

常用零部件的 AutoCAD

二维参数化绘图技术

郭术义 著

(上册)



科学出版社

常用零部件的 AutoCAD 二维参数化绘图技术

(上册)

郭术义 著

科学出版社

北 京

内 容 简 介

AutoLISP 语言是 AutoCAD 软件自身携带的编程语言,利用它可对 AutoCAD 软件进行二次开发。本书结合螺栓、螺柱、螺钉、螺母等常用紧固件、常用传动齿轮的国家标准以及此类零部件的机械制图标准,利用 AutoLISP、DCL 关联了此类零部件内部之间参数关系,编制了基于对话框的交互式 LISP 程序,实现了常用紧固件、齿轮的参数化绘图。本书分上、下两册,其中上册主要内容包括绪论,螺栓、螺柱、螺母、螺钉、圆柱齿轮、锥齿轮等的参数化绘图技术;下册主要内容包括角接触、外球面、推力类轴承,滚轮、滚针类轴承,深沟球、调心球、圆柱滚子类轴承、蜗轮蜗杆,圆柱螺旋弹簧等的参数化绘图技术。

本书可供从事二维 AutoCAD 绘图工作有关的高校教师、研究生和企业工程技术人员参考。

图书在版编目(CIP)数据

常用零部件的 AutoCAD 二维参数化绘图技术(上册)/郭术义著. —北京:科学出版社, 2015

ISBN 978-7-03-044285-7

I. ①常… II. ①郭… III. ①零部件—机械制图—AutoCAD 软件 IV. ①TH126

中国版本图书馆 CIP 数据核字 (2015) 第 100612 号

责任编辑:童安齐 / 责任校对:王万红
责任印制:吕春珉 / 封面设计:耕者设计工作室

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂印刷

科学出版社发行 各地新华书店经销

*

2015 年 6 月第 一 版 开本:787×1092 1/16

2015 年 6 月第一次印刷 印张:17 3/4

字数:404 000

定价:70.00 元

(如有印装质量问题,我社负责调换<路通>)

销售部电话 010-62134988 编辑部电话 010-62130750

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

名目种类繁多的螺栓、螺柱、螺钉和螺母等紧固件作为标准件在二维装配图中经常用到，而 AutoCAD 等常用二维专业绘图软件本身却并不提供其二维图。当此类零件种类、参数调整时，绘图员不得不重新进行图形计算、修改，造成资源的极大浪费，装配图中出错概率大增。

渐开线圆柱齿轮和直齿锥齿轮尽管不是标准件，然而在实际的机械传动系统中应用却最广泛。由于齿轮设计过程涉及诸如齿数、模数、压力角、顶隙系数等参数，机械机构改变时，齿轮类型和参数改变，设计人员不得不重新计算各变量值，重新绘图，工作量非常大，出错几率大增。

为尽可能使设计人员从繁杂的绘图工作中解放出来，减少设计图中的错误，提高机械设计的效率，必须采取相应技术手段使得此类零部件可以实现二维参数化自动绘图。

AutoLISP 语言是 AutoCAD 软件自身携带的编程语言，利用它可对 AutoCAD 软件进行二次开发。本书结合螺栓、螺柱、螺钉、螺母等常用紧固件、常用传动齿轮的国家标准以及此类零部件的机械制图标准，利用 AutoLISP、DCL 关联了此类零部件内部之间参数关系，编制了基于对话框的交互式 LISP 程序，实现了常用紧固件、齿轮的参数化绘图，出图效率大大提高。

此外，LISP 文件和 DCL 文件中的英文和标点符号，如引号、分号都是纯英文状态输入的，从文件拷贝到本书中可能出现某些格式变化，请读者仿照例子做的时候注意修改；根据本书附录行文需要，删除了 LISP 文件和 DCL 文件中部分非必需的空格和换行符，请仿写编程时注意；限于篇幅，本书中零部件采用半幅绘制，另外半幅请仿照，直径、角度等标注仅作参考，读者采用本程序时可根据需要做适当修改。

