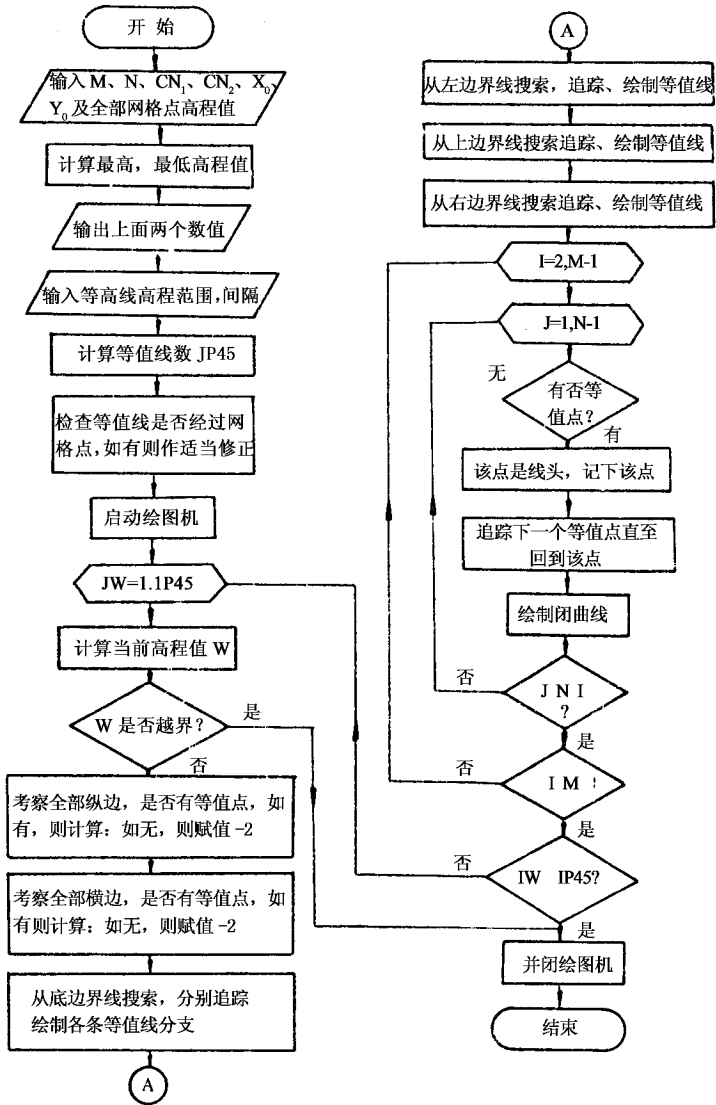


图 3-3-10 等值线程序框图

程序运行前应建立一个存储网格数据值的数据文件。程序运行时需要从键盘输入下列数据：

M 横向最大网格序号数

N 纵向最大网格序号数

CN₁ ,CN₂ :每一网格的横边和纵边的长度

X₀、Y₀ 矩形区域的左下角坐标

K :由等值点连接等值线 所要求给定的两值点之间的插值点数

P₄、P₅ :最小高程值和最大高程值

P₆ :等高线高程值增量

```

DIMENSION X( 50 ,50 ),Y( 50 ,50 ),BX( 50 ,50 ),DW( 2 ,50 )COMMON X ,Y ,DW ,J1 ,J1 ,
J2 ,J2 ,J3 ,J3 ,A1 ,A2 ,A3 ,B1 ,B2 ,B3 ,CN1 ,CN2 ,NN
WRITE( * , 5 )
5  FORMAT( 1X , 'M ,N ,CN1 ,CN2 ,X0 ,Y0 ,K = ' )
   READ( * , * )M ,N ,CN1 ,CN2 ,X0 ,Y0 ,K
   OPEN( 7 ,FILE = 'SHDATA ' ,STATUS = 'OLD' )
   READ( 7 ,10 )( BX( J ,I ) ,J = 1 ,N ) ,I = 1 ,M )
10  FORMAT( 8F4.1 )
   DO 15 I = 1 ,M
   DO 15 J = 1 ,N
   Y( J ,I ) = 0.0
15  X( J ,I ) = 0.0
   CJ4 = 99999.
   CJ5 = - CJ4
   DO 20 I = 1 ,M
   DO 20 J = 1 ,N
   IF( CJ4 · LE. BX( J ,I ) )GOTO 22
   CJ4 = BX( J ,I )
22  IF( CJ5 · GE. BX( J ,I ) )GOTO 20
   CJ5 = BX( J ,I )
20  CONTINUE

```

```

WRITE( *, 12 ) CJ4 , CJ5
12  FORMAT( 1X , 'CJ4 = ' , F7.2 , 'AX ' , CJ5 = ' , F7.2 )
WRITE( *, 14 )
14  FORMAT( 1X , 'P4 , P5 , P6 = ' )
READ( *, * ) P4 , P5 , P6
IP45 = IFIX( ( P5 - P4 ) / P6 )
DO 25 IW = 1 , IP45
W = P4 + P6 * ( IW - 1 )
DO 25 I = 1 , M
DO 25 J = 1 , N
IF( B( J , I ) - W .NE. 0. ) GOTO 25
B( J , I ) = W + 0 .001
25  CONTINUE
M1 = M - 1
N1 = N - 1
CALL GINIT
CALL SET( X0 , Y0 , 1. , 1. , 0 .0 )
DO 300 IW = 1 , IP45
W = P4 + P6 * ( IW - 1 )
IF( W .LT. CJ4 ) GOTO 300
IF( W .GT. CJ5 ) GOTO 300
DO 30 J = 1 , N
DO 30 I = 1 , M1
IF( B( J , I ) - B( J , I + 1 ) .NE. 0. ) GOTO 18
17  Y( J , I ) = - 2.0
GOTO 30
18  Y( J , I ) = ( W - B( J , I ) ) * ( B( J , I + 1 ) - B( J , I ) )
IF( Y( J , I ) .LE. 0. .OR. Y( J , I ) .GE. 1. ) GOTO 17
30  CONTINUE
DO 35 I = 1 , M

```

```

DO 35 J = 1 ,N1
IF( BX( J ,I )- BX( J + 1 ,I ). NE. 0. )GOTO 33
32  X( J ,I )= - 2. 0
    GOTO 35
33  X( J ,I )=( W - BX( J ,I ))( BX( J + 1 ,I )- BX( J ,I ))
    IF( X( J ,I ). LE. 0. . OR. X( J ,I ). GE. 1. )GOTO 32
35  CONTINUE
    DO 40 J = 1 ,N1
    IF( X( J ,I ). LE. 0. )GOTO 40
    I1 = 0
    I2 = 1
    J1 = J
    J2 = J
    CALL WK( 1. ,M ,N )
    CALL CONTOU( DW ,NN ,K ,I )
40  CONTINUE
    DO 50 I = 1 ,M1
    IF( Y( I ,I ). LE. 0. )GOTO 50
    I1 = I
    J1 = 0
    I2 = I
    J2 = 1
    CALL WK( 0. ,M ,N )
    CALL CONTOU( DW ,NN ,K ,I )
50  CONTINUE
    DO 60 J = 1 ,N1
    IF( X( J ,M ). LE. 0. )GOTO 60
    I1 = M
    J1 = J
    I2 = M

```

$J_2 = J$

$A_2 = (J_2 - 1 + 0.5) * CN_1$

CALL WK(1, ,M ,N)

CALL CONTOU(DW ,NN ,K ,I)

60 CONTINUE

DO 70 I = 1 ,M₁

IF(Y(N ,I). LE. 0.) GOTO 70

I₁ = I

J₁ = N

J₂ = I

J₂ = N

$A_2 = (J_2 - 1 - 0.1) * CN_1$

CALL WK(0. ,M ,N)

CALL CONTOU(DW ,NN ,K ,I)

70 CONTINUE

DO 100 J = 2 ,N₁

DO 100 I = 1 ,M₁

IF(Y(J ,I). LE. 0) GOTO 100

I₁ = I

J₁ = 0

J₂ = I

J₂ = J

CALL FY

J₁ = J₂

I₁ = I₂

I₂ = I₃

J₂ = J₃

A₂ = A₃

B₂ = B₃

JE = J₁

$$IE = I_1$$

$$A_1 = (J_1 - 1) * CN_1$$

$$B_1 = (I_1 - 1 + Y(J_1, I_1)) * CN_2$$

$$DW(1, 1) = A_1$$

$$DW(2, 1) = B_1$$

$$DW(1, 2) = A_2$$

$$DW(2, 2) = B_2$$

$$L = 0$$

105 CALL FY

$$A_1 = A_2$$

$$A_2 = A_3$$

$$B_1 = B_2$$

$$B_2 = B_3$$

$$I_1 = I_2$$

$$I_2 = I_3$$

$$J_1 = J_2$$

$$J_2 = J_3$$

$$L = L + 1$$

$$NN = L + 2$$

$$\text{IF} (\text{ABS}(A_2 - DW(1, 1)) \cdot LE \cdot 0.001 \cdot \text{AND} \cdot \text{ABS}(B_2 - DW(2, 1))$$

$$LE \cdot 0.001) \text{GOTO } 107$$

$$DW(1, NN) = A_2$$

$$DW(2, NN) = B_2$$

GOTO 105

107 (JE, IE) = -2.0

$$NN = NN - 1$$

CALL CONTOU(DW, NN, K, 2)

