HOPE

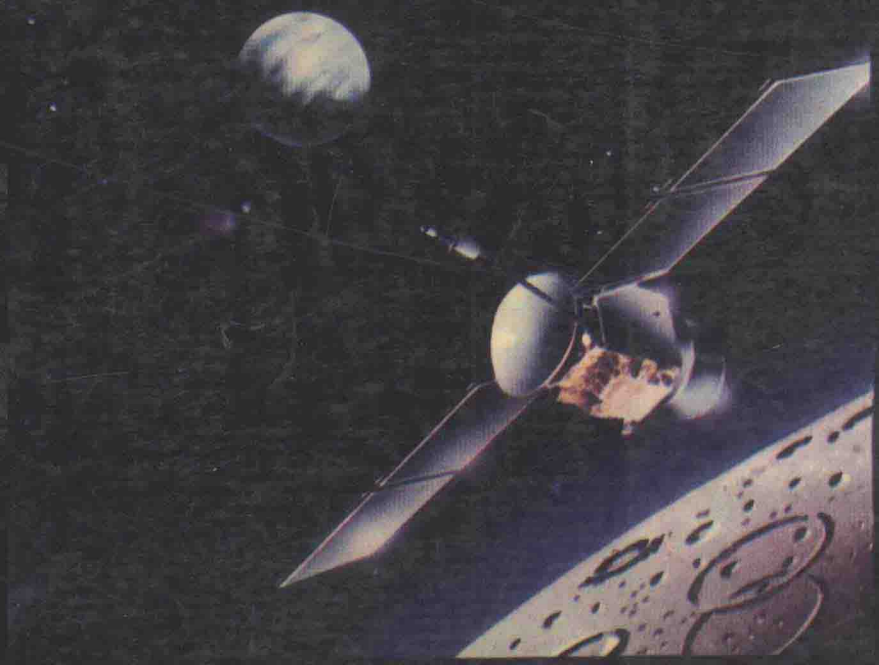
AutoCAD 10.0

计算机绘图软件包

AUTOLISP程序员参考手册

安装和性能指南

（下册）



中国科学院希望高级电脑技术公司

一九九〇年六月

AutoLISP 10.0版

程序员参考手册

目　　录

第一章 简介

AutoLISP是一种嵌入在AutoCAD的ADE－3软件包中的LSP程序设计语言。AutoLISP可使用户和AutoCAD的开发者能以极强功能的高级语言编写出适用于图形应用的宏程序和函数。LISP易学易用，十分灵活方便。

注 意：

不一定非要学会使用AutoLISP才能有效地应用AutoCAD；如果您没有计算机编程方面的经验，就只需阅读第二章中有关安装要求的内容——它会告诉你如何利用那些调用AutoLISP的AutoCAD菜单和应用程序。

如果您想能够熟练地编程，则应阅读其余章节内容，学会使用AutoLISP，以便将通用的AutoCAD设计软件包转变成适于专业需要的强有力工具。

本手册只是参考指南：它并不是LISP编程教科书。尽管在第三章中有如何使用AutoLISP的实例，我们仍建议用户先看一些有关LISP的教科书，以便能更快学会AutoLISP编程语言。推荐的参考书目有由Winston和Horn编写的《LISP》（第二版）和由Tony Hasemer编写的《Looking at LISP》，这两本书均由Addison Wesley出版社出版。LISP是一种具有很多语系的语言，它包括Mac LISP，Inter LISP，Zeta LISP和Common LISP。AutoLISP采用了和Common LISP（通用LISP）最相近的语法和习惯约定，但只是它的一个很小的子集，AutoCAD还为AutoLISP增设了很多专用函数。本参考手册列出了所有的AutoLISP函数及其使用方法。

1.1 为什么要采用LISP?

我们选择LISP作为AutoCAD的第一种应用接口语言，有如下一些考虑：

- LISP善长处理具有不同存储容量的各类数据对象，而这正是CAD系统如AutoCAD所需要的。
- LISP解释程序最适于那种专用于设计过程中的自由式的交互操作。
- LISP是最易于学习和掌握的编程语言。
- LISP也是我们选作用于研究和开发人工智能和专家系统的工具。
- 由于LISP本身具有特别简洁的语法，所以编写LISP解释程序变得相当简易；其程序尺寸也可非常短小。

今后，我们可能为AutoLISP提供其它应用接口，但Autodesk将保证长期支持AutoLISP，而且还要引进新的语言作为AutoLISP的补充，以为用户选用。