本书得到了“河南省高等学校青年骨干教师资助计划”支持，在此表示感谢！

由于著者水平有限，书中难免存在不足之处，恳请广大读者批评指正。

目 录

上 册

前言

第一章 绪论	1
1.1 常用零部件参数化二维绘图背景	1
1.2 AutoCAD、Visual LISP 与 DCL	2
第二章 螺栓、螺柱的参数化绘图	4
2.1 六角头螺栓参数化绘图	4
2.1.1 图框与标题栏设计	4
2.1.2 六角头螺栓参数化绘图编程	6
2.2 方头螺栓参数化绘图	11
2.2.1 方头螺栓参数计算	11
2.2.2 方头螺栓左视图编程	11
2.2.3 方头螺栓主视图编程	12
2.3 T 形槽用螺栓参数化绘图	15
2.3.1 T 形槽用螺栓参数计算	15
2.3.2 T 形槽用螺栓左视图编程	15
2.3.3 T 形槽用螺栓主视图编程	17
2.4 等长双头螺柱参数化绘图	20
2.4.1 等长双头螺柱关键点设置	20
2.4.2 等长双头螺柱编程	20
2.5 螺栓、螺柱对话框设计	22
2.5.1 对话框界面、DCL 文件设计	22
2.5.2 对话框驱动程序设计	24
第三章 螺母的参数化绘图	29
3.1 螺母的图形及参数分析	29
3.1.1 方形螺母 C 级的图形及参数分析	29
3.1.2 六角螺母的图形及参数分析	29
3.1.3 圆螺母的图形及参数分析	30
3.2 方螺母参数化编程	31
3.2.1 编程前的准备	31

3.2.2 程序编制	31
3.3 六角螺母参数化编程	34
3.3.1 六角螺母关键点	34
3.3.2 六角螺母编程	35
3.3.3 六角开槽螺母关键点	39
3.3.4 六角开槽螺母编程	39
3.4 圆螺母参数化编程	42
3.4.1 圆螺母关键点	43
3.4.2 圆螺母程序编制	43
3.5 对话框程序	46
3.5.1 对话框文件	46
3.5.2 对话框驱动程序	47
3.5.3 对话框驱动	48
第四章 螺钉的参数化绘图	49
4.1 螺钉对话框程序编写	49
4.1.1 螺钉基本参数	49
4.1.2 螺钉对话框 DCL 文件	50
4.1.3 螺钉对话框驱动程序	50
4.2 半螺纹开槽盘头螺钉参数化	54
4.2.1 螺钉关键点分析	54
4.2.2 半螺纹开槽盘头螺钉编程	55
4.3 全螺纹开槽盘头螺钉参数化	57
4.4 六角头不脱出螺钉参数化	58
4.4.1 主视图作图过程分析	58
4.4.2 左视图作图过程分析	61
4.5 内六角凹端紧定螺钉的参数化	62
4.5.1 作图过程分析	62
4.5.2 编程	63
4.6 内六角锥端紧定螺钉的参数化	64
4.6.1 作图过程分析	64
4.6.2 完整程序	64
4.7 方头短圆柱锥端紧定螺钉的参数化	67
4.7.1 主视图编程绘制过程	67
4.7.2 左视图编程绘制过程	69
4.7.3 尺寸标注	70
4.8 方头长圆柱球面端紧定螺钉的参数化	71
4.9 开槽锥端紧定螺钉的参数化	74

第五章 圆柱齿轮的参数化绘图	77
5.1 齿轮参数概述	77
5.1.1 齿轮主要参数	77
5.1.2 齿轮主要参数计算	77
5.2 齿轮对话框设计	78
5.2.1 DCL 文件构成	78
5.2.2 DCL 界面设计	78
5.2.3 DCL 文件	78
5.3 齿轮自动绘制编程	80
5.3.1 齿轮主视图编程绘制	80
5.3.2 齿轮左视图编程绘制	84
5.3.3 图纸设置	86
5.3.4 对话框驱动程序	87
第六章 锥齿轮的参数化绘图	91
6.1 锥齿轮参数概述	91
6.2 锥齿轮参数编程绘图	92
6.2.1 作图环境设置编程	92
6.2.2 图纸标题栏、技术说明设置编程	93
6.2.3 锥齿轮主视图编程	95
6.2.4 锥齿轮左视图——轮毂键槽端面的程序编程	99
6.2.5 锥齿轮主程序编程	100
6.3 锥齿轮对话框编程	101
6.3.1 对话框结构图编程	101
6.3.2 对话框驱动编程	103
主要参考文献	105
附录 I 螺栓、螺柱 LISP 程序	107
附录 II 螺母 LISP 程序	158
附录 III 螺钉 LISP 程序	185
附录 IV 圆柱齿轮 LISP 程序	234
附录 V 锥齿轮 LISP 程序	253