100 CONTINUE

300 CONTINUE

CALL GSTOP

STOP

END

SUBROUTINE WR(S₂,M,N)

DIMENSION X(50,50),Y(50,50),DW(2,50)

COMMON X,Y,DW,J₁,I₁,J₂,I₂,J₃,I₃,A₁,A₂,A₃,B₁,B₂,B₃,CN₂,CN₂,NN

CALL FY

I₁ = I₂

J₁ = J₂

J₂ = J₃

I₂ = I₃

A₂ = A₃

B₂ = B₂

A₁ = (J₁ - 1 + X(J₁,I₁))*S₂)*CN₁

B₁ = (I₁ - 1 + Y(J₁,I₁))*(1 - S₂))*CN₂

IF(S₂.EQ.1.0)X(J₁,I₁)= -2.0

IF(S₂.EQ.0.0)Y(J₁,I₁)= -2.0

DW(1,1)=A₁

DW(2,1)=B₁

DW(1,2)=A₂

DW(2,2)=B₂

K=0

200 IF(B₂.EQ.0.)RETURN

IF(I₂.EQ.M)RETURN

IF(A₂.EQ.0.0)RETURN

IF(J₂.EQ.N)RETURN

CALL FY

A₁ = A₂

A₂ = A₃

B₁ = B₂

B₂ = B₃

$$I_1 = I_2$$

$$I_2 = I_3$$

$$J_1 = J_2$$

$$J_2 = J_3$$

$$K = K + 1$$

$$NN = K + 2$$

$$DW(1, NN) = A_2$$

$$DW(2, NN) = B_2$$

GOTO 200

END

SUBROUTINE FY

DIMENSION X(50, 50), Y(50, 50), DW(2, 50)

COMMON X, Y, DW, J₁, I₁, J₂, I₂, J₃, I₃, A₁, A₂, A₃, B₁, B₂, B₃, CN₁, CH₂, NN

IF (I₂ · LE. I₁) GOTO 40

IF (Y(J₂, I₂) - Y(J₂ + 1, I₂)) / 10. ≥ 0.30

10 IF (Y(J₂, I₂) > 0.) GOTO 900

GOTO 920

20 IF (Y(J₂, I₂) < 0.) GOTO 910

IF (X(J₂, I₂) < 0.5) GOTO 900

GOTO 920

30 IF (Y(J₂ + 1, I₂) > 0.) GOTO 920

GOTO 900

40 IF (I₂ · LE. I₁) GOTO 80

IF (X(J₂, I₂) - X(J₂, I₂ + 1)) / 50. ≥ 0.70

50 IF (X(J₂, I₂) > 0.) GOTO 930

GOTO 910

60 IF (X(J₂, I₂) < 0.) GOTO 920

IF (Y(J₂, I₂) < 0.5) GOTO 930

GOTO 910

70 IF (X(J₂, I₂ + 1) > 0.) GOTO 910

GOTO 930

80 IF $A_2 \cdot LE.(J_2 - 1) * CN_1$)GOTO 120

IF $(Y(J_2, J_2 - 1) - Y(J_2 + 1, J_2 - 1)) \geq 0$,100 ,110

90 GOTO 960

100 IF $(Y(J_2, J_2 - 1) \cdot LT.0.0)$)GOTO 950

IF $(X(J_2, J_2) \cdot LE.0.5)$)GOTO 940

GOTO 960

110 GOTO 940

120 IF $(X(J_2 - 1, J_2) - X(J_2 - 1, J_2 + 1)) \geq 0$,130 ,140 ,150

130 GOTO 990

140 IF $(X(J_2 - 1, J_2) \cdot LT.0.)$)GOTO 980

IF $(Y(J_2, J_2) \cdot LE.0.5)$)GOTO 970

GOTO 990

150 GOTO 970

900 CALL F $X(J_2, J_2, 0.)$

RETURN

910 CALL F $X(J_2, J_2 + 1, 1.0)$

RETURN

920 CALL F $X(J_2 + 1, J_2, 0.)$

RETURN

930 CALL F $X(J_2, J_2, 1.0)$

RETURN

940 CALL F $X(J_2, J_2 - 1, 0.0)$

RETURN

950 CALL F $X(J_2, J_2 - 1, 1.0)$

RETURN

960 CALL F $X(J_2 + 1, J_2 - 1, 0.)$

RETURN

970 CALL F $X(J_2 - 1, J_2, 1.0)$

RETURN

```

980  CALL FZ(J2-1, J2, 0.0)
      RETURN
990  CALL FZ(J21, J2+1, 1.0)
      RETURN
      END
      SUBROUTINE FZ(J0, J0, S1)
      DIMENSION X(50, 50), Y(50, 50), DW(2, 50)
      COMMON X, Y, DW, J1, J1, J2, J2, J3, J3, A1, A2, A3, B1, B2, B3, CN1, CN2, NN
      I3 = J0
      I3 = I0
      A3 = (J3 - 1 + S1 * X(J3, J3)) * CN1
      B3 = (I3 - 1 + (1. - S1) * Y(J3, J3)) * CN2
      IF(S1 .EQ. 0.) Y(J3, J3) = -2.0
      IF(S1 .EQ. 1.) X(J3, J3) = -2.0
      RETURN
      END
      SUBROUTINE CONTOU(P, N, K, KEY)
      DIMENSION R(2, 50), Q(2, 50), A(2)
      F1(T) = 1.0 - 2.0 * T + T * T
      F2(T) = 2.0 * T - 2.0 * T * T
      F3(T) = T * T
      DO 10 I = 1, N
      I1 = (I + 1) - I / N * N
      DO 10 J = 1, 2
10   Q(J, 1) = 0.5 * (R(J, 1) + R(J, I1))
      IF(KEY .EQ. 2) GOTO 50
      CALL PLOT(R(1, 1), R(2, 1), 3)
      CALL PLOT(Q(1, 1), Q(2, 1), 2)
      N1 = N - 1
      DO 40 I = 2, N1

```

```
DO 30 M = 1 ,K
T = 1.0 * M/K
DO 20J = 1 2
20  A(J)= F1( T)* Q(J ,I - 1 )+ F2( T)* R(J ,I)+ F3( T)* Q(J ,I)
30  CALL PLOT( A( 1 ) ,A( 2 ) 2 )
40  CONTINUE
    CALL PLOT( R( 1 ,N ) ,R( 2 ,N ) 2 )
    RETURN
50  CALL PLOT( Q( 1 , 1 ) ,Q( 2 , 1 ) , 3 )
DO 80 I = 1 ,N
I1 =( I + 1 )- I/N * N
DO 70 M = 1 ,K
T = 1.0 * M/K
DO 60J = 1 2
60  A(J)= F1( T) ,* Q(J ,I )+ F2( T)* R(J ,I1 )+ F3( T)* Q(J ,I1 )
70  CALL PLOT( A( 1 ) ,A( 2 ) 2 )
80  CONTINUE
    RETURN
```

第四章 微机绘图软件 Auto CAD 简介

Auto CAD 是美国 Auto Desk 公司推出的微机通用绘图软件包,是一个功能强、应用广泛的绘图软件。该系统具有较强的图形编辑功能,内部嵌有 Auto LISP 语言,增加了系统的开放性。系统能够与高级语言以及 DBASE 数据库等进行数据通讯。Auto CAD 软件的另外一个显著优点是它能够支持 140 多种外部输入输出设备。自 1982 年首次推出以来,Auto CAD 已经过几次版本升级,国内已开发出汉化版本。

在目前的矿山 CAD 软件中,有很大一部分是以 Auto CAD 软件为基础开发的。本节简单介绍 Auto CAD 软件的基本情况。

第一节 软件的基本操作

一、启动及主菜单

软件的安装很简单,只需将软件拷贝在硬盘上的某个子目录即可。

启动时先进入软件所在子目录,然后键入 ACAD,数秒钟后屏幕上将显示 Auto CAD 主菜单。

0.Exit Auto CAD(退出 Auto CAD)

(1)Begin a New drawing(绘制新图)

(2)Edit an Existing drawing(编辑现有图)

- (3) Plot a drawing(绘图机绘图)
- (4) Printer Plot a drawing(打印机绘图)
- (5) Configure Auto CAD(配置 AutoCAD)
- (6) File Utilities(配置实用程序)
- (7) Compile Shape/fontdescriptionfile(编译形/字型描述文件)
- (8) Convert old drawingfile(转换原有的图形文件)

Enter selection(输入选择项)

用户输入选项号,程序则执行相应的功能。

如果用户是第一次使用 Auto CAD,一般情况下首先应该选择功能 5 即配置 Auto CAD。所谓配置 Auto CAD,是根据用户计算机系统中现有的显示器、数字化仪、绘图仪以及打印机等设备的类型,在菜单提示下,选择相应的设备驱动程序,使软件能在用户的计算机硬件环境里正常运行。

要绘制一张新图时选择功能 1,这时系统询问将要建立的新图的名字,用户输入的名字便成为该图形存盘时的文件名。输入文件名后,系统进入绘图状态,等待用户输入绘图命令,人机交互绘制图形。绘图状态等待绘图命令的提示符为:“Command”。