1.2 AutoLISP中的数据类型

AutoLISP支持下列几种不同的数据类型：

- 表
- 符号
- 字符串
- 实数
- 整数
- 文件描述符
- AutoCAD的实体"名"
- AutoCAD的选择集
- 子程序（内部函数）

在PC－DOS／MS－DOS系统中，整数为16位带符号数值，其范围是从－32768到＋32767。在32位机工作站上，AutoLISP的整数为32位带符号数值，其范围是从－2，147，483，648到＋2，147，483，647。但AutoLISP和AutoCAD之间的整数传输被限制在16位数值。实数用双精度的浮点数表示并具有至少14位有效精度。字符串可以是任意长度；它们的存贮空间是动态分配的。但字符串常数的最大长度为100个字符。

AutoLISP包括有几种用于二维和三维图形编程的内部函数。图形坐标的表示规则如下：

- 二维点
  以两个实数（X Y）的表形式表示，如：

  （3.4　　7.52）

  第一个值是X坐标，第二个值是Y坐标。

- 三维点
  以三个实数（X Y Z）的表形式表示，如：

  （3.4　　7.52　　1.0）

  第一个值是X坐标，第二个值是Y坐标，第三个值是Z坐标。

当AutoCAD要求某种类型的输入（如一个点或一个比例因子）时，则可使用该类型的AutoLISP表达式或使用可返回该类型结果的AutoLISP函数来提供所需的值。

1.3　AutoLISP的求值程序

每个LISP解释程序的核心就是其求值程序。求值程序读入用户的输入行，对它进行计算，并返回其结果。下面是AutoLISP的求值过程：

- 对于整型数，实型数，字符串，文件指针和子程序，以它们本身的值作为结果。
- 符号以它们当前的约束值作为计算结果。
- 表是根据它的第一个元素进行计算的。
  如果第一个元素计算的结果是一个表（或空nil），那么整个表就被假定为函数定义，在对函数的计算过程中用表中剩余的元素作为函数的变元。

如果第一个元素的计算结果是一个内部函数（子程序）名，那么表中剩余的元素作为形式变元传送给子程序，并由子程序进行计算。

如果用输入一个AutoLISP表达式来响应AutoCAD的命令提示，则AutoLISP计算该表达式的值并打印出计算结果。然后AutoCAD的"Command："提示重新出现。当打印实数时，AutoLISP将只显示到6位有效数字。

如果键入或从文件中读入一个错误的表达式，AutoLISP可能显示下列信息：

n>

这里的n是一个整数，它指示出没有封闭的左括号的级数。如果这种提示出现，必须要键入n个右括号，以退出这种状况。一种常见的错误是漏掉了正文串中闭合的双引号（"），在这种情况下，右括号将被解释为引起来的字符，键入右括号也不起作用。要纠正这种错误，先要键入一个双引号，然后键入n个右括号。

1.4 词法规则

AutoLISP的输入可以采用好几种形式。它们可以在运行AutoCAD时从键盘键入，可以从ASCII文件读入，或者可以从字符串变量读入。在所有这些形式中，必须要遵守一定的约定：

- 符号名称可以由除下列字符之外的所有可打印字符序列组成：

  （　）．'"；

- 下列字符用于结束符号名称或数值常数：

  （　）'"；（空格符）（行结束符）

- 表达式可扩展到多行。

- 符号之间留多个空格等价于留一个空格。
  虽然行首缩进不是必要的，但使用行首缩进可使用户函数的层次结构更加清晰。

- 在AutoLISP中符号和函数名称（子程序）的大小写等效。符号名称不能以数字开始。

- 整型常数之前可以加一个任选的"+"或"-"字符。如上所述，其值域是从－32768到＋32767。

- 实型常数由一个或多个数字组成，后面跟着小数点，小数点后是一个或多个数字；例如，".4"并不被认作为实数，"0.4才是正确的写法。同样，"5"不被认作实数，"5.0"才是正确写法。实数可以用科学法表示；即：它可以有一个任选的"e"或"E"，其后是指数部分。

- 文字字符串是一个由双引号引入的字符序列。在引入的字符串中，可使用反斜线符（＼），以使得某些控制字符可被正确识别。目前这些控制符表示成：

\\	表示字符"\"
\e	表示escape（ESC）
\n	表示换行
\r	表示回车
\t	表示制表符（Tab）
\nnn	表示其八进制码为nnn的字符。

如下面的文字串将在新行上发出提示：

```
( prompt "\nEnter first point: " )
```

- 单引号字符可以用作为QUOTE函数的缩写，因此：

```
'foo
```

与下列函数等价

```
( quote foo )
```

- 从磁盘文件装入的AutoLISP程序中可包括注解。注解要以分号开头，分号后一直到行尾都为注解部分。例如：

```
; This entire line is a comment
( setq area ( * pi r r ) ) ; Compute area of circle
```