下 册

前言

第七章 角接触、外球面、推力类轴承的参数化绘图	273
7.1 绘制滚动轴承流程图	273
7.2 分离角接触球轴承参数化绘图	273
7.2.1 分离角接触球轴承参数	273
7.2.2 分离角接触球轴承编程绘制基本过程	274
7.2.3 分离角接触球轴承程序绘制结果	275
7.3 角接触球轴承参数化绘图	276
7.3.1 角接触球轴承参数	276
7.3.2 角接触球轴承编程绘制基本过程	277
7.3.3 角接触球轴承程序绘制结果	277
7.4 圆锥孔处外球面球轴承参数化绘图	278
7.4.1 圆锥孔处外球面球轴承参数	278
7.4.2 圆锥孔处外球面球轴承编程绘制基本过程	279
7.4.3 圆锥孔处外球面球轴承程序绘制结果	280
7.5 带顶丝外球面球轴承参数化绘图	280
7.5.1 带顶丝外球面球轴承参数	280
7.5.2 带顶丝外球面球轴承编程绘制基本过程	281
7.5.3 圆锥孔处外球面球轴承程序绘制结果	281
7.6 推力球轴承参数化绘图	282
7.6.1 推力球轴承参数	282
7.6.2 推力球轴承编程绘制基本过程	282
7.6.3 推力球轴承程序绘制结果	283
7.7 推力圆柱滚子轴承参数化绘图	284
7.7.1 推力圆柱滚子轴承参数	284
7.7.2 推力圆柱滚子轴承编程绘制基本过程	284
7.7.3 推力圆柱滚子轴承程序绘制结果	285
7.8 对话框设计	285
7.8.1 对话框界面、DCL 文件设计	285
7.8.2 对话框驱动程序及主程序编制	288
第八章 滚轮、滚针类轴承的参数化绘图	293
8.1 滚针类轴承参数化绘图	293
8.1.1 滚针轴承关键点及关键程序分析	293
8.1.2 滚针轴承编程自动绘制	294

8.1.3 穿孔型冲压外圈滚针轴承关键点及关键程序分析	294
8.1.4 穿孔型冲压外圈滚针轴承编程自动绘制	295
8.2 滚轮类轴承参数化绘图	295
8.2.1 平挡圈滚轮满装滚针轴承关键点及关键程序分析	295
8.2.2 平挡圈滚轮满装滚针轴承编程自动绘制	297
8.3 滚轮滚针类轴承参数化绘图	297
8.3.1 无挡边滚轮滚针轴承关键点及关键程序分析	297
8.3.2 无挡边滚轮滚针轴承编程自动绘制	299
8.3.3 两面带密封圈、外圈双挡边的滚轮滚针轴承参数化绘图	299
8.4 对话框设计	300
8.4.1 对话框界面、DCL 文件设计	300
8.4.2 对话框驱动程序及主程序运行	303
第九章 深沟球、调心球、圆柱滚子类轴承的参数化绘图	304
9.1 深沟球轴承参数化绘图	304
9.1.1 深沟球轴承参数	304
9.1.2 深沟球轴承关键点分析	304
9.1.3 深沟球轴承程序编制	305
9.1.4 深沟球轴承程序运行	306
9.2 调心球轴承参数化绘图	307
9.2.1 调心球轴承参数	307
9.2.2 调心球轴承关键点分析	308
9.2.3 调心球轴承程序运行	309
9.3 圆柱滚子轴承参数化绘图	309
9.3.1 圆柱滚子轴承参数	309
9.3.2 圆柱滚子轴承关键点分析	310
9.3.3 圆柱滚子轴承程序运行	310
9.4 对话框设计	311
9.4.1 对话框界面、DCL 文件设计	311
9.4.2 对话框驱动程序及主程序运行	313
第十章 蜗轮蜗杆的参数化绘图	315
10.1 普通圆柱蜗杆参数化绘图	315
10.1.1 普通圆柱蜗杆参数	315
10.1.2 普通圆柱蜗杆关键点	315
10.1.3 普通圆柱蜗杆编程	316
10.1.4 对话框界面、DCL 文件设计	320
10.1.5 对话框驱动及主程序编程	321