要编辑修改磁盘上已有的图形,选择功能 2,系统询问图名,然后将该图调入系统并显示在屏幕上,进入绘图状态,等待编辑。

二、基本操作

(1) 功能键。IBMPC 功能键对应的图形编辑功能如下:Ins 显示矩形光标,Home 显示十字光标,End 光标停止,PgVp 光标快移,PgPn 光标慢移,箭头键 \uparrow 、 \downarrow 、 \leftarrow 、 \rightarrow 光标上移、下移、左移、右移,F₆ 坐标触发控制键,F₇ 网格触发控制、F₈ 正交触发控制,Ctrl E 等轴平面触发控制,Ctrl Q 打印机响应。

(2) 命令输入。当屏幕出现提示符 Command 时,可用键盘键入绘图命令,按回车键后命令生效,也可用光标选择屏幕菜单命令。此外也可通过数字化仪菜单传送命令。

(3) 数据输入。用户可通过键盘键入点的绝对坐标或相对坐标以及角度值等,也可移动光标对点进行定位,或者由图形输入设备输入点坐标。

第二节 基本图元及其绘图命令

基本图元有直线、点、圆、圆弧、轨迹、实心区以及文本等。下面说明其绘图命令。

一、LINE

LINE 命令绘制直线及折线,其格式如下:

Command :LINE↙

From Point 起点 ↙

To Point 到点 ↙

To Point 回车

输入 LINE 命令后,接着询问直线的两个端点,回答后就绘制一条直线。对于到点的询问是连续不断的,如果连续用坐标值回答就连续绘制折线,如果用 C 回答,系统将自动把最后一个到点和起点连接起来形成封闭多边形,如果用 U 回答,就从图上取消最后画的一条线段。

如果在“From Point”后直接回车响应,则新线的起点是此前最后画完的直线或圆弧的终点。如果是圆弧的终点,此后的直线即是圆弧的切线,回答“To Point”时只需输入线段的长度。

二、POINT

该命令画一个点,格式为

Command :POINT ↘

Point 点坐标

三、CIRCLE 该命令画一个圆,其格式如下

Command :CIRCLE

3P/2P/ Center :_____

若选用 3P 回答,表示过三点作圆,CIRCLE 命令接着提示输入三点坐标;若用 2P 回答,表示以直径上两端点作圆,该命令接着提示输入两点坐标,缺省方式是输入圆心坐

标 ,然后通过选择绘出半径或直径作圆。

四、ARC

ARC 是画弧命令。该命令可提供八种不同的画弧方法。命令的默认方法是指定弧上的三点画弧 ,其它确定弧的方法是先回答一个字母 ,接着绘出要输入的信息。ARC 命令选择项字母的含意如下 :A 为夹角 ;E 为终点 ;C 为圆心 ;L 为弦长 ;D 为起始方向 ;R 为半径。

ARC 提供的八种画弧方法是 给定弧上三点 给定始点、圆心和终点 给定始点、圆心和夹角 给定始点、圆心和弧长 给定始点、终点和半径 给定始点、终点和起始方向 给定始点、终点和夹角 ;与前面的直线或弧连接。

例如 ,要通过给定始点、圆心和终点画弧 ,命令格式如下 :

Command :ARC

Center/ Start Point :点

Center/End/ Second Point :C

Center :点

Angle/Lenth of chord/ End Point :点

五、TRACE

TRACE 命令绘制具有一定宽度的线条即轨迹 ,它与 LINE 命令的区别仅在于在输入始点之前要先回答宽度 ,其格式如下 :

Command :TRACE

Width :宽度

To Point :

To Point :回车

六、TEXT

TEXT 命令将文字写到图形上 ,格式如下 :

Command :TEXT

Start Point(or ACRS)

可用选择项之一来回答。选项的含义如下 :Start Point 为将文本向左对齐到指定点 ; A 为指定两点 ,使文本处于这两点之间 ;C 为指定一点 ,文本以该点为中心 ;R 为指定一点 ,文本向右对齐该点 ;S 为请求一种新字型。

在绘制文本之前 ,命令提示要求给出文本的高度、旋转角和文本字符串等。

第三节 图形编辑命令

常用的图形编辑命令有图形移动、缩放、复制、修改、删除、旋转、重画等。

一、ZOOM

ZOOM 命令放大或缩小屏幕显示的图形 ,格式如下 :

Command ZOOM

Magnification or type(ACELPW):

这时可以输入一个缩放系数或 ACELPW 中的任一选择项 ,选择项的含义如下 :A 为显示整幅图形的全部内容 ;C 为按指定的中心点和高度显示图形 ;E 为显示当前范围内的全部图象 ;L 为指定显示区域的左下角和高度 ;P 为恢复前一幅图形 ;W 为指定矩形窗口 ,窗口内的图形全部显示在屏幕上。

二、PAN

该命令在不改变比例的条件下 ,上下左右移动图形。命令要求给出移动的位移量 ,可以输入一对位移量 ,也可以输入两个点的坐标 ,这时 Auto CAD 计算从第一点到第二点的位移量。命令格式如下 :

Command :PAN

Displcement : 位移量或点坐标

Second Point : 点坐标或回车

三、ERASE

该命令删除指定的图形元素。格式如下 :

Command :ERASE

Select objects or window or last : _____

Select objects 是选择目录 ,用十字光标指定要删去的图元 ,然后回车即可 ;Window 是用窗口选择删除对象 ,键入 W ,然后指定窗口的两个角点 ,窗口内的图元被删除 ;Last 表示删除最后画的图元。

四、OOPS

当用 ERASE 删除一个图元后 ,OOPS 命令可将其恢复。

五、MOVE

该命令将图元移到新的位置。格式如下 :

Command :MOVE

Select objects or window or last : 选择图元

Base Point or displacement : 第一点或 x、y 位移量

Second Point of displacement : 第二点或回车

六、Copy

该命令把一个图形复制到另一个指定的位置。此命令与 MOVE 类似 ,不同之处在于原图保持不变。

七、BREAK

BREAK 命令可删去直线、圆弧等图元的一部分。该命令首先要求指定处理对象 ,然后再指定要断开的两个端点。命令格式如下 :

Command :BREAK

Select objects or widow or last : 选择目标

Enter First Point : 断点 1

Enter Second Point : 断点 2

八、REDRAW

该命令重画屏幕图形。重画可以清除遗留在屏幕上的书改痕迹。

第四节 实用命令

一、END

该命令将当前图存入磁盘文件并返主菜单。

二、QUIT

该命令返回主菜单而不修改原图文件 ,即不把当前图存入磁盘。

三、ENDSV

该命令象 END 命令一样退出绘图编辑状态返回主菜单 ,同时存储矢量文件 ,矢量文件包含的信息是显示在屏幕上的图形。

四、SAVE

该命令将当前图存入磁盘文件 ,但是保持绘图编辑状态。

五、LIMITS

该命令指明当前图的界限 ,并可控制这些界限的检验功能。

六、UNITS

该命令选择坐标格式和角度格式。

七、FILES

该命令在绘图状态下可以列出磁盘文件名、删除 ,重新命名和拷贝某些文件。

八、STATOS

该命令列出与当前图有关的状态信息 ,例如绘图极限和显示范围等。

九、AREA

计算一个圆或封闭多边形的面积或周长。

十、DIM

DIM 为标注尺寸命令。

第五节 图 层

一、基本概念

一幅图可以分成若干层,每一层称为一个图层。图层可以想象为透明的覆盖层。图上的图元可以放在不同的层上,因此可将图元分门别类按层分开绘制,一层图上可以含有与图的某一特别方面有关的图素,以便对这些图元进行分析和控制。一幅图可以有的层数是不受限制的。一幅图的各个图层具有相同的绘图边界、坐标系以及水例系数等,各层之间完全对齐。每一个图层都有一个图层名、线型和颜色号,不同的层可以有不同的颜色和线型。

每种线型都有一个名字和定义。用户可以用 Auto CAD 线型库中的线型,也可以自己定义。在某层上的图形将使用该图层的线型绘制。

颜色号表示不同的颜色。颜色号是从 1 到 255 的整数,开始 7 个号对应标准色名:1 为红,2 为黄,3 为绿,4 为青,5 为蓝,6 为洋红,7 为白。

一个图可以是可见的,也可以是不可见的。只有可见的图层才可以显示,输出和绘图。

当开始画新图时,Auto CAD 自动生成 0 号图层,颜色号为 7,线型为实线。

二、层命令

层命令 LAYER,用来建立图层,选择当前层以及为指定的层设置颜色和线型等,格式如下:

Command :LAYER

? /Set/New/ON/OFF/Color/Ltype/Freeze/Thaw :_____

各选择项的意义如下 :? 为列出当前定义的图层清单 ,包括层名、颜色、线型及可见性等 ;
S 为将新生成的层置为当前层 ,接着需给出当前层名 ;N 为生成一个新图层 ;ON 为打开
图层 ;OFF 为关闭一个图层 ,关闭了的图层不能显示和绘图 ;C 为给新生成的图层指定颜
色号 ;L 为选择线型 ;F 为层冻结 ,被冻结的层不能显示和绘图 ;T 为层解冻。

例如 ,屏幕上有一矩形 ,当前层为 0 层 ,线型实线 ,颜色为白色 ,现在要在矩型内用红
颜色虚线画一个小矩形 1234 ,并将小矩形画在 1 层上。操作如下 ,