1.5 记号约定

本参考手册使用了一定的约定来描述函数有关特性。例如：

```
( moo <string> <number>... )
```

其中函数名必须按原来所示的输入。在函数名之后尖括号中的项表示所带的变元数目和类型。

在这个例子中，函数"moo"要求有两个变元：一个字符串和一个数。省略号（"..."）表示可在函数后面加额外的数字变元。当调用函数时，不能包括尖括号和省略号。

对于以上所示的"moo"函数的格式，下面对"moo"函数的调用都是有效的。

```
( moo "Hello 5 )
( moo "Hi"1 2 3 )
```

下面的例子不符合所述的格式，将会出错：

```
( moo 1 2 3 )     ( 第一变元必须是一个字符串 )
( moo "Hello" )   ( 至少应有一个数字变元 )
( moo "do" '( 1 2 ) )   ( 第二个变元必须是一个数，而不应是一个表 )
```

当任选的变元只能出现一次而不能重复出现时，这个变元就用方括号（"[ ]"）括起来，如：

```
( foo <string> [ <number> ] )
```

这里，函数"foo"需要一个字符串（string）变元并可以接受一个任选的数字变元。例如，下面都是对"foo"函数的有效调用：

```
( foo "catch" )
( foo "catch" 22 )
```

下面的例子不符合所描述的格式，将会出错：

```
( foo  44  13 )  （第一个变元应为字符串）
( foo "foe"  44  13 )  （变元太多）
```

1.6 最新修改与补充

本手册页侧的竖杠区域标明自前一版本以来所作的最新修改。一般说，只对重要的修改才这样标明。

1.7 出错处理

如果AutoLISP在求值过程中遇到错误，它将打印下列形式的出错报告信息：

```
error:  text
```

其中text是出错信息。如果 * ERROR * 函数有定义（非空），AutoLISP将执行这个函数（用"text"作为它的单个变元），从而代替打印出错信息。如果 * ERROR * 没有定义或为空，AutoLISP将停止求值，这时就会对所调用的函数进行回溯，其深度多达100级，并将其显示出来。

第 二 章　AutoLISP的安装

2.1　发行格式

AutoLISP语言同AutoCAD的ADE－3软件一起提供。对于PC／DOS／MS－DOS系统，文件"Acadl. ovl"就是AutoLISP的覆盖文件；对于其它系统，该文件名为"Acadl"。

AutoCAD程序盘中有一名为"readme. doc"的文件，打印出此文件；其中包含了对AutoCAD和AutoLISP有关文档的最新修改或更动。

对PC－DOS／MS－DOS系统，提供了一种名为Extended AutoLISP的LISP编程语言。它是标准AutoLISP的增强版，由三个文件"Acadlx. ovl"，"extlisp. exe"和"remlist. exe"组成。详见下述章节内容。

2.2　标准的AutoLISP

本节内容适于所有不采用Extended AutoLISP的系统。AutoCAD第10版要求：

- 能够支持AutoCAD的计算机，至少具有640KB内存和硬盘，以及PC－DOS／MS－DOS 2.0版或更高版本。
- 与之配套的AutoCAD及其ADE－3软件。

2.2.1　配 置

当配置AutoCAD时，有一个工作参数可用来控制用户是否使用AutoLISP。在PC－DOS／MS－DOS系统上，配置程序还会显示另外一些信息以提示用户是否采用Extended AutoLISP，如果用户要使用标准AutoLISP，可对其回答"No"，详见AutoCAD安装和性能手册。

2.2.2　环境变量

在PC－DOS／MS－DOS系统上，必需为AutoLISP的使用预留出一定量的内存空间。如果你在使用象AutoCAD AEC类的应用软件，可以查阅其有关参考手册中对环境变量LISPHEAP和LISPSTACK的推荐设置值。详见AutoCAD安装和性能手册以及本手册中的第六章内容。

2.3　扩 展AutoLISP（Extended AutoLISP）

Extended AutoLISP第1.0版要求：

- 装有被AutoCAD所支持的Intel 80286或80386微处理器的计算机，内存至少为640KB以及硬盘。
- 至少为512KB的AT型扩展内存（非Lotus／Intel／Microsoft型扩存）而且该部分扩展内存不能用于其它目的（如RAM盘）。
- PC／DOS／MS－DOS 2.0版或更高版本。
- 以之配套的AutoCAD及其ADE－3软件。

如果以上要求均被满足，你便可选用Extended AutoLISP而不用标准AutoLISP，Extended

dutoLISP在80286／80386的"保护"方式下工作且常驻内存，这便保证AutoLISP程序及其取用的数据量都远超过标准AutoLISP的限度。使用Extended AutoLISP而不使用标准AutoLISP还有一大益处，即可使640KB以内的剩余内存用于AutoCAD的I／O分页存取作业。Extended AutoLISP不能在PC／XT型的计算机上运行，这主要因为8088或8086 CPU不支持"保护"方式下的操作。