10.2 普通圆柱蜗轮参数化绘图	322
10.2.1 普通圆柱蜗轮作图过程分析	322
10.2.2 普通圆柱蜗轮对话框及 DCL 文件	323
10.2.3 对话框驱动即主程序编程	325
10.3 直廓环面蜗轮蜗杆参数化绘图	325
10.3.1 直廓环面蜗轮蜗杆参数与关键点	325
10.3.2 直廓环面蜗轮蜗杆对话框	327
10.3.3 对话框驱动及主程序编程	328
10.4 平面二次包络环面蜗轮蜗杆参数化绘图	329
10.4.1 平面二次包络环面蜗轮蜗杆参数	329
10.4.2 包络环面蜗轮蜗杆对话框	332
10.4.3 对话框驱动及主程序编程	333
第十一章 圆柱螺旋弹簧的参数化绘图	335
11.1 圆柱螺旋拉伸弹簧参数	335
11.2 圆柱螺旋拉伸弹簧关键点	335
11.3 圆柱螺旋拉伸弹簧编程	337
主要参考文献	344
附录 I 常用角接触、外球面、推力类轴承 LISP 及 DCL 程序	346
附录 II 常用滚轮、滚针类轴承 LISP 及 DCL 程序	403
附录 III 深沟球、调心球、圆柱滚子类轴承 LISP 及 DCL 程序	470
附录 IV 蜗轮蜗杆 LISP 及 DCL 程序	524
附录 V 圆柱螺旋弹簧程序	581

第一章 绪 论

1.1 常用零部件参数化二维绘图背景

计算机辅助设计 (Computer Aided Design, CAD) 在全球一体化中的制造业中得到了广泛应用。经过近二十多年的发展, 我国的机械制造业已经普及了 CAD 绘图技术, 实现了“甩图板”的目标。现在工程界普遍缺乏掌握 CAD 二次开发的人才, 因而让广大工程技术人员进一步掌握 CAD 开发技术非常有必要。目前, 国内使用的 CAD 二次开发系统可以分为两大类: 一类是国内版的 CAD 系统, 另一类是国外商品化的 CAD 系统。国外 CAD 系统发展早、技术先进、产品成熟、售后服务周到, 因此其仍占据着主要市场份额。

在现有的 CAD 软件中, 美国 Autodesk 公司的 AutoCAD 以其强大的图形编辑处理功能、开放性结构以及良好的性价比, 获得了极大地市场优势, 得到了普遍应用, 是主要的 CAD 开发平台之一。

紧固件是用于紧固连接且应用极为广泛的一类机械零件, 常用于紧固连接两个或两个以上零件 (或构件), 使其成为整体构件。紧固件通常包括螺栓、螺柱、螺钉、螺母等十三类, 不仅种类繁多, 而且规格也很复杂。

齿轮传动比恒定、承载能力大、传动效率高、寿命长、尺寸紧凑、应用广泛, 是当前机械传动中最主要的一类传动。齿轮的分类方法很多, 其中按照根据啮合两齿轮轴的相对空间位置和轮齿的方向, 可分为圆柱齿轮、锥齿轮、交错轴斜齿轮。

蜗杆传动由蜗杆和蜗轮组成, 用于传递空间交错轴之间的运动和动力。蜗杆传动具有传动比大、传动平稳、传递功率范围大和结构紧凑等优点, 因此广泛应用于各种机器和仪器设备中。按蜗杆的几何形状不同, 蜗杆可分为圆柱蜗杆、环向蜗杆。

弹簧应用广泛, 种类繁多。按照弹簧的结构不同, 其可以分为丝形弹簧 (如圆柱螺旋弹簧)、板形弹簧、碟形弹簧、环形弹簧等; 按照承受的载荷不同, 其可以分为拉伸弹簧、压缩弹簧、扭转弹簧和弯曲弹簧等。

轴承用于支撑旋转轴及轴上零件, 有时也用于支撑其他转动零件。根据轴承工作时的摩擦性质不同, 可把轴承分为滚动摩擦轴承 (简称滚动轴承) 和滑动摩擦轴承 (简称滑动轴承) 两大类。而每一类轴承, 按其能承受的载荷方向的不同, 又可分为向心轴承或径向轴承 (承受径向载荷)、推力轴承或止推轴承 (承受轴向载荷) 和向心推力轴承 (同时承受径向和轴向载荷) 等。