Command :LAYER :

? /Set/New/ON/OFF/Color/Ltype/Freeze/Thaw :? _____(查阅当前图各层
参数)

Layer name(s) for Listing * :_____

Layer name State Color Linetype

0 on (white) CONTINUOUS

Current Layer 0

? /Set/New/ON/OFF/Color/Ltype/Freeze/Thaw :N

New Layer name(s) :1

? /Set/New/ON/OFF/Color/Ltype/Freeze/Thaw :S

New Current Layer o :1

? /Set/New/ON/OFF/Color/Ltype/Freeze/Thaw :C

Color :RED (或 1)

Layer name(s) for color (red) 1 :1

? /Set/New/ON/OFF/Color/Ltype/Freeze/Thaw :L

Linetype (or ?) CONTINUOUS :DASHED (虚线)

Layer name(s) for Linetype DASHED 1 :1

Command :UNE

From point : 点 1 坐标

To point : 点 2 坐标

To point : 点 3 坐标

To point : 点 4 坐标

To point : (构成封闭矩形)

第六节 块

一、基本概念

块是一组图元。Auto CAD 把块当作单一的对象进行处理。块可以由绘制在几个层上的若干图元组成,并保留各层的信息。每个块都有一个块名。块一经定义,就可以用编辑和查询命令对其进行处理。可以把块插入到某个图形中,此时块中的每个图元就在它原来的层上绘出,只有绘制在 0 层上的图元在插入时被绘制在当前层上。插入图形中的块可以具有不同的比例系数和旋转角。

二、块命令

(1)BLOCK。BLOCK 把当前图里的一组图元定义为一个块,格式如下:

Command :BLOCK

BLOCKname(or ?): 块名或?

Insertion base point : 插入基点

Select objects or window or last : 指定块的成份

在回答块名时可以输入新块名或“?”。如果输入的块名已经存在,则要回答是否重新定义。如果输入“?”则列出当前块名清单。

块的插入基点一般选在块的中心或左下角。以后当把块插入到图形中时,插入基点将与插入点对齐,块还可绕此点旋转。

指定块的成分时,其选择方法与图形编辑时选择图元的方法相同。

(2)INSERT。该命令把以前定义的插入到图中,格式如下:

Command :INSERT

Block name(or ?): 块名

Insert point : 插入点

X Scale factor 1 /corner/XYZ : X 比例系数

Y Scale factor(default = X): Y 比例系数

Rotation angle 0 : 旋转角

(3)WBLOCK。此命令将已定义的块写入一个磁盘文件,格式如下:

Command :WBLOCK

Filename : 存盘的文件名

Blockname : 块名

(4)BASE。此命令规定基准点,供插入块时作参考点,格式如下:

Command :BASE

Base Point 0.0000 0.0000 0 0.0000 点坐标

第七节 图形交换文件

一、基本概念

Auto CAD 本身可以作为一个完整的绘图编辑软件使用。但在许多应用中,需要其它程序检验和分析由 Auto CAD 产生的图,有时需要用 Auto CAD 显示或编辑其它程序产生的图形。图形交换文件就是 Auto CAD 与高级语言之间连接的媒介。Auto CAD 的内部图形数据库可以转换为图形交换文件格式,供高级语言程序读取,而由高级语言程序产生的图形交换文件也可由 Auto CAD 软件读取并转换为其内部图形数据库格式,进行图形显示或编辑。

二、图形交换文件格式

图形交换文件的后缀为“DXF”,所以也称为 DXF 文件。

(一)DXF 文件的一般结构

(1)HEADER(标题)段:给出图形的各种当前设置和参数。

(2)TABLES(表)段:包括线型表、层表、字体表、视图表

(3)BLOCK(块)段:对组成图形中的各个块的实体即图元进行描述。

(4)ENTITIES(实体)段:对图形中的图有实体进行说明。

(5)END OF FILE:文件结束标志 EOF。

一个 DXF 文件,不一定包含上面的全部内容。如果不需要重新设置标题变量,则整个标题段均可省略;表段中的四个表以至整个表段都可根据需要省略。最简单的 DXF

文件只包括实体段和文件结尾标志。

(二)组代码

DXF 文件的每一段、表等的开头与结束处,均用文件分界符来表示,如段头用 SECTION,表用 TABLE、文件结束用 EOF 表示。而 DXF 文件的每个段又由两行为一组的若干单元组成。第一行是组代码,它是一个非负整数,组的第二行是组值,采用的格式取决于由组代码规定的组的类型,具体说明如下:0~9 为字符型;10~59 为浮点型;60~79 为整数型。

组代码既可指明如上述的组值的类型,又可指出组的一般应用,下面进一步说明:0 为标识一个实体、表、文件分隔符的起始;1 为实体的主文字说明;2 为名称、块名、属性特征值;3~5 为其它文字或名称值;6 为线型名;7 为文本字样名;8 为图层名;9 为变量名标识符(仅用于 HEADER 段);10 为起始 x 坐标(线或文本的起点、圆心等);10~18 为其它 x 坐标;20 为初始 y 坐标;21~28 为其它 y 坐标;30~38 为 z 坐标;40~48 为浮点数值(字高、比例系数等);49 为重复值;50~58 为角度;62 为颜色;66 为图元跟随标记;70~78 为整数值,象重复计数、标记位或方式等。

(三)HEADER 段

HEADER 段描述了一组表示图形各种参数设置及当前状态的变量。每个变量由组代码 9 来规定,变量名用“\$”打头,其后跟若干组值。下面具体说明变量名,紧跟其后的组及其含义。

\$ACADVER 1. AutoCAD 版本号

\$ATTMODE 70.属性的可见性 0 为不可见,1 为标准的,2 为全部的

\$AUNITS 70 角度用的 UNITS 格式

\$AUPREC 70 角度用的 UNITS 精度

\$AXISMODE 70 如非 0,显示坐标轴

\$AXISUNIT 10.20 x 轴和 y 轴的刻度间隔

\$DIMTOL 70 如果非 0,生成尺寸线公差

\$DIMTP 40 正公差

\$DIMTSZ 40 尺寸刻度线大小,0 为无刻度线

\$DIMTXT 40 尺寸文本高度

\$DRAGMODE 70 如非 0,进入拖动方式

\$ELEVATION 40 用 ELEV 命令设置正视图

\$EXTMAX 10.20 图延伸到右上角

\$EXTMIN 10.20 图延伸到左上角

\$FILLETRAD 40 倒角半径

\$FILLMODE 70 如果非 0 ,进入 FILL 方式

\$GRIDMODE 72 如果非 0 ,进入 GRID 方式

\$GRIDUNIT 10.20 栅格的 x 和 y 间距

\$THICKNESS 40 用 ELEV 命令设置当前厚度

\$INSBASE 10.20 由 BASE 命令设置的插入点

\$LIMCHECK 70 如果非 0 ,允许边界检查

\$LTSCALE 40 全局的线性比例

\$LIMMAX 10.20 绘图范围的右上角

\$LIMMIN 10.20 绘图范围的左下角

\$LUNITS 70 坐标和距离用的 UNITS 格式

\$CLAYER 8 当前层名字

\$BLIPMODE 70 如果非 0 ,显示标记

\$CHAMFERA 40 第一个倒棱距离

\$CHAMFERB 40 第二个倒棱距离

\$DIMASZ 40 标准尺寸箭头大小

\$DIMCEN 40 圆心标记

\$DIMDLI 40 尺寸线增量

\$DIMEXE 40 延伸线增量

\$DIMEXD 40 延伸线偏移量

\$DIMHIM 70 如果非 0 ,生成根据偏差

\$DIMSCALE 40 总的尺寸比例系数

\$DIMSEI 70 如果非 0 ,取消第一延伸线

\$DIMSEZ 70 如果非 0 ,取消第二延伸线

\$DIMITAD 70 如果非 0 ,文本说明在尺寸线上

\$DIMITIH 70 如果非 0 ,文本说明在尺寸线线内

\$DIMITM 40 负公差

\$DIMITOH 70 如果非 0 ,文本说明在尺寸线之外

\$LUPREC 70 坐标和距离用的 UNITS 精度

\$MENU 1 菜单文件的名字

\$ ORTHOMODE 70 如果非 0 打开 ORTHO 方式
\$ OSMODE 70 物体捕捉方式
\$ QTEXTMODE 70 如果非 0 打开快显文本方式
\$ SKETCHINC 40 随手绘图记录增量
\$ SNAPANG 50 捕捉栅格旋转角度
\$ SNAPBASE 10.20 捕捉栅格基点
\$ SNAPISOPAIR 70 选择三维平面 0 = 左 ,1 = 顶 2 = 右
\$ SNAPMODE 70 如果非 0 打开捕捉方式
\$ SNAPSTYLE 70 捕捉方式 0 = 标准方式 ,1 = 等距方式
\$ SNAPUNIT 10.20 捕捉栅格 x 和 y 的间距
\$ TEXTSIZE 40 默认文本高度
\$ TEXTSTYLE 70 当前文本字样名字
\$ TRACEWID 40 默认线条宽度
\$ VIEWCTR 10.20 屏幕上当前视图中心
\$ VIEWDIR 10.20.30 用 VPOINT 设置测视点
\$ VIEWSIZE 40 屏幕当前视点的高度

(四) TABLES 段

TABLES 段包含 LTYPE、LAYER、STYLE 及 VIEW 四个表,每个表又包含有可变数目的表项。在 TABLES 段中,每个表都是由带有 TABLE 标记的一个 0 组来引入的,接着用 2 组来命名表,紧接着是一个 70 组,规定允许跟随的表项的最大数目,其后就是某一个表的表项。

每个表项都包含一个标识表项类型的 0 组(其后内容和表名相同),一个给出表项名称的 2 组和一个规定与表项有关的标志的 70 组以及若干给出表项值的附加组,每个表结束都用一个带有 'ENDTAB' 值的 0 组来指出。

下面分别给出各类表的附加组内容

(1) LTYPE。3[对线型的描述],72[对准方式代码],73[点划线的段数],40[总的图案长度],49[短划线长度 1],49[短划线长度 2],.....