除了在内存用法上的不同之外，Extended AutoLISP和标准AutoLISP在功能上没有任何差别，函数定义和程序运行都是完全相同的。

2.3.1 配 置

当配置AutoCAD时，有一个工作参数可用来控制用户是否使用AutoLISP。在PC－DOS／MS－DOS系统上，配置程序还会显示另外一些信息以提示用户是否采用Extended AutoLISP，如果用户要使用标准AutoLISP，可对其回答"No"，详见AutoCAD安装和性能手册。

2.3.2 环境变量

环境变量LISPXMEM用来控制那部分被Extended AutoLISP使用的扩展内存区的大小（如果用户没有指定LISPXMEM，那么Extended AutoLISP将用尽除开RAM盘占有的所有扩展内存区）。Extended AutoLISP会通知AutoCAD已有那部分扩展内存被自己占用，从而使得AutoCAD不能再将其作为I／O分页存取区使用。

分配给AutoLISP使用的扩展内存必需以"栈"的方式来设置和取用，如果你在使用象AutoCAD AEC类的应用软件，可查阅其参考手册中关于LISPSTACK的推荐设置方法（Extended AutoLISP不使用环境变量LISPHEAP）。

详见AutoCAD安装和性能手册以及本手册中第六章内容。

2.3.3 使用Extended AutoLISP

Extended AutoLISP作为单独的程序来提供，即EXTLISP，如果要使用必需在起动AutoCAD之前运行它。EXTLISP是一种中断和常驻内存（TSR）型的程序，它有一部分驻存在扩展内存中，它与AutoCAD的通讯是通过覆盖文件"Acadlx. ovl"来实现的。如果你愿意，可将需用的EXTLISP命令写入"Autoexe. bat"文件中。

如果需要，你还可将EXTLISP从内存中清除。为此，可在DOS提示下运行REMLISP命令，REMLISP还给DOS那部分曾被EXTLISP占用的空闲内存区。如果EXTLISP是最后装入的TSR程序，这将会恢复运行EXTLISP以前的内存状态。

2.4 文件"Acad. lsp"－自动装入功能

每及AutoCAD图形编辑阶段开始时，AutoLISP会自动装入名为"Acad. lsp"的文件（如果其存在），你可将常用的一些函数定义写在该文件中，这会使得在你每次开始编辑图形时，它们能被自动地定义。详见第四章中DEFUN函数的有关内容。

第 三 章　通过绘制花园路径的PATH命令来介 绍 AutoLISP

AutoCAD的主要威力在于能实现AutoCAD的用户化。在设计AutoCAD时，我们试图将尽可能多的功能交到用户手中。随着用户对AutoCAD的使用和对它逐渐熟悉，用户会经常发现，希望它应具备常用的某些功能。用户可能会着手将常用的命令序列加到屏幕，按钮，或图形板菜单中。用户可以定义新的线型、阴影线图案、或正文字体。当你在做这些时，你是在利用AutoCAD的"开放结构"（open architecture）－－即可将它扩展成和铸造成为个人设计工具的能力，它能和你想的和做的相吻合。

AutoLISP语言具有扩展AutoCAD的最强的能力。这种工具和AutoCAD ADE－3一起提供，它将LISP编程语言结合于AutoCAD之中。用AutoLISP编写程序，可以为AutoCAD增加新的命令，因此你可以拥有我们自己的软件开发者修改AutoCAD的大部分能力。

下面我们将为AutoCAD增加一条新的命令。在这过程中我们将介绍AutoLISP是怎样工作的，并示例说明如何使用AutoLISP的功能来完成用户的任务。虽然我们要开发的这条命令是面向场地建筑的，但用户从中学到的思想概念与具体的应用领域无关。

3.1　预备知识

我们假定读者是一个相当熟练的AutoCAD用户－－即读者已熟知AutoCAD命令以及AutoCAD所用的基本概念。我们还假定读者已会用构成ASCII文件的文本编辑程序。我们将在编辑过程中编写程序，用户应能使用文本编辑程序进行我们所要求的工作。

在本例子中，我们将使用AutoLISP中的一些函数，本手册后面章节将详细介绍所有这些函数。

3.2　目标

我们的目标是开发一条新的AutoCAD命令，它用于绘制花园中的路径，并铺上圆形混凝砖块。我们的新命令将具有下列提示序列：

```
Command: PATH
Start point of path:  (起点)
End point of path:  (终点)
Half width of path:  (数)
Radius of tiles:  (数)
Spacing between tiles:  (数)
```

其中"起点"和"终点"指定的是路径的中心线。还指定了路径的半宽度，输入了圆形砖块的半