以上零部件有国家系列标准, 在机械结构设计是最为常用的, 且大量的存在于机械装配图中。然而 AutoCAD 软件并没有以上零部件的二维图 (即使有, 也必须通过更换大量参数实现), 设计人员绘制 AutoCAD 图形时, 特别是当改变这些零部件的部分参数时, 必须重新查表, 确定参考点, 重新计算, 不断执行重复性工作, 极大地造成人力资源的浪费, 同时往往也导致绘制图形工作过程中错误不断出现。为尽可能减少 AutoCAD 图纸中的错误, 减轻绘图人员的负担, 使设计人员从繁杂绘图中解放出来, 把有限的精力投入到设计中, 提高机械设计效率, 针对此类零部件, 有必要对 AutoCAD 软件进行二次开发技术研究。

1.2 AutoCAD、Visual LISP 与 DCL

AutoCAD 是 Autodesk 公司出的一套功能强大的电脑辅助绘图软件,它是一款具备一体化、功能丰富、应用范围广等特性的先进设计软件,深得社会各界从事绘图工作的用户的青睐。其主要功能如下:

- (1) 具有完善的二维图形绘制功能。
- (2) 具有强大的图形编辑、修改功能。
- (3) 可以进行二次开发或自定义成专用的设计工具。
- (4) 支持大量的图形格式,在数据转换方面能力较强。
- (5) 支持多种外部硬件设备,例如专业的打印机与绘图仪等。
- (6) 支持多种模式的操作平台,让设计绘图多元化。
- (7) 简单易用,适用于不同领域的各类用户。

LISP 是广泛应用的一种程序设计语言,主要用于人工智能、机器人、专家系统、博弈、定理证明等领域。LISP 也被称为符号式语言,因为它处理的对象是符号表达式。LISP 语言语法简单,编写程序便捷、灵活。数据类型丰富,利用 LISP 语言可以很容易的定义或调用新的函数。

AutoLISP 是专门为 AutoCAD 二次开发而设计的编程语言,是嵌在 AutoCAD 内部的一种 LISP 语言和 AutoCAD 有机结合的产物,它采用了与 Common LISP 最相近的语法和习惯约定。

与 C、Pascal、Fortran 等过程性语言相比,AutoLISP 有以下显著特征:

(1) 内嵌式语言。不能脱离 AutoCAD 软件而单独运行,只能进入 AutoCAD 环境,在 COMMAND 状态下先装入程序,再执行。

(2) 解释型求值语言。

(3) 描述性语言。

(4) 表语言。

(5) 大量使用前缀表达式,表中第一项代表运算符和功能函数名。

(6) AutoLISP 不分数据与程序,面向符号处理而不擅长数值处理。

(7) 书写格式自由:不区分大小写,变元之间多个空格等于一个空格。

AutoLISP 既有 LISP 语言的智能化特性,又具有 CAD 强大的绘图能力,完美的将编程与绘图结合在一起,使设计与绘图融为一体,解决了复杂图无法绘出,或绘图过程较为复杂等难题,是广大机械设计师的福音,因此它在 AutoCAD 受欢迎的基础上更加得到了用户的喜爱,以方便快捷闻名于使用者之中。

利用 AutoLISP 可以进行工程分析计算,可以自动绘制复杂的图形、定义新的 AutoCAD 命令,可以驱动对话框、控制菜单。通过 AutoLISP 可以为 AutoCAD 扩充具有一定智能化、参数化的功能,使设计人员的主要精力用于产品的构思和创新设计上,实现真正意义上的计算机辅助设计。

Visual LISP 是 AutoCAD 自带的一个集成的可视化 AutoLISP 开发环境,最早的 AutoLISP 程序需要用文本编辑工具如记事本等编辑,然后在 AutoCAD 中加载调试,很不方

便。从 AutoCAD2000 开始,有了集成的开发环境: Visual LISP。作为开发工具, Visual LISP 提供了一个完整的集成开发环境 (IDE),包括编译器、调试器和其他工具,可以实时调试 AutoLISP 命令,定义 AutoCAD 的效率。 Visual LISP 具有自己的窗口和菜单,但它并不能独立于 AutoCAD 运行。

Visual LISP 可视化编程提供了更多的实用功能:

- (1) 提供控制台,在控制台的命令行作 AutoLISP 命令,可以直接看到结果。
- (2) 彩色字符源代码检查,可以在编辑窗中同时显示 AutoLISP 和 DCL 源程序的命令、注释、提示等以各种颜色区分表示出来,易于检查。
- (3) LISP 程序自动缩进和标准格式化。
- (4) 括号匹配检查。
- (5) 多窗口同时编辑 LISP 和 DCL 文件,并提供 DCL 对话框预览功能。
- (6) 可直接将你的 LISP+DCL 文件编译成一个 ARX 程序,并可连接 ARX, VC, VB, ActiveX 的 AutoCAD 程序。

使用 Visual LISP 编程可避免 LISP 程序繁锁的调试过程,提高了编程效率,它不仅对 CAD 编程人员提供了新的强大编程工具,而且对每个 AutoCAD 使用者都提供了更简单易用的 LISP 环境。

对话框就是一种使用便捷的固定的窗口,它形象、直观,可以随意地修改、随意地输入,鼠标和键盘可以同时并用,因此它是一种深受用户喜欢的界面,更是一种先进的、流行的人机交互的界面。应用对话框可以方便直接地实现我们的子程序设计时的数据传输和信息的传输,很是方便。

AutoCAD 自带自己的对话框设计语言,即是对话框控制语言 (Dialog Control Language, DCL)。该语言以 ASCII 码的形式来定义对话框,对话框是由许多控件组成的,而控件的尺寸及功能则是由控件的属性来控制。因此,用户只需要提供所需要的最基本的信息, AutoCAD 就可以自动确定对话框的大小尺寸以及部件的布局。可编程的对话框的功能模块在初始状态已经预先定义了控件的类型,然后再通过将各个控件编写到列与行之中,来选择是否要在这些编组后的控件的周围添加封闭的边框与框架。

本书结合螺栓、螺柱、螺钉、螺母等常用紧固件、常用传动渐开线圆柱齿轮和直齿锥齿轮的国家标准以及此类零部件的机械制图标准,利用 AutoLISP、DCL 关联了此类零部件内部之间的参数关系,编制了基于对话框的交互式 LISP 程序,实现了常用紧固件、齿轮的参数化绘图。滚动轴承、弹簧和蜗轮蜗杆等常用零部件的参数化绘图参见本书下册。

第二章 螺栓、螺柱的参数化绘图

螺栓、螺柱种类形式很多,采用 AutoLISP 完全对所有螺栓、螺柱进行参数化绘图编程不太可能,在此以六角头螺栓、方头螺栓等为例阐述螺栓、螺柱的绘制过程,其他类型的参数化绘制编程参见附录。

2.1 六角头螺栓参数化绘图

2.1.1 图框与标题栏设计

应用 AutoLISP 编程,首先需要确定图形对象的图纸背景,在此以 A4 图纸为例,绘制图框和标题栏,为后续编程做准备工作。

1. 操作环境及参数初始化设置

```
(defun c:A4( / v1 v2 v3)
  (setq v1(getvar "osmode")) ;获取当前普通命令提示信息的状态
  (setq v2(getvar "cmdecho")) ;获取当前光标标记的状态
  (setq v3(getvar "blipmode")) ;获取当前对象捕捉的模式
  (setvar "osmode" 0) ;关闭对象捕捉
  (setvar "cmdecho" 0) ;关闭普通命令提示信息
  (setvar "blipmode" 0) ;关闭光标
  (setq ljmch(getstring "\n输入零件名称:"))
  (setq cl(getstring "\n输入零件材料:"))
  (setq shl(getint "\n输入零件数量:"))
  (setq shl(itoa shl))
  (setq bl(getstring "\n输入绘图比例:"))
```

2. 调用 command 函数,设置字体和线型,定义矩形对角点来绘制矩形线框

```
(command "style" "hz0" "gbenor,gbcbig" "" "" "" "" "")
(command "layer" "m" "xixian" "c" "1" "")
(command "limits" "0,0" "210,297")
(command "zoom" "a")
(command "line" "0,0" "210,0" "210,297" "0,297" "c")
```