(2) LAYER。62[颜色号,负值表示层关闭],6[线型名字]

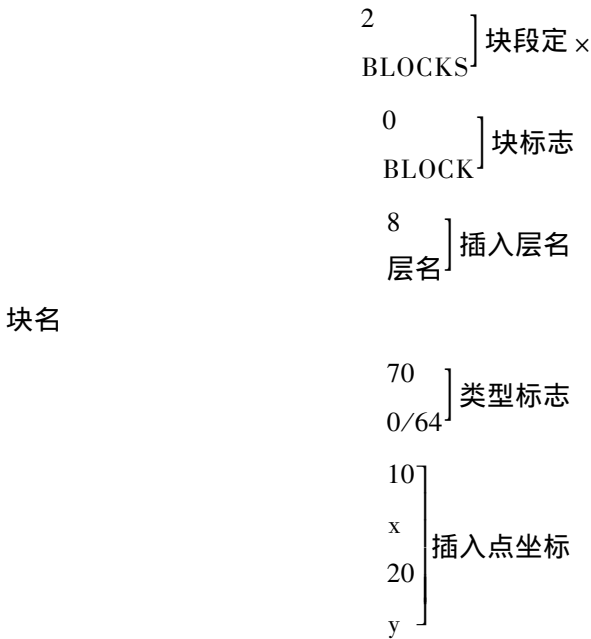
(3) STYLE。40[固定的文本高度,如果不固定,则为 0],50[倾斜角度],71[文本生成标志],42[最近使用的高度],3[文本字体或形文件名字]。

(4) VIEW。40 和 41[视图高度和宽度],10 和 20[视图中心点],11、21 和 31[从原点

的视图方向]

(五)BLOCKS 段

BLOCKS 包括所有块的定义以及在绘图中使用的块的图元。该段中的图元格式与下面的 ENTITIES 段中的描述格式相同 ,因此在 BLOCKS 段中说明过的实体(图元) ,在 ENTITIES 段中不再说明 ,而只需用 0 组 INSERT 简要说明层名、块名、插入点坐标即可。BLOCKS 段结构如下 :



块中画图图元

(六)ENTITIES 段

ENTITIES 段对图形中的实体进行描述。如前述 ,实体在 BLOCKS 与在 ENTITIES 中的格式是一样的。定义一个图元 ,有些组一定要有 ,而某些组是任选的 ,并且仅在它们与缺省值不同时才出现。每个实体都从给出实体名字的 0 组开始 ,紧跟其后的 8 组给出保存实体的图层名 ,然后根据作图需要选择实体。下面给出各种实体的名称、意义及其相应的组代码。

LINE 线段 ,10 和 20〔 始点 〕 ,11 和 21〔 终点 〕

POINT 点 ,10 和 20〔 点坐标 〕

CIRCLE 圆 ,10 和 20〔 圆心 〕 ,40〔 半径 〕

ARC 弧 ,10 和 20〔圆心〕 40〔半径〕 50〔起始角〕 51〔终止角〕。

TRACE 轨迹 ,10 和 20 ,11 和 21、12 和 22、13 和 23〔定义线条的四个角点的坐标〕

SOLID 域 ,同上 ,若该域为三角形 ,由 12 和 22 与 13 和 23 组规定的坐标将相等。

REPEAT 表示重复实体 ,无组代码。

ENDREP 意义同上 ,70〔重复列数〕71〔行数〕 40〔列距〕 41〔行距〕。

TEXT 文本 ,10 和 20〔插入点〕 40〔文本高〕 ,1〔文本值〕 50〔旋转角〕 41〔x 比例系数〕 51〔倾斜角〕 ,7〔文本字样〕 ,71〔生成标志〕 ,72〔对准类型〕 ,11 和 21〔对准点〕。

SHAPE 形 ,10 和 20〔形插入点〕 40〔大小〕 2〔形名〕 50〔旋转角〕 41〔x 比例因子〕 , 51〔倾斜角〕。

BLOCK 块 ,2〔块名〕 ,70〔块类型〕 ,10 和 20〔块基点〕。

ENDBLK 块结束 ,无组。

INSERT 插入块 ,66〔属性〕 2〔块名〕 ,10 和 20〔插入点〕 41〔x 比例系数〕 50〔旋转角〕。

ATTDET 属性 ,10 和 20〔文本起点〕 40〔文本高〕 ,1〔默认值〕 ,3〔提示字符串〕 ,2〔属性标志串字符串〕 ,70〔属性标志〕 ,73〔段长〕 50〔文本旋转角〕 ,7〔文本字体名〕 ,71〔文本生成标志〕 ,72〔文本对齐方式〕 ,11 和 21〔对齐点坐标〕。

ATTRIB 同上 ,1〔文字内容〕 ,无 3 组代码。

POLYLINE 连续线段 ,70〔标记〕 40〔默认开始宽度〕 41〔默认结束宽度〕。

VERTEX 连续线段顶点 ,10 和 20〔位置〕 40〔开始宽度选择〕 41〔结束宽度选择〕 , 42〔凸出〕 ,70〔顶点标记〕 50〔曲线拟合切线方向选项〕。

SEQEND 连续线段顶点或插入的属性实体的结束标记 ,无组。

第八节 DXF 文件命令

一、DXFOUT

DXFOUT 将当前图形信息转换为一个 DXF 文件 ,格式如下：

Command : DXFOUT

File name default

此时可输入文件名,也可直接回车,缺省名与当前图形文件名相同,但后缀为“DXF”。

二、DXFIN

此命令读入一个 DXF 文件并将其转换为一个 Auto CAD 图形文件(即后缀为“DWG”的文件)格式如下:

Command :DXFIN。

File name : DXF 文件名

第九节 编写 DXF 文件的接口程序

所谓接口程序,是由高级语言编写的能够生成和读取 DXF 格式数据文件的程序。

一、生成 DXF 文件

要编写一个能够构造一个完整的 DXF 文件的程序是比较困难的。幸运的是 Auto CAD 允许省略 DXF 文件中的许多项。原则如下:如果不需要设置任何标题变量,整个 HEADER 段就可省去;根据应用情况可以省去 TABLES 段中的任何一个表以至整个 TABLES 段;如果图中没有用块定义,则 BLOCKS 段也可略去。

下面给出一个用 BASIC 语言编写的生成一个简单的 DXF 文件的程序,这个 DXF 文件绘制一个任意边数的正多边形。

```
1010 REM polygon generator
1030 INPUT " DXF file name : "; A$
1040 OPEN A$ + ".DXF" FOR OUTPUT AS #1
1050 PRINT #1 0
1060 PRINT #1 " SECTION "
1070 PRINT #1 2
1080 PRINT #1 " ENTITIES "
1090 PI = ATN(1) * 4
1100 INPUT " Number of sides for polygon ; "; S%
```

```
1110 INPUT " starting point ( x , y ) : " ; x , y
1120 INPUT " side length : " ; D
1130 AL = ( 2 * PI ) / s %
1140 A = AL / 2
1150 FOR I % = 1 To s %
1160 PRINT # 1 , 0
1170 PRINT # 1 " , LINE "
1180 PRINT # 1 , 8
1190 PRINT # 1 " , 0 "
1200 PRINT # 1 , 10
1210 PRINT # 1 , x
1220 PRINT # 1 , 20
1230 PRINT # 1 , y
1240 NX = D * cos ( A ) + x
1250 NY = D * sin ( A ) + y
1260 PRINT # 1 , 11
1270 PRINT # 1 , NX
1280 PRINT # 1 , 21
1290 PRINT # 1 , NY
1300 X = NX
1310 Y = NY
1320 A = A + AL
1330 NEXT I %
1340 PRINT # 1 , 0
1350 PRINT # 1 " , ENDSEC "
1360 PRINT # 1 , 0
1370 PRINT # 1 " , EOF "
1380 CIOSE
1390 STOP
1400 END
```

二、读取 DXF 文件

编写读取接口程序比较容易,因为用户通常只对 DXF 文件中的有关信息感兴趣,而无关部分可以略去。下面给出一个读取前面构造的 DXF 文件的 BASIC 程序:

```

10 REM extract lines from dxf file
20 DEFINT I
30 DIM X(100),Y(100),X2(100),Y2(100)
80 INPUT "enter file name : "; c1$
90 c1$ = c1$ + ".dxf"
100 OPEN "I" c1$
110 INPUT #1 c$
120 IF c$ = "ENTITIES" GOTO 110
130 INPUT #1 c$
140 IF c$ = "ENDSEC" GOTO 710
150 IF c$ = "LINE" GOTO 130
160 I = I + 1
170 INPUT #1 c
180 INPUT #1 c
190 INPUT #1 ,N%
200 INPUT #1 ,X11
210 X(I) = X11
220 INPUT #1 ,N%
230 INPUT #1 ,Y11
240 Y(I) = Y11
250 INPUT #1 ,N%
260 INPUT #1 ,X22
270 X2(I) = X22
280 INPUT #1 ,N%
290 INPUT #1 ,Y22
300 Y2(I) = Y22
310 PRINT "X(" ; I ; ") = " ; X(I) ; " Y(" ; I ; ") = " ; Y(I)

```

```
320 PRINT " X(" ; I ; ") = " ; X(I) ; " Y(" ; I ") = " ; Y(I)
330 GOTO 130
710 CLOSE " 1 "
720 STOP
730 END
```

第十节 使用 Auto LISP

Auto LISP 是安装在 Auto CAD 内部的解释性程序语言,用户可以在绘图编辑状态下运行 Auto LISP 程序。Auto LISP 除了具有数字计算、定义函数等功能外,程序还可以访问图形的图元、执行 Auto CAD 的绘图命令,同时可对磁盘文件进行读写。因此,Auto LISP 为用户提供了较强的编程能力,以弥补 Auto CAD 功能的不足。用户可通过 Auto LISP 建立用户菜单和应用程序,控制交互图形设计过程。

由于篇幅所限,这里不准备介绍 Auto LISP 的详细情况。下面简单说明使用 Auto LISP 的一般过程。

一、准备工作

在首次使用 Auto LISP 之前,应进行下述准备工作。

(1) 启动 Auto CAD,键入 ACAD 后,屏幕出现软件的主菜单:

- 0.Exit Auto CAD
- 1.Begin a NEW drawing
- 2.Eixt an EXISTING drawing
- 3.Plot a drawing
- 4.Printer plot a drawing
- 5.Lonfigure Auto CAD
- 6.File Vtilities
- 7.Compils shape/font description file
- 8.Conure old drawing flie

Enter Selection :_____

(2) 选择功能 5,键入 5 后,屏幕出现:

- 0.Exit to Main Menu
- 1.Show current configuration
- 2.Ailow defailed configuration
- 3.Configure chisplay
- 4.Configure Chigitizer
- 5.Configure Plottev
- 6.Configure printer plottor
- 7.Configure system console
- 8.Configure operating parameters

Enter Slectlon : _____

(3)选择功能 8 输入 8 后 ,屏幕出现 :

- 0.Exit to configuration menu
- 1.Alarm on error
- 2.Initial dtawing setup
- 3.Default plot file name
- 4.Plot spooler directory
- 5.Placement of temporary files
- 6.Network node name
- 7.Auto lisp feture

Enter selection : _____

选择功能 7 输入 7 后 ,屏幕上出现 :

Do you want Auto Lisp enabled? Y :

直接回车或回答 Y 后 ,Auto LISP 程序系统就被调入 ,今后就可以使用 Auto LISP 了。

二、Auto LISP 程序的建立和执行

(1)程序的建立。通常的作法是用文件编辑程序建立 Auto LISP 的用户程序 ,程序后缀为“ LSP ”。设已建好用户程序 Demo.LSP。

(2)程序的执行。在 Auto CAD 绘制图形状态下 ,装入应用程序 ,例如 :

Command :Load"demo"

就将上面建立的程序调入 Auto CAD。

此后 ,只需在 Command 提示符下键入程序名 ,程序即开始运行。

第五章 矿山 CAD 系统的开发原理及软件实例

第一节 矿山 CAD 系统的开发原理

矿山 CAD 系统是以计算机图形信息处理技术为主要手段,进行矿山工程设计工作的计算机软硬件系统。由于矿山 CAD 技术应用的范围涉及到矿山工程的各方面,各个矿山的的具体情况又互不相同,因此这里无法给出矿山 CAD 系统的统一模式。下面对矿山 CAD 系统的一般开发原理,作简单地概括。

一、系统结构

矿山 CAD 系统,从系统功能来看,一般可分为以下几个部分:数据管理、应用模型、图形和数据的输入输出、交互图形设计以及计算优化模型等。上述几个部分应构成一个有机的整体,才能使软件系统使用方便,运行稳定可靠。

二、数据管理

矿山 CAD 系统需要大量处理的数据通常有地质勘探数据、生产测量数据等。通过这些数据,系统获取地质生产信息、绘制地质采矿生产图件。这些数据的特点一是量大,二是随着地质勘探和采矿生产的不断进行,需要补充和修改。为了提高设计系统的效

率,应建立有效的数据库管理系统来管理这些数据,一方面能及时更新数据库内容,另一方面能很及时地响应设计系统的请求,方便地提供有关数据。

除了字符和数值型的数据外,往往还有大量的人工绘制的地质采矿图件也是矿山 CAD 系统需要使用的原始信息。根据实际情况对这些图形信息也应有效地管理起来,提高 CAD 系统的效率。

三、应用模型

应用模型是根据具体应用目的建立的模型,应用模型包含着需要解决的问题的描述信息。这些信息通常包括地质模型、矿体几何信息、开采条件以及开采现场的几何描述、设计规则及针对某些问题的知识结构等。从某种意义上讲,应用模型是 CAD 系统的加工对象。因此,应该设计合理的数据结构来表示模型信息。应用模型的数据结构,不仅要能合理准确地表示客观事物,同时要便于和图形设计系统和数据库管理系统进行接口。

四、交互图形设计

交互图形设计是矿山 CAD 系统的核心内容,是在图形处理软件和数据库系统的支持下,以人机对话的计算机图形处理方式,对应用模型进行加工处理的过程。通过交互图形设计,可以绘制各种地测图件,设计各种采矿工程、模拟露天矿采剥过程、编制生产计划等。设计矿山 CAD 系统的图形交互设计功能,应考虑以下问题:

(1)图形支撑软件的选择。对于矿山 CAD 系统,通常以 Auto CAD 软件作为图形支撑软件。其优点是图形处理功能强,支持多种图形输入输出设备。并具有高级语言二次开发接口。

(2)屏幕的划分。交互设计都是以屏幕为设计平台的,因此对屏幕要充分利用,根据显示的图形内容和菜单作合理的划分,对显示格式要认真设计。

(3)菜单设计。菜单是交互设计系统的用户接口,用户通过选择菜单的选项,启动系统执行相应的图形设计功能。因此,应将用户可选择和控制的图形处理功能、对象、数据或其它有关项目设计成菜单,通过选择菜单、进行交互设计。菜单的设计方式一般有以下几种:

- ①固定式,一般适用于静态菜单,它自始至终显示在屏幕的某一固定区域;
- ②翻页式,其菜单按层次分页,进入一层菜单就象翻过一页;
- ③拉帘式,其用户可象拉帘子一样拉出一个菜单;

- ④增长式 这种菜单的显示就象是慢慢长出来的；
- ⑤弹出式 这是目前最常用的菜单出现方式 ,菜单好象是弹出来的。

(4)输入输出功能。矿山 CAD 系统通常要输入大量的图件和数据 ,因此要制定统一的输入接口。对于图件的输入 ,可设计屏幕菜单和数字化仪菜单辅助进行。

由矿山 CAD 系统输出的图表的格式要符合矿山的规范。

第二节 矿山 CAD 软件实例

一、某地下金矿生产管理和设计 CAD 软件

下面首先介绍我国某矿山设计研究院为某地下金矿开发的 CAD 软件。

(一)建立矿山分层、分块地质图库

利用 Auto CAD 绘图软件图层特性和随意选择输出图纸比例及选择笔型、线型等特点 ,把地质图件上具有不同特征特性的图形元素分门别类 ,分别存入不同的层中 ,并以不同的颜色、线型进行显示、存储。此外 ,还可应用 Auto CAD 绘图软件的 INSERT 命令 ,把多个图件根据实际工作的要求进行合并处理 ,或是根据图层的可见性 ,把某一层进行“冻结”或“解冻” ,单独输出某个层或某几个层中的地质图形元素。

(1)层名、线型名、颜色号与层中图形元素的约定。

软件给出的层名及选择的颜色线型等与图形元素的对应关系如下：

0 层	白色	连续线	图名	图边框	坐标网格	坐标标注
1 层	红色	连续线	巷道和其它井巷工程及其标注等			
2 层	黄色	连续线	地质界线、岩脉			
3 层	绿色	长虚线	地质界线、断层线			
4 层	青色	连续线	品位样品及样品标号			
5 层	蓝色	单点划线	表内矿矿体界线			
6 层	紫色	双点划线	表外矿矿体界线			
7 层	白色	短虚线	采空区界线			

(2)图形文件建立过程。根据上述约定建立地质图形文件 ,过程如下：

- ①在 Auto CAD 绘图状态提示符“ Command：”下 ,输入命令 ;LAYER；

- ②响应该命令的“NEW”提示,建立新层,并赋以层名 0,1,2……,7;
- ③响应“Set”提示,设置当前层,即选择工作层名;
- ④响应“Color”提示,按照前述约定设置当前层的颜色;
- ⑤响应“Ltype”,按照前述的层名与线型对应关系的约定,设置当前层的线型。