3. 调用 command 函数,切换图层,绘制内侧矩形线框及标题栏线框

```
(command "layer" "m" "cuxian" "c" "7" "" "lw" 0.6 "" "")
(command "line" "25,5" "205,5" "205,292" "25,292" "c")
(command "line" "85,5" "85,27" "205,27" "")
(command "line" "85,13" "205,13" "")
(command "line" "100,13" "100,27" "")
(command "line" "120,13" "120,27" "")
(command "line" "132,5" "132,27" "")
```

```
(command "line" "162,5" "162,13" "")
(command "line" "180,5" "180,27" "")
```

4. 调用 `command` 函数，切换图层进行文字设置，填写标题栏（图 2.1）

```
(command "layer" "m" "xixian" "")
(command "line" "85,20" "132,20" "")
(command "layer" "m" "wenzi" "c" "5" "" "")
(command "text" "m"(list 108.5 9)5 0 "华北水利水电大学")
(command "text" "m"(list 147 9)5 0(strcat "材 料 " "cl"))
(command "text" "m"(list 171 9)5 0(strcat "数 量 " "shl"))
(command "text" "m"(list 192.5 9)5 0(strcat "比 例 " "bl"))
(command "text" "m"(list 92.5 23.5)5 0 "制 图")
(command "text" "m"(list 92.5 16.5)5 0 "校 核")
(command "text" "m"(list 156 20)10 0 "ljmch")
```

制图			螺栓	
校核				
华北水利水电大学		材料 40CR	数量1	比例 1:1

图 2.1 标题栏的绘制及填写

5. 还原环境设置，输出二维图（图 2.2），静默退出

```
(setvar "osmode" v1) ;恢复普通命令提示信息原状态
(setvar "cmdecho" v2) ;恢复光标标记原状态
(setvar "blipmode" v3) ;恢复原对象捕捉模式
(princ) ;静默退出
)
```

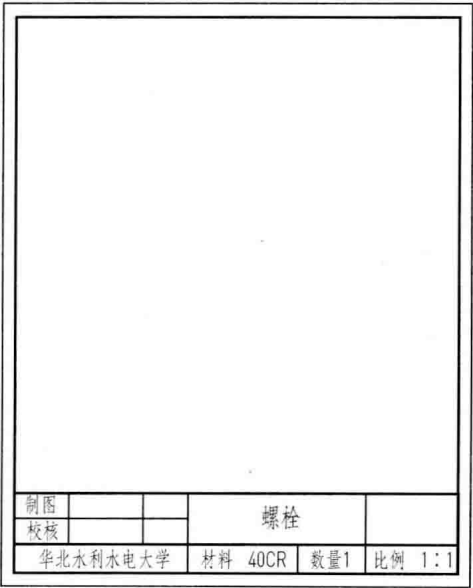


图 2.2 图框及标题栏

2.1.2 六角头螺栓参数化绘图编程

六角头螺栓的绘制过程分为三部分：第一部分，螺栓头部参数计算；第二部分，定位点确定及左视图绘制；第三部分，确定副定位点及主视图绘制。

1. 六角头螺栓参数设计

1) 六角头螺栓头部曲线弧度相关尺寸

如图 2.3 所示，对 d_s 进行计算（计算过程须依据机械设计手册或 GB/T 5782—2000 六角头螺栓参数表格）

$$d_s = 0.125e \tan 25^\circ$$

$$e = 2\sqrt{3} \div 3s$$

$$s \approx 1.5d$$

$$d_s = 0.086d$$

则

2) 倒圆轴向尺寸的确定

如图 2.3 所示，对 d_m 的确定。国家标准中圆形倒角半径 r 给出了最小值，所以为设计要求起见，此处取 r 最小值，则 $d_m = r$ 。读者根据编程需要可进行修改。