重复上述的③、④、⑤步骤,对每个层逐一进行设定,然后返回命令状态,便建立了一个带有层的图文件。

(3)图件的数字化仪输入。有了前述的准备工作,就可以通过数字化仪等将地质图件输入 Auto CAD 构成图形文件。输入过程中使用的主要命令有:

LAYER:设置当前层,并改变其可见性。

LINE:顺序地输入各种连续的、或间断的地质、工程描述等信息。

TEXT:标注品位样品号和各种井巷工程序号。

TRACE:描述品位样品的填空部分。

INSERT:插入这些图件中的特殊标注,如岩脉符号等,或用此命令合并多幅图件,形成大幅图。在使用此命令之前,应该用 Block 命令把要插入的图件定义成块文件。

上述的各种命令中的任一命令可连续执行多次,也可穿插执行不同的命令。如能通过数字化仪菜单辅助进行,则更为方便。

(二)绘制矿块地质剖面图

(1)用数字化仪输入矿块各分层地质平面图,并形成图形数据文件。

(2)用线性插值方法求出各剖面所切割的各分层平面的特征点坐标和特征值。各剖面的切点坐标和特征值形成的数据文件,存储起来。

(3)把切割出的各个特征点按照矿体和岩体的区别依剖面的顺序显示在屏幕上,并把该剖面的参考剖面显示在背景上,供连图参考。

参考剖面的选取原则如下:每个矿块事先都有矿块两侧所在的通风井或溜井所在的剖面图,此剖面图可作为靠近该剖面的新作的剖面的参考面。如果剖面离参考面太远,便把连好的前一剖面视为参考面。

(4)利用链盘或鼠标器采用人机对话方式移动光标进行连图工作。在程序中设定功能键来完成连接、删除、闭合等操作命令,连好某一区域,赋以相应的特征值,以区别矿体和夹石的类型。

这里需要说明的是,上述的求交点和连图过程以及下面编制采掘技术计划的程序都是用 BASIC 语言编写的。BASIC 语言程序通过 DXF 文件与 Auto CAD 进行数据通讯。

(三)编制地下矿采掘技术计划

地下金矿以矿块为采掘中心,各项地质、采矿数据的计算和矿山采掘技术计划的编

制都是围绕矿块进行的。

1. 计算矿块储量及数据准备

矿块储量的计算以矿块的分层地质平面图和若干剖面图为依据。

首先由前面所述的地质图库中提取矿块的分层地质平面图,通过接口程序提取其中与计算矿块地质储量有关的矿岩界线和巷道界线,并形成坐标数据文件。然后建立矿块分层块段面积模型(块段尺寸为 $0.4 \times 0.4\text{m}$)。在矿块分层块段面积模型中,共包含有该分层内矿体、矿体内工程、夹石和矿柱(工程包围的矿体)等信息。

由分层块段地质模型,可求出分层的矿体面积,进而求出矿块的地质储量,方法如下:

$$F = \frac{1}{3} \left[\frac{1 + \sqrt{S_1 \cdot S_2}}{S_1 + S_2} \right]$$

上式称为地质 F 值计算矿量法。式中 F 为 F 值系数, S_1 为上分层矿体面积, S_2 为下分层矿体面积。接下来计算矿体体积,公式为

$$V = F \cdot (S_1 + S_2) \cdot H$$

式中, H 为两分层的高差; V 为体积。

编制采掘计划要求矿块的块量必须剔除矿体内工程的矿量,同时还要求计算出矿块顶底柱、间柱的矿量和矿房的矿量等数据。因为分层块段模型中已有矿体内巷道的信息,所以只要输入巷道的高度,即可求出矿体内工程的矿量。除此之外,再输入矿块顶底柱、间柱的尺寸,矿层尺寸、千吨采准比等数据,即可按要求求出矿块的地质储量、矿房矿量、矿柱矿量、采准工程量、采切矿量等数据,为编制采掘计划作好准备。

2. 编制采掘计划

编制地下矿采掘计划时,既要考虑矿块间的几何关系,还要考虑矿块矿量间的大小搭配关系,以及矿块与采掘运输巷道间的关系和整个矿床的开采顺序。

首先根据提供的矿块资料和计算的各矿块的矿量等准备数据,把各矿块按照矿块的大小分类显示不同的颜色,同时把矿块所在阶段的主要运输巷道和工程,矿体的勘探线等信息显示在大屏幕彩色显示器上。这样,便把矿块的位置关系、采掘顺序、矿块大小配比、采掘与运输的关系在屏幕上清楚地呈现出来。计划设计者即可根据实际情况的要求和采矿技术的要求确定采掘的矿块。如果某一矿块过小或设计者需要对局部显示进行详细的察看,还可通过设定窗口的方法进行局部放大,把某一部分的情况全部显示在整个屏幕上。

软件通过鼠标器和光标键移动屏幕光标来选择需要采掘的矿块,用不同的功能键

“+”、“-”INS”等,对某一矿块进行采出、恢复和询问等工作。当确认某一矿块采出后,用户即可根据采掘技术计划的要求,按照矿块各个不同工程项目进行该矿块的各季度、年度的采掘量分配工作。因为下一年度的计划是在本年度的六月份左右编制,所以编制计划即要编制下一年度四个季度的计划,还要编制本年度的第三、第四季度的采掘计划。在计划量分配的同时,程序自动计算出季度合计、年度合计和矿块该项工程量的剩余量。当某一矿块编入计划时,该矿块在屏幕上的显示颜色发生改变,表示该矿块已被采出。

当某一阶段的计划编制完毕,接着可以顺序地编制下一个阶段的采掘计划。一个年度采掘计划可以重复编制,直至满足实际情况的要求并使用户满意为止。

编制好的采掘技术计划可根据矿山的要求打印各种报表。此外,还可根据各矿块的采掘量的大小和地质赋存情况的关系,绘制矿体采掘状态投影图和阶段矿块采掘状态平面图,并通过 Auto CAD 软件和绘图机绘图输出。

(四)绘图机图纸输出

下面进一步说明如何通过 Auto CAD 软件,用绘图机输出矿块分层地质图、剖面图以及上述的矿块采掘状态平面图等。

这里设所要输出的图件都是以 DXF 文件格式存储的。矿山工程图件,除了本身的内容之外,通常都带有标准的坐标网格和图题标栏等。因此应提前制作好坐标网格和图题标栏的图形块文件,然后在输出工程图件时利用 Auto CAD 的 INSERT 命令,很方便地把坐标网格和图题标栏等加入到要输出的矿山工程图件中,形成一个合乎工程规格要求的图件。下面假设上述工作已完成,说明绘图机绘图输出过程。

(1)接通绘图机电源,待绘图机自检完毕,装入绘图纸和笔。

(2)在 Auto CAD 的“Command”状态下输入 Plot 命令,并选择绘图输出的范围,一般情况下都是选择“Display”。

(3)响应软件提示,根据图层和颜色、线型对应关系的约定,详细设置输出图层的输出参数,选择绘图单位、笔宽、绘图原点以及填充方式等。

(4)确定绘图输入的化例。比例的定义是绘图机输出的多少 mm 相当于屏幕绘图状态的一个绘图单位。如果假定一个绘图单位代表 1m,选择输出的比例是 0.25,则绘图机画出的 0.25mm 是实际图纸的 1m,因此该图纸的比例是 1:4000,若选择 0.5,则是 1:2000。

(5)微机暂停输出,再次提醒用户装纸或笔,确认后按回车键,绘图机便启动绘图。

二、露天矿计算机辅助短期开采计划软件实例

下面介绍加拿大某公司专为编制露天矿短期计划而开发的软件——STEPS(Short—

Term Planning and Scheduling)系统。

STEPS 系统包含一套计算机程序、具有如下功能:数据采集与显示;矿石储量报表及统计;经济及成本模型;交互图形设计;露天坑评价;短期进度计划。每一种功能是由系统中的子模块完成的。各模块之间的关系如图 3-5-1 所示。每一个模块包含有若干具有独立功能的程序。整个系统由菜单控制,使用方便。

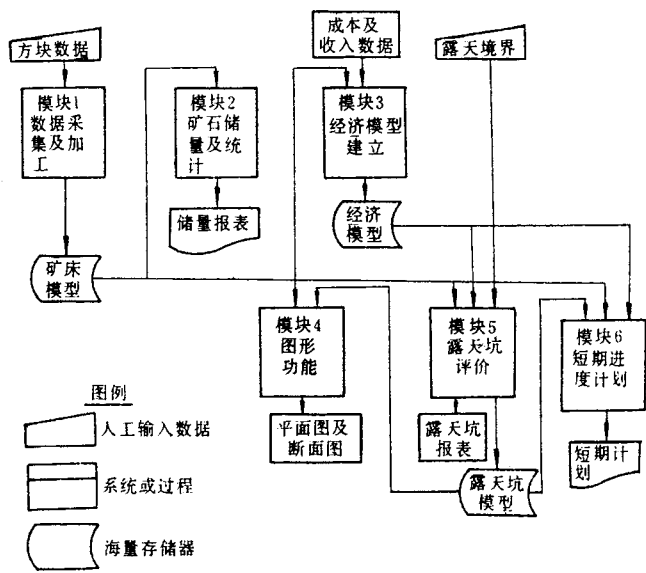


图 3-5-1 STEPS 系统的程序框图

软件编制计划的基础是三维矿体块段模型,模型的块段尺寸由用户确定,其中包含的变量有地质数据,比重及品位值等。

模块 1 包含若干个程序,用于采集方块数据并加以验证。利用这些程序,可以访问方块模型的任何部分,以便在获得更多信息时修改变量值。一旦采集好数据并确认后,可以在方块数据基础上完成矿石储量报表、简单的统计及吨位——品位分析。

该系统除了能建立三维矿体块段模型外,还能建立三维经济模型,这一功能由模块 3 完成。经济模型的大小及维数与矿体模型相同,不过它含有由矿石收益与采矿成本算出来的某些经济指数。采矿成本包括钻孔、爆破、装载等固定成本和与方块位置有关的运输成本等可变成本。

模块 4 是绘图功能模块,用来绘制平面图与断面图,其上表明选定方块的数据、台阶轮廓线或露天坑的断面。

模块 5 用于评价数字化的露天矿境界。评价露天坑境界时,计算圈定的矿石量、废

石量。在用户规定边界品位后,还可以求出吨位、品位以及经济指标。矿坑境界评价完毕后,就可用模块 5 编制短期采剥生产计划。

模块 5 是 STEPS 系统的核心。用户可利用这个模块,通过人机对话作出进度计划。在计算机屏幕上,可以有选择地显示当前露天坑中的采矿工作面及计划开采范围的图形。用户利用“橡皮筋式”交互式画线技术控制作业面的位置,在露天坑内模拟各种开采增量。用计算机迅速计算这些增量,使用户可能对大量开采方案进行研究、比较和优化。

屏幕可显示方块模型的平面图形及当前的露天坑综合台阶图,在图形输出的其它部位显示各种报表。如果矿体模型及露天坑境界很大,可以对露天坑进行局部放大。

计划工程师利用上述技术,能在设计工作帮、道路与坡道位置以及水坑布置时,得到高度的精确性。

由若干事先定义的专用功能键控制程序的执行。每一个键选择不同的程序,例如移动工作面位置、改换水平、计算、图像放大等等。用户利用这些功能键,动态地模拟开采过程与计算开采增量,既不费力,又少出错,提高系统的稳定性。

三、软件实例之三——Minenet 软件系统

采矿设计软件包 Minenet 是芬兰某公司开发的,用于公司的矿山及采矿工程设计。该软件已被公司一个矿山的地测及采矿设计部门用于日常工作中,该矿的采矿设计工作包括地质及矿石储量的数据更新,矿体及采矿场的三维模型建立,采场方案评价,长短期生产计划及采准进度计划的编制与控制,以及矿山测量。Minenet 软件适用于大多数常规工作。可用来更新全部原始数据、生成断面图及平面图,以及优化采矿设计等等。

(一)Minenet 软件系统

(1)硬件。Minenet 能在任何使用 Unix 操作系统的计算机上运行。现在的 Minenet 一些新版本,用于使用 Ultrix 操作系统的 VAX11/750 小型机以及使用 AIX 操作系统的 IBMRT 工作站及 IBM/2 80 微型机。Minenet 的最低硬件需求包括 4 兆字节 RAM 的主存,115 兆字节硬盘及一个浮点运算器。

(2)可行性研究。Minenet 的两个模块——“矿石储量估算”及“经济计算”用于可行性研究。“经济计算”的输入包括开工费、闭坑费、间接费与作业费等。这些费用是根据公司其他矿山的经验进行估计的。矿区以及境界内的矿石储量由 Minenet 估算,Minenet 根据金属价格及选矿数据计算矿石价值。

(3)采场设计及生产进度计划。建立不同采场方案的模型进行评价。将数据输入“采场寄存器”,并在其中进行采场的经济评价和排序。最优方案的数据传输给“生产进

度计划”模块,按照不同的时间长度及矿石的价格对生产作业及其后的作业进行估价。

(4)系统的进一步开发。Minenet 的开发工作将继续进行。将增加一个露天矿设计及优化模块。也有可能跟外部的矿山测量和工程管理系统相接。开发工作中特别注意使系统便于使用。用户接口设备将被精心设计。

(二)软件在矿山的应用

该矿原为露天开采,达到露天矿经济底部时关闭了矿山。80 年代初期经航空地球物理勘探发现异常现象,经解释后表明原露天矿底部以下的矿体扩大了。于是在矿体处布置了钻探工程,探明矿体随深度增加而变厚。

利用 Minenet 软件系统,该矿进行了地质数据处理,矿山设计及评价等工作。

1. 矿体构模

将岩心登录中获得的信息,包括岩种、节理、岩石质量指标等,输入 Minenet,甚至在样品未到之前,亦可生成矿体初始的概略模型。样品一旦输入,即可在钻探断面图上画出矿体边界线,并用数字化仪输入 Minenet,不同的矿石品位区间可用颜色加以区分,然后用直方图进行检验。

当各断面图中的全部矿体边界线数字化输入计算机后,就可由这些断面图构造矿体的三维模型。这种模型的精度主要依赖于断面图的间距以及矿体边界线数字化时点的数量。在三维模型生成后,可以自由旋转,并按任意方位切割平面,生成采矿设计所需的断面图及平面图。图 3-5-2 是三维模型的纵断面图。

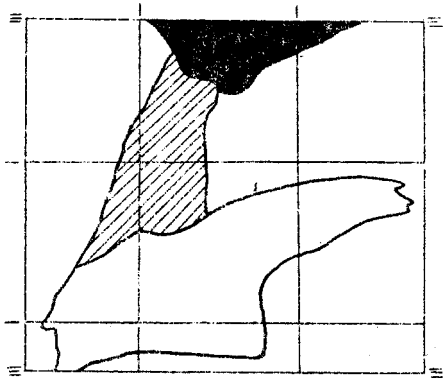


图 3-5-2 矿体的纵断面图

2. 储量估算

可用常规方法和地质统计学方法计算矿石储量。

3. 可行性研究

在知道矿体的吨位、品位和规模大小后,进行可行性分析。采矿成本、选矿成本、运输成本和间接成本是根据公司其他矿山的经验进行估计的。除此之外,还要考虑开工费、闭坑费以及一项临时费用的备用金。在收益方面,要计算或估算贫化率、选矿回收率、净冶炼回收率和价格等因素。进行灵敏性分析,检验金属价格、投资费用、选矿回收率等对建设项目综合经济效益的影响。

4. 井巷工程布置

根据可行性研究结果作出开矿的决策。设计过程的第一步是开拓设计。设计下山、通风天井和运输平巷,并将这些巷道的中心线数字化后存入 Minenet 系统。巷道中心线数字化后,根据预先确定的巷道断面可以自动构成巷道几何模型(图 3-5-3)。将溜井及破碎机硐室等加入巷道网络之中。

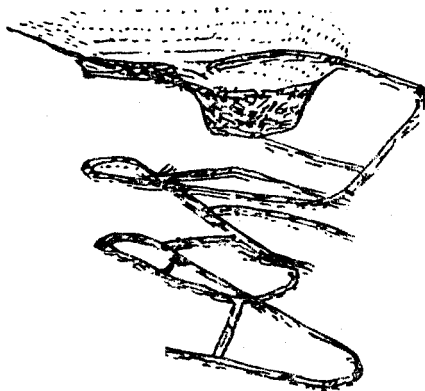


图 3-5-3 三维坑道图

全部开拓数据存入采场寄存器中,供以后使用。因为矿石巷道和废石巷道所用的费用不同,其数据分别存放。系统的生产控制模块产生开拓进度计划。每月修改一次采场寄存器来监控开拓进展情况。

随着开拓工作的进行,用金刚石钻探、泥浆取样及磁化率测量不断获得更多的信息,地质数据不断修正,并对三维矿体模型作必要的修改。应用 Minenet 系统,在矿体模型变化时可以容易地修改井巷布置,并很方便地分析这些变化对今后实现开采设计的影响。

5. 采场设计

为了进行可行性研究,在对地质调查的结果进行初步分析的基础上,完成了初步回采设计。当信息密度足够充分时,开始更详细的设计。

当采场边界已经确定时,计算被设计的采场的矿石储量。其中包括计算实际矿石损失,内部及外部废石贫化及保有的采准量。采场矿石储量存储在采场寄存器中备用。

Minenet 中的一个模块可以进行炮孔图的设计和绘图。模块可利用已有的采场模型、巷道模型及矿体模型并根据抵抗线、孔间距、倾角等参数 ,计算及绘制扇形炮孔图。输出的是孔数、孔长、炮孔总米数以及孔倾角及扇形排炮的体积(图 3-5-4)。

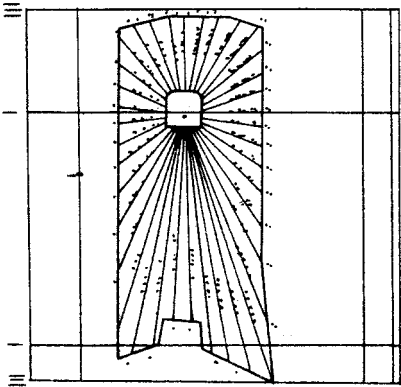


图 3-5-4 扇形炮孔

6. 生产进度计划

Minenet 系统还能编制短期和长期生产计划。二者均以被设计的采场及修正的地质信息为基础 ,并由月生产计划推导得出。