2. 六角头螺栓左视图编程

绘制六角头螺栓前需要寻找关键点，以该点为基准，定义其他相关点，再编辑程序源代码。

1) 程序开头要对编程操作环境进行初步设置、参数初始化

其程序代码为：

```
(defun c:bolt(/ dk d s l k r b dw alf) ;定义局部参数
(setq v1(getvar "osmode")) ;获取当前的目标捕捉类型,将其赋值给 v1
(setq v2(getvar "cmdecho")) ;获取当前的普通命令提示状态,赋值给 v2
(setvar "osmode" 0) ;关闭对象捕捉模式
(setvar "cmdecho" 0) ;关闭普通命令提示信息
(setq p0(getpoint "\n 输入螺栓的定位点:")) ;提示输入主要参数
(setq d(getdist p0 "\n 输入螺栓的犬径:"))
(setq l(getdist p0 "\n 输入螺栓的长度:"))
(setq s(getdist p0 "\n 输入螺栓头部尺寸:"))
(setq k(getdist p0 "\n 输入螺栓头长度尺寸:"))
(setq r(getdist p0 "\n 输入倒角半径尺寸:"))
(setq b(getdist p0 "\n 输入螺纹长度尺寸:"))
(setq dw(getdist p0 "\n 输入倒角直径尺寸:"))
(setq alf(getangle p0 "\n 输入螺栓的旋转角:"))
```

定义参数之间的代数关系，以便编程时减少运算，有

```
(setq d1(* 0.5 s)e(* 1.155 s)d2(* 0.5 e)d3(* 0.5 d)d4(* 0.425 d))
(setq d5(* 0.086 d)d6(* 0.5 dw)d7(- e d)d8(* 0.25 d7))
```

2) 左视图关键点的确定

图 2.4 为六角头螺栓左视图参数定义示意图,从图示所知, p_0 为参数设计过程中的关键点。以其为基准点,对六角头螺栓左视图中各点进行定义,编程如下:

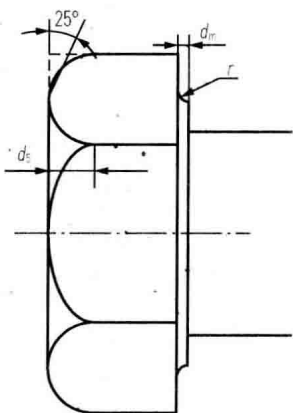


图 2.3 六角头螺栓头部示意图

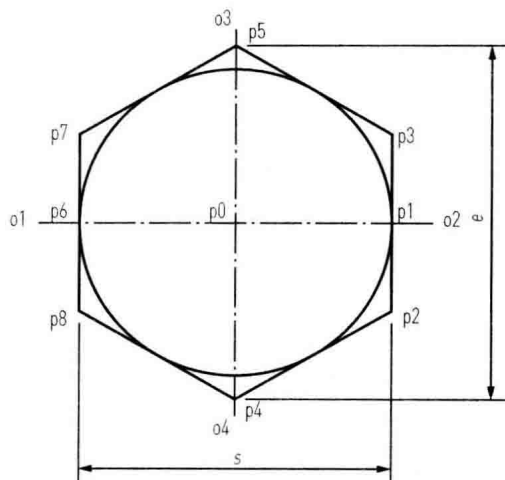


图 2.4 六角头螺栓左视图定义点示意图

```
(setq p1(polar p0 al f d1))
(setq p2(polar p1(+ al f(* -0.5 pi))(* 0.25 e)))
(setq p3(polar p1(+ al f(* 0.5 pi))(* 0.25 e)))
(setq p4(polar p0(+ al f(* -0.5 pi))d2))
(setq p5(polar p0(+ al f(* 0.5 pi))d2))
(setq p6(polar p0(+ al f pi)d1))
(setq p7(polar p6(+ al f(* 0.5 pi))(* 0.25 e)))
(setq p8(polar p6(+ al f(* -0.5 pi))(* 0.25 e)))
(setq o1(polar p0(+ al f pi)(+ d1 4)))
(setq o2(polar p0 al f(+ d1 5)))
(setq o3(polar p0(+ al f(* 0.5 pi))(+ d2 4))
(setq o4(polar p0(+ al f(* -0.5 pi))(+ d2 5)))
```

注: 上述程序段中 p_1 点是为确定端点 p_2 、 p_3 所引入的辅助点, p_6 点是为确定 p_7 、 p_8 所引入的辅助点。若不引入辅助点也可以根据参照点与待定点之间的位置关系来确定端点位置, 则此程序段可改写为:

```
(setq p2(polar p0(+ al f(/ -pi 6))(* 0.5 e)))
(setq p3(polar p0(+ al f(/ pi 5))(* 0.5 e)))
7
(setq p7(polar p0(+ al f(/ 5pi 5))(* 0.5 e)))
(setq p8(polar p0(+ al f(/ -5pi 5))(* 0.5 e)))
```

3) 调用 command 函数, 设置线型, 绘制左视图

创建当前图层, 绘制中心线: