

第二十四章 ZEMAX 程序设计语言

介绍

➡ 这个功能只有 ZEMAX 的 XE 和 EE 版本才有

在 ZEMAX 的应用中 ZEMAX 程序设计语言 (ZPL) 为了方便使用宏语言而特意设计的语言。ZPL 提供了用户自行扩展的功能。这意味着如果你一种特别的计算或图解显示,而这些功能在 ZEMAX 中是没有的,那你就可以自行编写 ZPL 程序来做这些工作。ZPL 程序能被储存在磁盘里并能从 ZEMAX 内部被调用。你可以建立一个 ZEMAX 程序库,并且可以和其他 ZEMAX 用户一起共享。

ZPL 类似与 BASIC 程序设计语言,除了不是所有的 BASIC 结构和关键字都被支持,以及加了一些新性能和独特的功能来进行光线追迹以外。如果你熟悉 BASIC 语言,你将很快地学会 ZPL。如果你不懂 BASIC 语言,或者你以前根本不懂程序设计,请不要惊慌。ZPL 是容易使用的,这一章将给你一些简单的用法说明和例子,让你从头开始学习。

创建 ZPL 程序

如果一个已存在的程序所能实现的任务与你想要完成的任务相似,那么在这个程序的基础上创建一个新程序去实现这个功能可能是容易的。如果你正尝试着写你的第一个 ZPL 程序,你可能会想去看一下这一章末尾的例子一节。在 \MACROS 目录下也能找到一些 ZPL 宏指令的例子,它们在“ZEMAX 例子文件”一章中被介绍。

应用任何 ASCII 码文本编辑器(如 NOTEPAD 编辑器)去创建 ZPL 文件。只要你希望,你可以给文件起任何描述性文件名,但文件必须以 .ZPL 扩展名结尾。

运行 ZPL 程序

为了运行你的 ZPL 程序,从主菜单选择宏,编辑/运行 ZPL 宏,一个 ZPL 控制对话框将会出现,对话框里有着以下选项:

现有文件:一个可用的宏的下拉式列表。所有列出的宏都是以扩

展名 .ZPL 结尾的 ASCII 码文件。文件必须在为 ZPL 宏而设置的目录下,参见“文件菜单”一章中关于参数选择对话框中目录表的部分。

执行后关闭:如果这一项被选中,那么在执行宏指令后 ZPL 控制对话框将自动关闭。

安静模式:如果这一项被选中,默认的文本输出窗口将不再显示。这对于那些不产生有用文本的图表宏指令是十分有用的。

状态:再执行宏指令的过程中,ZEMAX 在这个区域显示状态信息,这个状态信息是用来说明正在执行的宏指令的“行”号。这个状态信息每四分之一秒更新一次。

终止:终止按钮将使目前正在运行的宏指令停止运行。

取消:如果一个宏指令正在运行,取消按钮将终止当前宏指令,如果没有宏指令在运行,取消按钮将关闭 ZPL 控制对话框。

编辑:编辑按钮将调用 Windows NOTEPAD 编辑器。这个编辑器可以用来修改宏指令和对宏指令进行改名。

查看:查看按钮将在一个文本窗口中显示宏指令文件的内容,在这里它可以上下滚动和打印出来。在查看窗口中不允许编辑。

从“现有文件”中选择要运行的宏指令,然后点击执行。

ZEMAX 将开始运行程序。任何从 PRINT 语句中输出的文本或错误信息将被放置在一个文本文件中。当宏指令终止执行以后,这个文本文件将在一个窗口显示出来。可以用 CLOSEWINDOW 关键字来抑制这个输出窗口的显示。

ZPL 概述

ZPL 程序包括一系列命令(称为语句),它们被储存在 ASCII 码文本文件中。你可以用 ZEMAX 外的任何文本编辑器来创建 ZPL 程序(你也可以编辑 ZEMAX 内部的程序,这将在后面部分进行说明)。对于大多数(但不是全部)命令,ZPL 语言使用与 BASIC 相似的语法结构。例如,语句

x=5

版权声明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者),“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的进行学术交流,如果您有任何意见和建议,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

是一种有效的 ZPL 语句。在这一简单的语句中要注意一些重要的东西。首先，变量不需要被说明。这意味着被称为变量的“x”在被赋予 5 这个值之前不需要已经存在。如果“x”已经被赋值，那么它现在将被重新赋值。因此变量在使用之前不需要被说明。第二，不需要特殊符号来终止一个语句，例如在 C 语言中的“;”。因此，每个 ZPL 语句都必须自成一。ZPL 支持行函数调用。下列形式的语句是允许使用的：

```
x= SQRT(5)
```

```
y= SINE(x)
```

```
z= SQRT(x+5*(7-x))
```

函数 SQRT (平方根) 和 SINE (正弦) 在 ZPL 中已被建立。有许多这样的单操作数函数，所有这些在后面部分详细说明。注意 ZPL 是不分大小写的，SQRT() 和 sqrt() 是一样的。ZPL 文档将使用一般惯例，函数和关键字用大写字母，其他的用小写字母。前面出现的所有语句都有一个共同的特点：它们都是赋值语句。这意味着等号右边的表达式将被求值，结果被赋予等号左边的指定变量。在 ZPL 中还有一种语句，称为关键字。

关键字的一个例子是 PRINT。关键字 PRINT 后面跟着一列术语，它们有逗号隔开，这些都将被打出来。例如，ZPL 语句

```
x = 3
```

```
y = 4
```

```
z = SQRT(x*x+y*y)
```

```
PRINT 'The hypotenuse is ',z
```

在屏幕上将打印出下列内容：

```
The hypotenuse is 5.0000
```

注意 ZPL 强制使运算符优先。在任何一对括号里，运算操作按着特殊的顺序执行。ZPL 使用下列由高到低的优先级：函数（如 SQRT），逻辑运算（如==），乘法和除法，然后是加法和减法。圆括

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！

号总是不考虑优先级，在这个意义上，它有着最高的优先级。

有许多其他的关键字将在后面部分被介绍。

ZPL 变量

变量为数值量提供了临时性的储存单元，当编写程序时这些数值量的精度是不知道的，但当运行程序时精度将被规定。当你需要一个新的变量时，ZEMAX 将为你完成大部分的工作。例如，前面出现的简单语句

$x = 5$

使 ZEMAX 给这新变量分配储存单元，并记录下与它有关的值。变量一旦被定义，它就可以使用在后面任何一个表达式中。然而，关于 ZPL 变量的使用也有一些规则。

首先，变量名不能包含任何 ZPL 用来逻辑运算和分界的“特殊”符号，如(,), =, +, -, *, /, !, >, <, ^, &, |, #, ; 和空格号。第二，一个变量不能与关键字和函数同名，例如 THIC 和 RAYX，由于 ZEMAX 是不分大小写的，所以你也不能通过用 rayX 和 Thic 来避开这一条规则。第三，任何一个变量名都必须少于 28 个字母。违反了这些规则，虽然程序运行看起来没有错误，但都将产生语法错误，根本不能得到你想要的结果。ZPL 中的变量数最多不得超过 200 个。任何 ZPL 变量都是以 64 位双精度数据形式储存。

ZPL 函数

ZPL 函数是可以用在赋值语句的右边，在形式上它与关键字是不一样的。函数可以不要自变量，也可以有一个自变量或多个自变量。所有的函数都将返回一个值。例如一个函数，如 PWAV() (主波长)，将返回一个不依靠自变量的值，因此它不需要自变量。但无任如何，圆括号仍然需要。

许多函数都有相对应的关键字。例如，函数 RADI()将返回与自变量相对应的表面的曲率半径(RADI(3)返回第三表面的曲率半径)。同样也有一个关键字 RADI 设定表面的曲率半径。详细信息参见关键字的说明。

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的旨在进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！

所有的函数都在下面的表格中列出。如果给出一个函数形如 FUNC(), 那么表明它不需要自变量, FUNC(x) 则表明它需要一个自变量, FUNC(x, y) 表明它需要两个自变量, 等等。

ZPL 函数表

函 数	自 变 量	返 回 值
ABSO (X)	数字表达式	表达式的绝对值
ACOS (X)	数字表达式	反余弦值 (弧度表示)
APMN (X)	表面编号	最小半口径值。对于星形通光孔径是指臂宽; 对于长方形和椭圆通光孔径是指口径的 X 方向半宽度
APMX (X)	表面编号	最大半口径值。对于星形通光孔径是指臂数; 对于长方形和椭圆通光孔径是指口径的 Y 方向半宽度
APXD (X)	表面编号	孔径的 X 方向的偏心值
APYD (X)	表面编号	孔径的 Y 方向的偏心值
APTP (X)	表面编号	描述指定表面口径类型的整数代码
ASIN (X)	数字表达式	反正弦值 (弧度表示)
ASPR ()	无	当前绘图设备的尺寸比例
ATAN (X)	数字表达式	反正切值 (弧度表示)
ATYP ()	无	系统光圈类型代码: 0 指 EPD, 1 指 F/#, 2 指 NA, 3 指随光栏尺寸移动
AVAL ()	无	系统光圈值
CONF ()	无	返回当前的结构编号, 在 1 和最大结构数之间, 包括这两个数

CONI (X)	表面编号	表面的圆锥系数
COSI (X)	数值表达式 (弧度表示)	表达式的余弦值
CURV (X)	表面编号	表面的曲率
EDGE (X)	表面编号	表面在指定半口径时的边缘厚度
EOFF ()	无	文件结束标记。如果已到文件末尾则返回 1 , 否则返回 0。仅在执行关键字 READ 后函数才有效
ETIM ()	无	从最后一个 TIMER 开始所经过的时间 (以秒表示)
EXPE (X)	数值表达式	以 e 为底数的指数值
函 数	自 变 量	返 回 值
EXPT (X)	数值表达式	以 10 为底数的指数值
FICL (vec#)	Vector # , 在 1 和 4 之间 , 包括 1 和 4	光纤耦合效率。参见下面关于 FICL 函数使用方法的内容
FLDX (X)	视场编号	指定视场的 X 方向的角度或高度
FLDY (X)	视场编号	指定视场的 Y 方向的角度或高度
FTYP ()	无	如果视场类型是以度数表示的角度则为 0 , 是物高则为 1 , 是理想像高则为 2 , 是实际像高则为 3 , 高度以镜头单位为单位 (见 UNIT)
FVCX (X) FVCY (X)	视场编号	指定视场的 X 方向和 Y 方向的压缩渐晕因子
FVDX (X) FVCY (X)	视场编号	指定视场的 X 方向和 Y 方向的偏心渐晕因子
FWGT (X)	视场编号	指定视场的权重

GABB (X)	表面编号	玻璃库里指定表面对应的玻璃的阿贝常数
GETT(window, line,column)	要记录的窗口数、行数、列数的数值	从任何一个打开的文本窗口中记录数值。这功能允许我们利用 ZEMAX 在窗口中显示的任意值进行计算
GIND (X)	表面编号	玻璃库里指定表面对应的玻璃的 d 光折射率
GLCA (X)	表面编号	指定表面球面顶点的 X 矢量
GLCB (X)	表面编号	指定表面球面顶点的 Y 矢量
GLCC (X)	表面编号	指定表面球面顶点的 Z 矢量
GLCX (X)	表面编号	指定表面球面顶点的 X 坐标
GLCY (X)	表面编号	指定表面球面顶点的 Y 坐标
GLCZ (X)	表面编号	指定表面球面顶点的 Z 坐标
GNUM (A\$)	字符串变量名	如果字符串 A\$是一种有效玻璃名，如 BK7，那么返回这种玻璃在玻璃库中的编号。关键字 GLAS 可以用玻璃编号来设定表面的玻璃类型。如果 A\$不是玻璃库中的玻璃名，返回 0。如果字符串是“MIRROR”，返回 -1
函 数	自 变 量	返 回 值
GPAR (X)	表面编号	玻璃库里指定表面对应的玻璃的相对部分色散系数
GRIN(s,w,x,y,z)	表面#，波长#，x，y 和 z 轴坐标	返回编号为 w 的波长在 s 表面的指定位置的折射率。对于梯度和非梯度介质都有效

INDX (X)	表面编号	主波长的折射率
INTE (X)	数值表达式	返回小于自变量的最大整数
LOGE (X)	正的数值表达式	表达式的自然对数值
LOGT (X)	正的数值表达式	表达式的常用对数值
MAGN (x,y)	X 和 y 是实数	X 平方和 Y 平方之和的平方根
MAXF ()	无	以度数表示的最大半视场角，或着以镜头单位表示的半物高或半像高。如果视场是由角度、像高或物高定义的，那么返回值不依赖于自变量
MAXG ()	无	当前使用的玻璃的数量
MFCN ()	无	返回当前的评价函数值
NCON ()	无	返回结构个数
NFLD ()	无	定义的视场个数
NPAR(surf, object,param)	Surf 是表面编号 ,object 是目标编号 , param 是要返回的值的参数编号	从非连续元件的编辑界面的参数栏中返回一个值
NPOS(surf, object,code)	Surf 是表面编号,object 是目标编号 ,code 是 0-6 , 分别代表要返回的位置相对于参考位置的 x,y,z 坐标 , x,y,z 倾斜	从非连续元件的编辑界面的位置栏中返回一个值
NSUR ()	无	指定表面的编号
NWAV ()	无	指定波长的编号
ONUM (A\$)	字符串名	如果字符串 A\$是一个有效的优化操

		作数名, 如 EFFL, 则 ONUM 返回这个操作数的 id 编号。关键字 SETOPERAND 可以用操作数 id 编号来设定评价函数编辑界面中的操作数类型。如果 A\$不是操作数名, ONUM 返回 0
函 数	自 变 量	返 回 值
OPDC ()	无	光程差。仅在 RAYTRACE 被调用后才有效
OPER(row,col)	优化函数编辑界面中的行(操作数编号)和列(数据类型)	从优化函数编辑界面的指定行和列中记录数据。行号与操作数编号一样; 列号中 1 指操作数类型, 2 指 int1, 3 指 int2, 4-7 指 Hx-Py, 8 指目标值, 9 指权重, 10 指实际值, 11 指贡献百分值。也可参见 MFCN 和关键字 SETOPERAND
OPTH (X)	表面编号	沿着光线到指定表面的光程长度。与 RAYT 和 RAYO 不同, OPTH 考虑了由衍射引起的相位变化, 如光栅、全息、二元光学等
PARn (X)	表面编号	表面参数 “ n ”
PMOD ()	无	如果近轴模式关闭则为 0, 否则为 1
POWR (x,y)	X 是正数, y 为任意数	y 的 x 次方。X 必须是正数, 否则用它的绝对值
PVHX ()	无	优化操作数 ZPLM 的 Hx 参数
PVHY ()	无	优化操作数 ZPLM 的 Hy 参数
PVPX ()	无	优化操作数 ZPLM 的 Px 参数
PVPY ()	无	优化操作数 ZPLM 的 Py 参数

PWAV ()	无	主波长编号
RADI (X)	表面编号	表面的曲率半径
RAGX (X)	表面编号	光线截止点的 x 球形坐标 , 仅在 RAYTRACE 被调用后才有效
RAGY (X)	表面编号	光线截止点的 y 球形坐标 , 仅在 RAYTRACE 被调用后才有效
RAGZ (X)	表面编号	光线截止点的 z 球形坐标 , 仅在 RAYTRACE 被调用后才有效
RAND (X)	正的数值表达式	均匀分布在 0 和表达式值之间的随机浮点值
RANX (X)	表面编号	表面法线的 x 方向的余弦 , 仅在 RAYTRACE 被调用后才有效
RANY (X)	表面编号	表面法线的 y 方向的余弦 , 仅在 RAYTRACE 被调用后才有效
函 数	自 变 量	返 回 值
RANZ (X)	表面编号	表面法线的 z 方向的余弦 ,
RAYE ()	无	光线追迹错误标记 , 没有错误则返回 0 , 仅在 RAYTRACE 被调用后才有效。详细信息参见有关关键字 RAYTRACE 的内容
RAYL (X)	表面编号	穿过表面后光线在 x 方向的余弦。仅在 RAYTRACE 被调用后才有效
RAYM (X)	表面编号	穿过表面后光线在 y 方向的余弦。仅在 RAYTRACE 被调用后才有效
RAYN (X)	表面编号	穿过表面后光线在 z 方向的余弦。仅在 RAYTRACE 被调用后才有效

RAYO (X)	表面编号	从前一表面到指定表面光线的光程长度。光程长度是长度乘以折射率，这两个数都有可能是负数。仅在 RAYTRACE 被调用后才有效。也可参见 OPTH 和 RAYT
RAYT (X)	表面编号	从前一表面到指定表面光线的光程长度。光程长度有可能是负数。仅在 RAYTRACE 被调用后才有效。也可参见 OPTH 和 RAYO
RAYV ()	无	如果无渐晕则为 0，否则为渐晕面的编号。仅在 RAYTRACE 被调用后才有效
RAYX (X)	表面编号	光线截止点的 x 坐标。仅在 RAYTRACE 被调用后才有效
RAYY (X)	表面编号	光线截止点的 y 坐标。仅在 RAYTRACE 被调用后才有效
RAYZ (X)	表面编号	光线截止点的 z 坐标。仅在 RAYTRACE 被调用后才有效
RELI (f)	相对照度	在指定视场位置的 RI
SAGG(x,y,z)	x、y 是编号为 z 的表面坐标,以镜头单位表示	表面上指定点的矢高,以镜头单位表示
SCOD (A\$)	任意一个代表 ZEMAX 表面类型的字符串变量或文字	字符串必须是有效的表面类型名,如 STANDARD , EVENASPH , PARAXIAL , 等等。使用的每一种表面类型的有效文件名都将在指定的数据报告种显示出来
函 数	自 变 量	返 回 值

SCOM(A\$,B\$)	任意两个字符串变量	如果两个字符串相等, SCOM 返回 0; 如果 A\$小于 B\$, SCOM 返回一个小于 0 的值; 否则返回一个大于 0 的值
SDIA (X)	表面编号	指定表面的半口径
SIGN (X)	数值表达式	如果自变量小于 0 则返回-1, 自变量是 0 则返回 0, 自变量大于 0 则返回 +1
SINE (X)	数值表达式 (弧度表示)	表达式的正弦值
SLEN (A\$)	任意一个字符串变量	在字符串 A\$中字母的个数
SQRT (X)	正的数值表达式	表达式的平方根
STYP (n)	表面编号 n	表面的表面类型代码, 参见 SCOD
SVAL (A\$)	任意一个字符串变量	字符串值。返回字符串 A\$的浮点值
TANG (X)	数值表达式 (弧度表示)	表达式的正切值
TMAS ()	总重量 (以克表示)	从第一面到像面之间所有镜头的总的重量 (以克表示)
THIC (X)	表面编号	指定表面的厚度
UNIT ()	无	如果当前单位类型是毫米、厘米、英寸或米, 则分别返回 0、1、2 或 3
VEC1 (X)	正的下标值	返回数组中指定下标的元素的值
VEC2 (X)	正的下标值	返回数组中指定下标的元素的值
VEC3 (X)	正的下标值	返回数组中指定下标的元素的值
VEC4 (X)	正的下标值	返回数组中指定下标的元素的值

VERS ()	无	返回一个四位数的版本号。最典型的版本号是 5500。以后 ZEMAX 版本将返回更大的版本号
WAVL (X)	波长编号	波长 (以微米表示)
WWGT (X)	波长编号	波长的权重
XMIN ()	无	图表窗口中的最小 x 坐标
XMAX ()	无	图表窗口中的最大 x 坐标
YMIN ()	无	图表窗口中的最小 y 坐标
YMAX ()	无	图表窗口中的最大 y 坐标

函数 FICL () 的使用

函数 FICL 是计算光纤耦合的。因为函数要使用许多自变量，所以自变量要先放在一个向量组里，然后再调用函数。这向量组必须是已定义的四维数组。放在向量组里的数值定义如下：

0 = 采样精度

1 = 波长#

2 = 视场#

3 = 忽略来源标志 (0 代表否，1 代表是)

4 = 输出方 NA

5 = 接收方 NA

6 = 输出方 x 角度错误

7 = 输出方 y 角度错误

8 = 接收方 x 角度错误

9 = 接收方 y 角度错误

10 = 接收方 x 方向偏差

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有 (本站均注明出处和作者)，“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系 (info@photics.net)，我们会在最短的时间内给您答复并采取相应的补救措施！

11 = 接收方 y 方向偏差

12 = 接收方 z 方向偏差

通过调用函数 FICL (n) 来计算光纤耦合 , 这里 n 是包含自变量列表的向量组的个数。

ZPL 数学运算

ZPL 宏指令包括基本的数学运算 , 如加、减、乘、除。各类语法结构显示如下 :

$x = y + z$

$x = y - z$

$x = y * z$

$x = y / z$

其余所有的运算只有通过使用 ZPL 函数或 ZPL 逻辑运算来实现 , 这些将在后面的章节中介绍。

ZPL 逻辑运算符

逻辑运算符被用来构造复合语句 , 这个语句的最后的的结果是 1 或者 0。大部分逻辑运算采用形式

(表达式) (运算符) (表达式)

这类似于数学语句 , 如 $1 + 2$ 。补集运算是用非运算符 “ ! ” , 它仅需一个自变量 , 它的形式是 ! (表达式)。逻辑运算使用了一个惯例 , 0 代表 “ 假 ” , 非 0 值代表 “ 真 ”。如果表达式的值为 0 (假) , 非运算返回 1 (真) ; 如果表达式的值为非 0 值 (真) , 则返回 0 (假)。非运算常用在 IF 语句中 , 例如 :

IF !x THEN x is zero. ”

其他逻辑运算也可用来作为 IF 语句的部分自变量。例如 , 一个 IF 语句可以包含两个条件 , 当这两个条件都是真时才能执行 THEN 语句 :

IF (x>1) & (y<2) THEN PRINT Both conditions are true. ”

版权申明 : 本文由 “ 光学在线 ” 收集及整理 , 所有版权归文章原始作者所有 (本站均注明出处和作者) , “ 光学在线 ” 是一个介绍光学及相关科学的专业网站 , 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展 , 不为任何赢利目的 , 所以传播此文的目的进行学术交流 , 如果您有任何意见和看法 , 请与我们联系 (info@photics.net) , 我们会在最短的时间内给您答复并采取相应的补救措施 !

这两个条件通过用&表示的“与”运算符联系起来。注意这里的圆括号是用来强制使括号里的内容优先。ZPL 支持的逻辑运算符列表如下：

ZPL 逻辑运算符

逻辑运算符	说 明
&	与，仅当两个表达式都为非 0 值时返回 1
	或，当至少有一个表达式为非 0 值时返回 1
^	异或，当仅有一个表达式为非 0 值时返回 1
!	非，当（表达式）为非 0 值时返回 0，否则返回 1
==	等于，当表达式相等时返回 1
>	大于，当左边的表达式大于右边的表达式时返回 1
<	小于，当左边的表达式小于右边的表达式时返回 1
>=	大于等于，当左边的表达式大于或等于右边的表达式时返回 1
<=	小于等于，当左边的表达式小于或等于右边的表达式时返回 1
!=	不等于，当表达式不相等时返回 1

ZPL 字符串变量及其运算

ZPL 支持字符串变量和基本的字符串操作。字符串变量不需要说明，但可以在任何时候用一个定义语句来创建字符串变量，例如：

```
newstring$ = Here is the new string. "
```

注意，字符串变量和数值变量以字符串末尾是否跟字母\$来区分。字符串变量可以用操作符“+”连接起来。例如：

```
C$ = A$ + B$
```

连接句中也可包含字符串常量：

```
total$ = 'A$ is ' + A$ + ' and B$ is ' + B$
```

也可以有用来摘录文本数据的字符串函数，例如：

```
title$ = $LENSNAME()
```

注意这里的函数\$LENSNAME()是以字母\$开头。这样可以认为函数是返回一个字符串结果。字符串函数可以用在定义语句中，例如：

```
this$ = 'Here is the lens title: ' + $LENSNAME()
```

字符串变量可以像其他字符串一样被打印出来：

```
PRINT 'Here is A$: ' + A$
```

注意函数 PRINT 可以仅仅打印字符串变量；如在打印语句中有串联操作或字符串函数则不被支持。正确的方法是先把字符串连成一个新的字符串，然后再打印新的字符串：

```
A$ = B$ + C$
```

```
PRINT A$
```

另外，逗号也可以作为连接操作符：

```
PRINT A$, B$, C$
```

字符串不能像这样直接打印：

```
PRINT $LENSNAME()    ! 不正确 !!!
```

其实正确的方法是先把函数值赋给一个变量，然后再打印这个变量：

```
Z$ = $LENSNAME()
```

```
PRINT Z$
```

\$STR()是一个很重要的函数。这个函数可以把任意表达式当成自变量，只要这个表达式的值一个数字。\$STR 在把一个数值数据转变为字符串的格式变换中十分有用：

```
A$=          The          expression          evaluates          to ' +
```

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和建议，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁰⁶

\$STR(SQRT(3*3+4*4+z*y))

其相反功能的函数是 SVAL(A\$)，它把字符串转化成浮点数。

下面的表格中列出了可用的字符串函数。

ZPL 字符串函数

函 数	说 明
\$COMMENT(i)	返回第 i 面的注释字符串
\$DATE()	返回当前的日期和时间字符串。它的格式在环境对话框中被规定。
\$FILENAME()	返回当前镜头的文件名，不带路径
\$FILEPATH ()	返回当前镜头的文件名，带完整路径
函 数	说 明
\$GLASS (i)	返回编号为 I 的表面对应的玻璃的名称
\$LEFTSTRING(A\$, n)	返回字符串 A\$左边的 n 个字母。如果 A\$字母个数少于 n , 剩余的空间用空格填补。这允许字符串有固定的长度格式。参见\$RIGHTSTRING
\$LENSNAME()	返回在通用数据系统对话框中定义的镜头标题
\$NOTE(line#)	返回在通用数据系统对话框中定义的注解信息
\$PATHNAME()	返回当前镜头文件的路径名。这对于确定镜头文件储存的哪个目录里很有用
\$RIGHTSTRING(A\$,n)	返回字符串 A\$右边的 n 个字母。如果 A\$字母个数少于 n , 剩余的空间用空格填补。这允许字符串有固定的长度格式。参见\$LEFTSTRING
\$STR(expression)	返回一个格式由关键字 FORMAT 确定的字符串。这个数字表达式可以是任意形式，包括常数、变量和函数的组合形式

\$UNITS()

根据当前镜头单位返回 MM、CM、IN 和 M 中的一个

ZPL 字符串逻辑运算

字符串逻辑运算与前面讲述的 (数值) 逻辑运算很像。关键的差别在于表达式是字符串, 而不是数字。可用的字符串逻辑运算符在下列表格中进行说明。

ZPL 字符串逻辑运算符

逻辑运算符	说 明
\$==	等于, 如果左右两边的字符串是一样的则返回 1
\$>	大于, 如果左边的字符串大于右边的字符串则返回 1
\$<	小于, 如果左边的字符串小于右边的字符串则返回 1
\$>=	大于等于, 如果左边的字符串大于右边的字符串、或者和右边的字符串一样则返回 1
\$<=	小于等于, 如果左边的字符串小于右边的字符串、或者和右边的字符串一样则返回 1
\$!=	不等于, 如果左右字符串不一样则返回 1

例如, 一个 IF 语句可以来比较字符串, 如下:

```
A$ = "TEST"
```

```
B$ = "TEST"
```

```
IF ( A$==$B$ ) THEN PRINT "Strings are identical."
```

ZPL 关键字

ZPL 关键字提供了直接指导程序流程、产生输出、执行一些重要任务 (如光线追迹和修改镜头规定等) 的功能。每个关键字都在下面的章节里被详细介绍。功能相似或者有联系的关键字被列在一起。

APMN, APMX

用途

版权声明: 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有 (本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的旨在进行学术性质的交流, 如果您有任何意见和看法, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

用来设定表面通光孔径的最小/最大半口径。

语法结构

APMN(surface) = (new_value)

APMX(surface) = (new_value)

说明

这两个关键字需要两个表达式，一个用来指定表面编号，另一个用来定义一个新的数值。求出表达式(surface)的值，然后取整，用这个整数来作为表面编号。如果这个表面编号小于 0 或者大于表面的数量，那么这条命令将被略过。否则，求出表达式(new_value)的值，被赋值来作为该表面通光孔径的最小 (APMN)或最大半口径 (APMX)。虽然通光孔径的半口径数据的意义与使用的孔径类型有关，但它还是适用于所有通光孔径，两者的关系在下表中说明。

通光孔径类型和 APMN 使用方法

孔径类型	APMN 用成	APMX 用成
无孔径 (None)	不用	不用
环形口径(Circular aperture)	设定最小半口径	设定最大半口径
环形挡光 (Circular obscuration)	设定最小半口径	设定最大半口径
星形(Spider)	设定臂宽	设定臂数
长方形口径 (Rectangular aperture)	设定 x 半宽度	设定 y 半宽度
长方形挡光 (Rectangular obscuration)	设定 x 半宽度	设定 y 半宽度
椭圆口径(Elliptical aperture)	设定 x 半宽度	设定 y 半宽度
椭圆挡光 (Elliptical obscuration)	设定 x 半宽度	设定 y 半宽度

自定义口径(User aperture)	不用	不用
自定义挡光(User obscuration)	不用	不用
浮动口径(Floating aperture)	不用	不用

例子：

$$\text{APMN}(3) = 1.75$$

$$\text{APMX}(3) = 3.50$$

APXD , APYD

用途

用来设定表面通光孔径的 x/y 偏心。

语法结构

$$\text{APXD}(\text{surface}) = (\text{new_value})$$

$$\text{APYD}(\text{surface}) = (\text{new_value})$$

说明

这两个关键字需要两个表达式，一个用来指定表面编号，另一个用来定义一个新的数值。求出表达式(surface)的值，然后取整，用这个整数来作为表面编号。如果这个表面编号小于 0 或者大于表面的数量，那么这条命令将被略过。否则，求出表达式(new_value)的值，被赋值来作为该表面通光孔径的 x 偏心(APXD)或者 y 偏心(APYD)。

例子：

$$\text{APXD}(3) = -3.6$$

$$\text{APYD}(5) = -1 * \text{APYD}(3)$$

AFTP

用途

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的旨在进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴¹⁰

用来设定表面的通光孔径类型。

语法结构

$\text{AFTP}(\text{surface}) = \text{integer_code}$

说明

这两个关键字需要两个表达式，一个用来指定表面编号，另一个用来定义一个新的数值。求出表达式(surface)的值，然后取整，用这个整数来作为表面编号。如果这个表面编号小于 0 或者大于表面的数量，那么这条命令将被略过。否则，求出表达式 (integer_code) 的值，取整后赋值来作为通光孔径类型。

通光孔径类型被存成正数代码形式，详见下表说明。

表面类型代码

正数代码	通光孔径类型
0	无(None)
1	环形孔径(Circular aperture)
2	环形挡光(Circular obscuration)
3	星形(Spider)
4	长方形孔径(Rectangular aperture)
5	长方形挡光(Rectangular obscuration)
6	椭圆孔径(Elliptical aperture)
7	椭圆挡光(Elliptical obscuration)
8	自定义孔径(User aperture)
9	自定义挡光(User obscuration)
10	浮动口径(Floating aperture)

例子

ATYP(3) = 1

ATYP , AVAL

用途

用来设定系统光圈的类型和值。

语法结构

ATYP = (code)

AVAL = (new_value)

说明

这两个关键字是用来设定系统光圈的类型和值。光栏类型用下列代码来定义：

系统光圈类型代码

光圈类型	代码
入瞳直径 (Entrance Pupil Diameter)	0
像空间 F/# (Image Space F/#)	1
物空间数值孔径 (Object Space Numerical Aperture)	2
通过光栏尺寸浮动 (Float By Stop Size)	3

例子

! Set the EPD to be 35

ATYP = 0

AVAL = 35.0

BEEP

用途

产生可以听见的嘟嘟声。

语法结构

BEEP

说明

当运算结束或者需要输入时用这条命令来提醒用户。

CLOSE

用途

关闭前面用 OPEN 命令打开的 ASCII 码文件。

语法结构

CLOSE

说明

参见关键字 OPEN 的说明

CLOSEWINDOW

用途

禁止显示默认输出窗口。

语法结构

CLOSEWINDOW

说明

CLOSEWINDOW 用来使 ZPL 宏指令进入“安静”模式。如果宏指令中的任何一行中有关键字 CLOSEWINDOW ,那么通常在宏指令执行结束时显示的文本窗口将不再显示。CLOSEWINDOW 对宏指令的执行不产生任何其他影响。

COLOR

用途

用来设定图表功能中的线条和文字的墨水颜色。

语法结构

COLOR (n)

说明

n 的值必须是整数。如果自变量是 0 , 颜色被设成黑色。否则 , 由这个整数决定的颜色将用在所有后面出现的图表模式中的线条和文字。12 种有效颜色加上黑色 , 用数字 0-12 表示。ZEMAX 通常用不同的颜色来表示不同的视场和波长 , 如颜色 1 表示视场 1 或者波长 1 , 等等。

CONI

用途

CONI 用来设定表面的二次曲线常数。

语法结构

CONI(surface) = (new_value)

说明

这两个关键字需要两个表达式 , 一个用来指定表面编号 , 另一个用来定义一个新的数值。求出表达式(surface)的值 , 然后取整 , 用这个整数来作为表面编号。如果这个表面编号小于 0 或者大于表面的数量 , 那么这条命令将被略过。否则 , 求出表达式(new_value)的值 , 然后赋值。

例子

CONI 1 = -1

相关关键字

UPDATE

CURV

用途

CURV 用来设定表面的曲率。

语法结构

$\text{CURV}(\text{surface}) = (\text{new_value})$

说明

这两个关键字需要两个表达式，一个用来指定表面编号，另一个用来定义一个新的数值。求出表达式(surface)的值，然后取整，用这个整数来作为表面编号。如果这个表面编号小于 0 或者大于表面的数量，那么这条命令将被略过。否则，求出表达式(new_value)的值，然后赋值。

例子

$\text{CURV}(3) = -\text{CURV}(4) + .001$

相关关键字

UPDATE

DELETE

用途

从电子表格中删除表面

语法结构

DELETE (n)

说明

n 这个值必须是一个整数表达式。参见 INSERT。

例子

DELETE 5

DELETE i+2*j

EDVA

用途

设定表面的特别数据。

语法结构

EDVA surf,ed_value,new_value

说明

表达式 surf 指明要修改的表面。表达式 ed_value 的值必须是与一个特殊数据值的位置对应的整数,有关表面的特殊数据值的内容可参见“表面类型”一章。求出表达式 new_value 的值,用来作为该表面特殊数据的更新值。

例子

EDVA 5, 6, x+y

END

参见 GOSUM

FINDFILE

用途

用来查找储存在磁盘里的文件名。

语法结构

FINDFILE TEMPNAME\$, FILTER\$

说明

这个关键字需要两个表达式,一个用来指定存放文件名的字符串变量,另一个用来指定包含“筛选器”字符串的字符串变量。这个筛选器字符串通常用来指定路径名和与想得到得文件类型相符合的通配符。见下面的例子。

版权申明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者),“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的进行学术交流,如果您有任何意见和看法,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

FINDFILE 对于要列出一个目录下所有的某一类型的文件或者分析大量类似的镜头文件是很有用的,只要调用一个带有不同筛选器的 FINDFILE 语句,然后再调用带有原始筛选器名的 FINDFILE 语句。每次调用一个带有新筛选器的 FINDFILE 语句,都将复位返回这个筛选器遇到的第一个文件。

例子

```
FILTER$ = C:\ZEMAX\*.ZMX "
PRINT Listing of all ZEMAX files in ; FILTER$
FINDFILE TEMPFILE$, FILTER$
LABEL 1
If (SLEN(TEMPFILE$))
    PRINT TEMPFILE$
    FINDFILE TEMPFILE$, FILTER$
    GOTO 1
ENDIF
PRINT No more files. "
```

FLDX, FLDY, FWGT, FVDX, FVDY, FVCX, FVCY

用途

FLDX 和 FLDY 用来改变指定的视场点。FWGT 用来改变视场的权重。FVDX 和 FVDY 用来改变 x 和 y 的渐晕偏心因子。FVCX 和 FVCY 用来改变 x 和 y 的渐晕压缩因子。

语法结构

FLDX (field number) = (new_value)

FLDY (field number) = (new_value)

FWGT (field number) = (new_value)

FVDX (field number) = (new_value)

FVDY (field number) = (new_value)

FVCX (field number) = (new_value)

FVCY (field number) = (new_value)

说明

这些关键字需要两个表达式，一个用来指定视场编号，另一个用来定义一个新的值。求出表达式(field number)的值，然后取整，用这个整数来作为视场编号。如果视场编号小于 1 或者大于视场的数量，那么这条命令被略过。否则，求出表达式(new_value)的值，再赋值。因为视场值的改变会影响光线坐标和表面的半口径，所以建议再改变任何视场设定后执行以下 UPDATE 命令。

例子

FLDX 1=0.0

FLDY 1=12.5

FWGT 1=1.0

相关函数

FLDX , FLDY , FWGT , MAXF , FTYP , FVDX , FVDY , FVCX , FVCY

相关关键字

UPDATE , FTYP

FOR , NEXT

用途

关键字 FOR 和 NEXT 定义了一个程序块，这个块将在一个循环里执行指定次数。

语法结构

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴¹⁸

FOR variable=start_value,stop_value,increment

Statements...

NEXT

说明

关键字 FOR 标志着一组要多次执行的语句的开始。FOR 需要一个变量作为计数器 (不一定要是整数) 来说明计数器的起始值、终止值和增量。关键字 NEXT 标志着这组语句的结束。FOR-NEXT 循环可以嵌套。FOR 语句和 NEXT 语句的数量必须一样多。

在到达 FOR 语句之前,关于循环开始、终止和增量的表达式的值就已经被求出来,并保存起来。甚至当确定这些值的表达式中含有一些在程序块中其值要改变的变量时,终止值和增量也不再重新计算。这些值在 FOR 循环开始使用时才有效。如果开始值和终止值一样,循环体正好执行一次;如果开始值小于终止值,直到计数器变量大于终止值循环体才停止执行;如果开始值大于终止值,直到计数器变量小于终止值循环体才停止执行。

例子

FOR i=1,25,1

PRINT i

NEXT

j=5

k=0

FOR i=j,j+5,2

k=i+j+k

NEXT

版权申明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者),“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的在于进行学术性质的交流,如果您有任何意见和看法,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

FORMAT

用途

为后面的 PRINT 语句规定数值精度格式。

语法结构

FORMAT m.n [EXP]

说明

整数 m 和 n 被一个小数点隔开。m 是指要打印的总的字母个数，这些字母中的一些可能是空格。n 是指小数点后的位数。因此，FORMAT 8.4 将使后面的 PRINT 语句打印出小数点后面带有 4 位数字的 8 个字母。FORMAT .5 将使 PRINT 语句显示 5 个小数位，总的位置数是任意的。FORMAT 只会影响 PRINT 语句的数值输出。如果一个数值太大了，不在 m 个数字位之内，那么 FORMAT 语句的 m 部分将被忽略。在符号 m.n 后的可选择的关键字 EXP 规定了是否使用指数形式。

例子

x = 1.123

FORMAT 12.0

PRINT 'An integer portion = ',x

FORMAT 12.8

PRINT 'A decimal portion = ',x

FORMAT 12.8 EXP

PRINT 'Exponential notation = ',x

FTYP

用途

FTYP 用来使视场类型在角度 (以度表示) 物高和像高三者之间转变。

语法结构

FTYP = (new_value)

说明

表达式(new_value)的值必须是 0、1、2 或 3 之中的一个。如果值是 0 则表示视场类型是角度(以度表示), 1 则表示是物高(以镜头长度单位表示), 2 则表示是理想像高(以镜头长度单位表示), 3 则表示是实际像高(以镜头长度单位表示)。视场类型的设定不会改变视场大小, 因此也不用使系统更新。不论视场类型什么时候被改变, 视场大小通常都是用关键字 FLDX 和 FLDY 来调整。

相关函数

FLDX , FLDY , FWGT , MAXF

相关关键字

UPDATE , FLDX , FLDY , FWGT

GDATE

用途

GDATE 用来将镜头文件名下的当前日期放到用户自定义图表界面中的文本框中。

语法结构

GDATE

说明

GDATE 主要用来使你自定义的图表看起来像其他的 ZEMAX 图表。

例子

版权声明: 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有 (本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的进行学术交流, 如果您有任何意见和看法, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

参见 GRAPHICS 部分。

GETEXTRADATA

用途

从特殊数据编辑界面中得到特殊数据值 ,并把它放到向量组变量 (VEC1、VEC2、VEC3 或 VEC4) 中的一个里。

语法结构

GETEXTRADATA vector_expression,surface_expression

说明

数据将被储存在指定的 VECn 数组变量中。例如 , 如果命令 GETEXTRADATA 1 , 5 被执行 , 那么表面 5 的特殊数据将被储存在 VEC1 中。数据将用如下的格式来储存 , 这里每行的第一个数字是指数组中的位置 :

0 : 向量中特殊数据值的个数

1 : 第一个特殊数据

n : 第 n 个特殊数据

有关特殊数据值的叙述请参见 “ 表面类型 ” 一章。

GETGLASSDATA

用途

从当前的玻璃库中得到任意一种玻璃的数据 ,并把它放到向量组变量 (VEC1、VEC2、VEC3 或 VEC4) 中的一个里。

语法结构

GETGLASSDATA vector_expression,glass_number

说明

数据将被储存在指定的 VECn 数组变量中。例如 , 如果命令

版权申明 : 本文由 “ 光学在线 ” 收集及整理 , 所有版权归文章原始作者所有 (本站均注明出处和作者) , “ 光学在线 ” 是一个介绍光学及相关科学的专业网站 , 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展 , 不为任何赢利目的 , 所以传播此文的目的旨在进行学术性质的交流 , 如果您有任何意见和看法 , 请与我们联系 (info@photics.net) , 我们会在最短的时间内给您答复并采取相应的补救措施 !

GETGLASSDATA 1, 32 被执行, 那么第 32 号玻璃的数据将被储存在 VEC1 中。玻璃的编号可用函数 GNUM 得到。数据将用如下的格式来储存, 这里每行的第一个数字是指数组中的位置:

- 0 : 向量中数据值的个数
- 1 : 公式编号: 1 代表 Schott, 2 代表 Sellmeier 1, 3 代表 Herzberger, 4 代表 Sellmeier 2, 5 代表 Conrady
- 2 : MILNUM
- 3 : Nd
- 4 : Vd
- 5 : -30 到+70 摄氏度时的热膨胀系数
- 6 : +70 到+300 摄氏度时的热膨胀系数
- 7 : 密度 (以克/立方厘米为单位)
- 8 : 与正常线之间的偏离值 P_{gf}
- 9 : 最小波长
- 10 : 最大波长
- 11-16 : 相对部分色散常数 (它的值与计算公式有关)
- 17-22 : 热色散常数

GETMTF

用途

计算当前载入的镜头文件的子午和弧矢 MTF、实部、虚部、相位或者方波响应曲线等的值, 并把这些值存放在向量组变量 (VEC1、VEC2、VEC3 或 VEC4) 中的一个里。

语法结构

GETMTF freq,wave,field,sampling,vector,type

版权声明: 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有 (本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的旨在进行学术性质的交流, 如果您有任何意见和看法, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

说明

自变量 freq 是指要求的以线对/毫米为单位的空间频率, 如果这个频率小于 0 或者大于截止频率, GETMTF 返回 0。自变量 wave 是指计算中使用的波长编号相对应的整数, 0 说明采用全部波长进行计算。自变量 field 必须是在 1 到最大视场编号之间的一个整数, 它的值指定了使用的视场。自变量 sampling 可以是 1(32*32) 2(64*64), 3(128*128), 等等....., 一直到 2048*2048。自变量 vector 必须是 1 到 4 之间的整数, 它规定了这些数据将被储存在哪个向量组里。自变量 type 是指数据类型: 1 代表 MTF, 2 代表实部, 3 代表虚部, 4 代表相位 (以度表示), 5 代表方波 MTF。如果任何一个自变量超出了规定的有效取值范围, 那么将用取值范围内最接近的一个数值来代替它。

数据将被以下面的格式返回到一个向量组中: 数组位置 0: 子午响应曲线; 数组位置 1: 弧矢响应曲线。

例子

! 这个宏指令计算当前载入镜头在 30lp/mm 处、采用全部波长、在最大规定视场处、网格密度为 32*32 (sampling=1) 时的 T&S 曲线, 数据将被存放在向量组 1 中, 下面就是的到该数据的全部语句:

```
GETMTF 30,0,NFLD(),1,1,1
```

```
PRINT Tangential response: ",vec1(0)
```

```
PRINT Sagittal response: ",vec1(1)
```

GETPSE

用途

计算当前载入镜头文件的衍射的点扩散分布函数, 并把这些数据存放在向量组变量 (VEC1, VEC2, VEC3 或 VEC4) 中的一个里。

语法结构

```
GETPSE wave,field,sampling,vector
```

说明

自变量 wave 是指与计算时使用的波长编号相对应的整数, 0 说明采用全部波长进行计算。自变量 field 必须是在 1 到最大视场编号之间的一个整数, 它的值指定了使用的视场。自变量 sampling 可以是 1(32*32) \ 2(64*64) \ 3(128*128) 等等..... , 一直到 2048*2048。自变量 vector 必须是 1 到 4 之间的一个整数, 它规定了这些数据将被存放在哪个向量组里。如果任何一个自变量超出了规定的有效取值范围, 那么将用取值范围内最接近的一个数值来代替它。

数据将被以下面的格式返回到一个向量组中:

数组位置 0: 在向量组中 PSF 数据点的总的数目。通常, 这个数为 $4*n*n$, 这里 n 是采样尺寸 (32, 64, 等等)。例如, 如果采样密度是 2, 那么瞳孔采样尺寸是 64*64, 在向量中将有 128*128 或者 16384 个数值。这里每个数值需要 8 位空间, 或者说总共需要 131kb 空间。如采样密度为 1024 则至少需要 8Mb 空间来存放向量组; 如果采样密度再高, 则需要 64Mb 或者更多的空间来计算 PSF。位置 0 也能返回其他数值, 如错误信息代码。如果位置 0 的数值为 0, 则说明计算失败; 如果为 -1, 则说明向量组太小, 容纳不下所有的数据, 使用关键字 SETVECSIZE 使数组增大; 如果为 -2, 则说明系统内存太小, 不能计算这个 PSF 数据。

数组位置 1 用 $4*n*n$ 空间把 PSF 数据的强度保存下来, 并将之归一化。第一个 $2n$ 的值代表第一行, 从 -x 到 +x 扫描; 接下来的 $2n$ 的值代表其他行, 从 +y 到 -y 扫描。

数组位置 $4*n*(n+1)$ 保存了各个数值点之间的间隔 (以微米表示)。

例子

!这个宏指令将计算当前载入镜头采用全部波长、在第一视场处、网格密度为 32*32(sampling=1)时的 PSF , 数据将被储存在 vector1 中。

SETVECSIZE 4500

版权申明: 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有 (本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的进行学术交流, 如果您有任何意见和看法, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

```
GETPSF 0,1,1,1
Np=vec1(0)
IF (np==0)
    PRINT  "PSF Computation aborted. "
    GOTO 1
ENDIF
IF (np==-1)
    PRINT  "SETVECSIZE too small for PSF data. "
    GOTO 1
ENDIF
IF (np==-2)
    PRINT  "Not enough system RAM for PSF data. "
    GOTO 1
ENDIF
PRINT  "There are ",np, " data points,spaced ",vec1(np+1),
" microns apart "
LABEL 1
GETSYSTEMDATA
```

用途

得到大部分特定的系统数据，如有效焦距、工作 F/#、照度分布因子和其他一些与特定面无关的数据。并把这些数据存放在向量组变量 (VEC1 , VEC2 , VEC3 或 VEC4) 中的一个里。

语法结构

GETSYSTEMDATA vector_expression

说明

这些数据被储存在指定的数组变量 VECn 中。例如, 如果命令 GETSYSTEMDATA 1 被执行后, 这些系统数据将被存放在 VEC1 中。数据将用如下的格式来储存, 这里每行的第一个数字是指数组中的位置:

- 0 : 向量中系统数据的个数
- 1 : 光圈值
- 2 : 照度分布因子
- 3 : 照度分布类型 (0 : 均匀分布 ; 1 : 高斯分布 ; 2 : 正切分布)
- 4 : 使用 Env 数据 (如果用则为 1 , 不用则为 0)
- 5 : 以摄氏度表示的温度 (仅当使用 Env 时有效)
- 6 : 以大气压表示的压力 (仅当使用 Env 时有效)
- 7 : 有效焦距
- 8 : 像空间 F/#
- 9 : 物空间数值孔径
- 10 : 工作 F/#
- 11 : 入瞳直径
- 12 : 入瞳位置
- 13 : 出瞳直径
- 14 : 出瞳位置
- 15 : 理想像高
- 16 : 理想放大率
- 17 : 角放大率

18 : 总长度

19 : 使用光线定位 (如果用则为 1 ; 不用则为 0)

20 : 光瞳 x 移位

21 : 光瞳 y 移位

22 : 光瞳 z 移位

23 : 光栏数值

GETVARDATA

用途

得到当前所有优化变量的数量、类型和值, 并把这些数据存放在向量组变量 (VEC1 , VEC2 , VEC3 或 VEC4) 中的一个里。

语法结构

GETVARDATA vector

说明

这些数据将被储存在指定的数组变量 VECn 中。例如, 如果命令 GETVARDATA 1 被执行, 那么数据将被存放在 VEC1 中。数据将用如下的格式来储存, 这里每行的第一个数字是指数组中的位置:

0 : n , 变量的个数

1 : 第一个变量的类型代码

2 : 第一个变量的参数编号

3 : 第一个变量的对象编号

4 : 第一个变量的值

5*q- 4 : 第 q 个变量的类型代码

5*q- 3 : 第 q 个变量的参数编号

5*q- 2 : 第 q 个变量的对象编号

5*q- 1 : 第 q 个变量的值等等.....

整数 q 从 1 到 n , 这里 n 是变量的个数。如果 n 为 0 , 则没有有效数据返回。在数组 0 位置的数 n 的值总是有效的。变量的类型代码将在下表中说明 , 表面编号、参数编号、对象编号的值可能与变量有关 , 也可能无关。

GETVARDATA 类型和 id 代码

变量类型	类型代码	表面编号	参数编号	对象编号
曲率	1	表面#	—	—
厚度	2	表面#	—	—
二次曲线常数	3	表面#	—	—
折射率	4	表面#	—	—
阿贝常数	5	表面#	—	—
相对部分色散 Pg,F	6	表面#	—	—
膨胀系数	7	表面#	—	—
参数值	8	表面#	参数#	—
特殊数据值	9	表面#	特殊数据#	—
多重结构操作数值	10	操作数#	结构#	—
不连续对象位置 x 坐标	11	表面#	—	对象#
不连续对象位置 y 坐标	12	表面#	—	对象#
不连续对象位置 z 坐标	13	表面#	—	对象#
不连续对象 x 方向倾斜	14	表面#	—	对象#
不连续对象 y 方向倾斜	15	表面#	—	对象#
不连续对象 z 方向倾斜	16	表面#	—	对象#
不连续对象参数	17	表面#	参数#	对象#

版权申明 : 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有 (本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的旨在进行学术性质的交流, 如果您有任何意见和建议, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

GETZERNIKE

用途

计算当前载入的镜头文件的泽尼克边缘系数,并把这些数据存放在向量组变量 (VEC1 , VEC2 , VEC3 或 VEC4) 中的一个里。

语法结构

GETZERNIKE maxorder,wave,field,sampling,vector

说明

自变量 maxorder 可以是 1 到 37 之间的任意一个值,它与要求的最高的泽尼克边缘限制有关。自变量 wave 和 field 是波长和视场编号对应的整数值。自变量 sampling 的值规定了适合该系数的网格的尺寸,sampling 可以是 1(32*32) 2(64*64) 3(128*128) 等等.....,一直到 2048*2048。自变量 vector 必须是 1 到 4 之间的整数,它规定了这些数据将被存放在哪个向量组里。如果任何一个自变量超出了规定的有效取值范围,那么将用取值范围内最接近的一个数值来代替它。

这些数据将以下列格式返回到一个向量组里:数组位置 1:波形的最高点;数组位置 2:光程差为 0 时的均方根(这个值没有实际意义,但可供参考);数组位置 3:主光线的均方根;数组位置 4:像中心的均方根;数组位置 5:变化量;数组位置 6:斯特列尔变化率;数组位置 7:均方根错误;数组位置 8:最大错误。余下的数组位置由实际的泽尼克系数数据组成,泽尼克数据 1 在数组位置 9,泽尼克数据 2 在位置 10,等等。

例子

! 这个宏指令计算了当前载入的镜头在波长 1、视场 1、网格密度为 32*32 (sampling=1) 处时的前面 37 个泽尼克边缘系数。这些系数将被存放在向量 1 中。先得到这些数据:

GETZERNIKE 37 , 1 , 1 , 1 , 1

! 现在还不能输出

版权申明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者),“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的旨在进行学术性质的交流,如果您有任何意见和看法,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

FORMAT 16.6

PRINT Peak to Valley : ",vec1(1)

PRINT RMS to chief : ",vec1(3)

PRINT RMS to centroid : ",vec1(4)

PRINT Variance : ",vec1(5)

PRINT Strehl ratio : ",vec1(6)

PRINT RMS Fit Error : ",vec1(7)

PRINT Maximum Fit Error : ",vec1(8)

i=1

label 1

FORMAT 2.0

PRINT Zernike # ",i, " = "

FORMAT 16.6

PRINT vec1(8+i)

i=i+1

IF (I<38) THEN GOTO 1

PRINT 'All Done! "

GLAS

用途

GLAS 是用来设定表面的玻璃类型。

语法结构

GLAS(surface)=(glass_number)

GLAS(surface)=glass\$

说明

这个关键字需要两个表达式，一个用来指定表面编号，另一个定义玻璃编号。如果玻璃编号是-1，则玻璃类型设成“平面镜”；如果玻璃编号是0，则玻璃类型为空格或者统一的折射率；否则玻璃编号必须是1到由MAXG返回的数值之间的一个数。GLAS也承认包含所要求的玻璃名称的字符串变量，如“BK7”。

例子

```
GLAS 3=MAXG()
```

```
A$= " BK7 "
```

```
GLAS 4=GNUM(A$)
```

```
GLAS 4=A$
```

相关关键字

UPDATE

GLENSNAME

用途

GLENSNAME 将使当前的镜头名放在文本框或者用户自定义的图表窗口的左上角。

语法结构

GLENSNAME

说明

GLENSNAME 主要是用来使你定义的图表看起来与其他的图表一样。

例子

参见 GRAPHICS 部分。

GOSUB、SUB、RETURN and END

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴³²

用途

这四个关键字是一起用来定义和调用 ZPL 宏指令文件中的子程序的。每个关键字都有其特殊的用途。GOSUB 指引程序流程到自定义的子程序；SUB 用来定义子程序名，同时也象征子程序体的开始；RETURN 象征着程序将在最近的 GOSUB 命令出现的地方继续执行；END 象征着程序将立即停止。

语法结构

参见例子部分中的结构范例。

说明

每个 ZPL 宏指令文件都可以有不超过 50 个的子程序。每个子程序都必须用 RETURN 语句来终止，在一个子程序体中只能有一个返回语句。如果定义了子程序，那么至少必须有一个 END 语句用来象征主程序体的结束，主程序体必须在文件的顶部。

在一个宏指令中可以使用不超过 50 个的“嵌套级”。例如，如果子程序 ABC 调用了子程序 XYZ，那么嵌套级为 2；如果子程序 XYZ 又调用了子程序 DEF，那么嵌套级为 3。

在 ZPL 中所有的变量都是全局变量，任何一个在子程序中使用或定义的变量，在主程序中同样继续存在。

例子

```
x=1
```

```
y=2
```

```
GOSUB add
```

```
PRINT " the sum of ", x, " and ", y, " is ", z
```

```
END
```

```
SUB add
```

```
z = x+y
```

RETURN

GOTO

用途

通常，每个程序行都将依次执行下去。但 GOTO 允许程序在任意一点继续执行。GOTO 一般使用在带有 LABEL 命令的程序中。

语法结构

GOTO label_number

GOTO text_label

说明

在程序中必须有一个带有相关 label_number 或者 text_label 的 LABEL 命令，否则程序将产生错误。

例子

LABEL 1

x=RAND(10)

IF x<=5 THEN GOTO 1

PRINT "X is greater than 5. "

GRAPHICS

用途

创建一个标准的 ZEMAX 图表界面，这个界面中带有作为副标题的刻度行。

语法结构

GRAPHICS

GRAPHICS OFF

说明

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴³⁴

如果 GRAPHICS 被单独定义，那么将产生一个标准的 ZEMAX 图表窗口。所有后面出现的图表命令将被发送到这个新建窗口中。GRAPHICS OFF 将关闭现有打开的图表窗口，然后显示这个关闭的窗口。

例子

GRAPHICS

xmx =xmax()

xmn =xmin()

ymx =ymax()

ymn =ymin()

xwidth =xmx-xmn

ywidth =ymx-ymn

xleft =xmn+(.1*xwidth)

xrigh =xmn+(.9*xwidth)

ytopp =ymn+(.1*ywidth)

ybott =ymn+(.7*ywidth)

LINE xleft,ytopp,xrigh,ytopp

LINE xrigh,ytopp,xrigh,ybott

LINE xrigh,ybott,xleft,ybott

LINE xleft,ybott,xleft,ytopp

GTITLE the rain in spain falls mainly on the plain ”

GLENSNAME

GDATE

GTEXT xmx/2,ymx/2,0, “ start this text in the center. ”

GTEXTCENT ymx*.05, “ center this text near the top. ”

GTEXT xmx*.05,ymx*.75,90, “ place me vertically near left edge. ”

GTEXT xmx*.15,ymx*.68,30, “ orient me at 30 degrees. ”

GRAPHICS OFF

产生的图表显示如下。

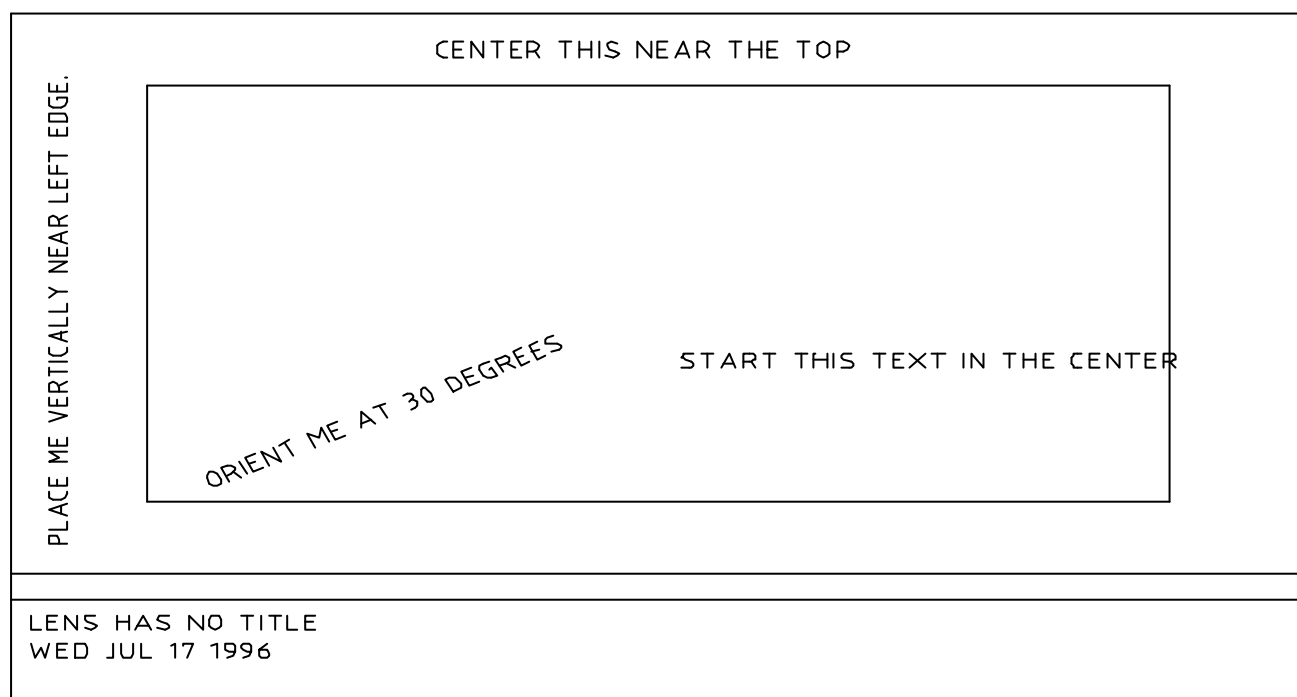


图 1

GTEXT

用途

GTEXT 用用户自定义文字作为图表块的标注。

语法结构

GTEXT x,y,angle,user_text

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和建议，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴³⁶

GTEXT x,y,angle,A\$

说明

坐标 x 和 y 指的是文字字符串显示的地方的左下角位置，“user_text”既可以是提供的字符串常量，也可以是字符串变量名。angle 规定了文字相对于图表框是如何旋转的，它的默认值是 0（水平）。也可参见 SETTEXTSIZE。

例子

参见 GRAPHICS 部分。

GTEXTCENT

用途

GTEXT 用用户自定义文字作为图表块的中心标注。

语法结构

GTEXTCENT y,user_text

说明

坐标 y 是指文字字符串 user_text 的垂直位置。也可参见 SETTEXTSIZE。

例子

参见 GRAPHICS 部分。

GTITLE

用途

除了文字需要被指定外，GTITLE 的其他部分和 GTEXT 类似，GTITLE 指定的文字显示在图表的标题栏的中心。GTITLE 有利于使你定义的 ZPL 图表看起来像标准的 ZEMAX 图表。

语法结构

GTITLE user_text

说明

GTITLE 主要用来使你定义的图表与其他 ZEMAX 图表看起来一样。

例子

参见 GRAPHICS 部分。

HAMMER

用途

调用锤形优化法则使用当前评价函数来优化当前镜头。

语法结构

HAMMER

HAMMER number_of_cycles

说明

如果没有提供自变量,那么锤形优化运行 1 圈;如果提供了一个自变量,那么这个自变量必须是 1 到 99 之间的一个整数,且锤形优化法则将运行指定的圈数。

相关函数

MFCN

例子

PRINT Starting merit function: ;MFCN()

HAMMER 3

PRINT Ending merit function: ;MFCN()

IF-THEN-ELSE-ENDIF

用途

IF 提供了条件程序的执行和分支。

语法结构

IF (expression)

(statements)

ELSE

(statements)

ENDIF

或者

IF (expression) THEN (statement)

说明

IF-ELSE-ENDIF 结构被用来选择执行跟在 IF 语句后面的语句群或者跟在 ELSE 语句后面的语句群,但不能全部执行。如果表达式的值为 0 则认为它是假的,否则则认为它是真的。这个表达式可以是任意一种有效的 ZPL 表达式类型,包括函数、变量、操作数和常量。虽然 ELSE 语句是随意的,但 IF 语句必须与 ENDIF 成对出现。IF-ENDIF 语句对可以随意嵌套。

IF-THEN 结构是单一指令的选择执行语句的简易形式。如果指定了一个 THEN 关键字,那么 IF 语句被终止,而且不需要 ENDIF 语句。IF-THEN 结构不支持 ELSE 关键字。

例子

x=1

y=2

IF (x<y)

PRINT 'x is less than y. '

ELSE

IF (x==y) THEN PRINT 'x equals y. '

```
IF (x>y) THEN PRINT  x is greater than y. "
```

```
ENDIF
```

```
INPUT
```

用途

INPUT 提供了一种在宏指令运行时提示用户输入数值或文本数据的方法。

语法结构

```
INPUT Prompt String ; variable
```

```
INPUT variable
```

```
INPUT Prompt String ; string_variable
```

```
INPUT string_variable$
```

说明

variable 必须是有效的变量名。如果变量名是一个字符串变量，那么输入的内容将被认为是文字字符串；否则则认为是数值。如果没有提供提示字符串，那么 INPUT 命令将使用一个“？”提示符。提示符一直显示在屏幕上，而且输入的内容总是只从键盘键入。

例子

```
INPUT Enter value for x: ; x
```

```
PRINT X= ; x
```

```
INPUT Enter a value for A$: ; A$
```

```
PRINT A$
```

```
INSERT
```

用途

INSERT 在电子表格中插入一个新的表面。

语法结构

INSERT(n)

说明

n 必须是一个整数表达式的值。也可参见 DELETE 和 SURFTYPE。

例子

INSERT 5

INSERT i+2*j

LABEL

用途

LABEL 为 GOTO 命令提供目标,详细内容参见“GOTO”部分。

语法结构

LABEL label_number

LABEL text_label

说明

label_number 必须是一个大于 0 的整数,如 1 或 7。如果使用文字标记,那么它不能包括空格和其他一些用作分隔符的特殊符号。LABEL 本身对程序流程没有影响。

例子

LABEL 7

LABEL startover

LINE

用途

版权申明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者),“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的进行学术交流,如果您有任何意见和看法,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

LINE 是图表显示中的简单的画直线的函数。

语法结构

LINE oldx,oldy,newx,newy

说明

LINE 将求出四个表达式的值，并画出指定点的连线。坐标系是指当前图表框架，而且被限定在由 XMIN、YMIN、XMAX 和 YMAX 指定的边界内。虽然仅只有整数像素值才可被准确划分，但 LINE 也接收实数值作为其自变量，并把这实数坐标值四舍五入，取与其最接近的整数值。LINE 仅在图表模式中才有效。线的颜色由当前的墨水颜色控制，而墨水颜色则用关键字 COLOR 指定。

例子

参见 GRAPHICS 部分。

LOADCATALOG

用途

为当前载入的镜头重新载入玻璃及其存在的目录。

语法结构

LOADCATALOG

说明

当镜头被载入时，如果它的一些相关的玻璃目录和数据文件，包括文件 COATING.DAT，还没有被载入，那么它们将被自动载入。然而，如果这些目录已被修改过了，也许是被 ZPL 宏指令自己改的，那么可能要用关键字 LOADCATALOG 来强制重新载入这些目录。除非 COATING.DAT 和玻璃 AGF 目录文件被修改过了，否则从当前的 ZEMAX 方案开始设计以来都不会使用这个关键字。

LOADLENS

用途

从磁盘中载入一个新的镜头文件来取代当前内存中的镜头文件。

语法结构

```
LOADLENS filename "[appendflag]
```

```
LOADLENS file$ [appendflag]
```

说明

LOADLENS 将从磁盘中载入一个新的镜头文件。如果文件名中包含完整的路径, 如 C:\MYDIR\MYLENS.ZMX, 那么指定的文件将被载入; 如果没有路径, 那么将使用默认路径作为镜头的路径 (参见文件菜单一章中环境下的内容)。

如果 appendflag 的值为 0 或者没有 appendflag, 那么 LOADLENS 只简单地载入文件; 如果 appendflag 的值大于 0, 那么文件被加到由 appendflag 的值指定的表面的后面。

例子

```
LOADLENS " COOKE. ZMX "
```

相关关键字

SAVELENS

LOADMERIT

用途

从磁盘中载入一个评价函数文件来取代当前的评价函数。

语法结构

```
LOADMERIT filename "
```

```
LOADMERIT file$
```

说明

LOADMERIT 将从磁盘中载入一个新的评价函数。如果文件名中

包含完整的路径, 如 C:\MYDIR\MYLENS.MF, 那么指定的文件被载入; 如果没有路径, 那么将使用默认路径作为镜头的路径 (参见文件菜单一章中参数选择下的目录)

NEXT

参见 FOR

NUMFIELD

用途

设置定义的视场的数目。

语法结构

NUMFIELD <expression>

NUMWAVE

用途

设置定义波长的数目。

语法结构

NUMWAVE <expression>

OPEN

用途

打开一个现有的 ASCII 数值文本文件, 让 READ 命令读入数据。

语法结构

OPEN filename "

OPEN A\$

说明

指定的文件名必须是提供的有效文件, 或者包含文件名的字符串变量。参见关键字 READ 和 CLOSE。通常在读完所有的数据后才关

闭文件。

例子

```
PRINT Reading the double-column file TEST.DAT! "
```

```
OPEN TEST.DAT "
```

```
READ x1, y1
```

```
READ x2, y2
```

```
READ x3, y3
```

```
CLOSE
```

OPTIMIZE

用途

调用优化法则来用当前评价函数优化当前镜头。

语法结构

```
OPTIMIZE
```

```
OPTIMIZE number_of_cycles
```

说明

如果没有提供自变量，那么优化在“自动”模式下运行，当运算法则探测到这个过程已在聚集在一点时优化结束。如果提供了自变量，它必须是 0 到 99 之间的整数值，优化法则将运行指定圈数。

相关函数

```
MFCN
```

例子

```
PRINT Starting merit function: ", MFCN( )
```

```
OPTIMIZE
```

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁴⁵

PRINT Ending merit function: ", MFCN()

OPTRETURN

用途

通过使用优化操作数 ZPLM 返回数值给优化法则。

语法结构

OPTRETURN (datafield) = (result)

说明

OPTRETURN 采用两个由等号隔开的自变量。表达式(datafield)的值必须在 0 到 50 之间。datafield 是指存放表达式(result)的值的数组位置。OPTRETURN 的唯一的目的是使在 ZPL 宏指令中计算出的值可以被最优化。

优化操作数 ZPLM 必须使用在评价函数中来调用 ZPL 宏指令，再用 OPTRETURN 得到返回的值。详细内容参见“优化”一章。

例子

x = SQRT(THIC(3)+RADI(5))

OPTRETURN j = x+5

OUTPUT

用途

指定文本输出的目的地,即可以输出到屏幕上，也可以输出到文件中。

语法结构

OUTPUT SCREEN

OUTPUT filename "

OUTPUT filename "APPEND

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁴⁶

OUTPUT A\$

OUTPUT A\$ APPEND

说明

如果单独定义了 OUTPUT SCREEN, 那么后面执行的 PRINT 语句将直接输出内容到屏幕上; 如果提供了一个有效地文件名, 那么后面执行的 PRINT 语句将输出内容到该文件名指定的文件中。为了关闭先前创建的文件, 可使用 OUTPUT SCREEN, 它将直接将后面的 PRINT 的内容输出到屏幕上; SHOWFILE 将关闭这个文件, 并把它发送到作为屏幕显示的文本阅览程序中; PRINTFILE 将关闭这个文件, 并将它再当前定义的打印设备上打印出来。

如果有关键字 APPEND 跟在文件名后面, 那么后面输出的内容将被加在这个文件的后面。否则, 这个文件的内容将被覆盖。

例子

```
OUTPUT x.txt "
```

```
PRINT "This will not appear on the screen, but in the file x.txt. "
```

```
OUTPUT SCREEN
```

```
PRINT "This will appear on the screen. "
```

```
OUTPUT x.txt "APPEND
```

```
PRINT "This will appear after the first line in the file x.txt. "
```

相关关键字

CLOSE , OPEN , SHOWFILE , PRINTFILE

PARn

用途

PARn 用来设定表面编号的第 n 个参数。这个参数值在 “ 表面类型 ” 一章中已详细说明。每种表面模型都使用了 0 到 8 个参数值。

语法结构

PAR1 (surface) = (new_value)

PAR8 (surface) = (new_value)

说明

这个关键字需要两个表达式，一个用来指定表面编号，另一个用来定义一个新的值。求出表达式(surface)的值，取整后得到表面编号。如果表面编号小于 0 或者大于表面数，则这条命令将被略过。否则，求出表达式(new_value)的值，再将之赋值。注意，PARn 是指 8 个不同的命令：PAR1，PAR2，.....，PAR8。

例子

PAR1 (1) = 250

相关关键字

UPDATE

PARAXIAL

用途

用来控制对于近轴光线或者实际光线是否使用光线定位。

语法结构

PARAXIAL ON

PARAXIAL OFF

说明

当前的近轴模式可以通过调用一个函数 PMOD 来建立，如果近轴模式关闭则函数返回 0，否则则返回 1。可用这个特点被来选择是用实际光线还是用近轴光线进行光线追迹。某些计算，如畸变测量，和计算一些第一顺序特性，如有效焦距，需要用近轴光线进行追迹。

例子

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的在于进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁴⁸

```
mode = PMODE( )
```

```
IF mode THEN PRINT "Paraxial mode is on! "
```

```
IF !mode THEN PRINT "Paraxial mode is not on! "
```

```
PARAXIAL ON
```

```
PRINT "Now paraxial mode is on! "
```

```
PRINT "Restoring original mode... "
```

```
IF !mode THEN PARAXIAL OFF
```

PAUSE

用途

暂停程序执行，并且显示状态信息。

语法结构

```
PAUSE
```

```
PAUSE x
```

```
PAUSE "Ready to continue... "
```

```
PAUSE x+SQRT(5)
```

说明

当关闭图表窗口时，这个特性将被自动调用。

PIXEL

用途

在当前的图表窗口中打开一个像素点。

语法结构

```
PIXEL xcoord,ycoord
```

说明

这个特性有利于制作点列图。

POLDEFINE

用途

为后面的 POLTRACE 命令定义输入的偏振状态。

语法结构

POLDEFINE Ex, Ey, Phax, Phay

说明

关键字 POLDEFINE 被用来为后面的偏振光追迹定义输入的偏振状态。POLDEFINE 需要两个规格化的电场强度 Ex 和 Ey, 同样需要 X 和 Y 方向的相位角 (以度为单位)。默认值分别是 0, 1, 0, 0。偏振状态一旦被定义, 它将一直保持到其被改变为止。

例子

POLDEFINE 2.0, 2.0, 45.0, -66.0

相关关键字

POLTRACE

POLTRACE

用途

调用 ZEMAX 偏振光追迹程序去追迹通过当前系统的特殊光线。

语法结构

POLTRACE (Hx), (Hy), (Px), (Py), (wavelength), (vec), (surf)

说明

表达式(Hx)和(Hy)必须在-1 到 1 之间取值, 它们代表归一化的对象坐标。瞳孔坐标由表达式 Px 和 Py 指定, 它们也必须在-1 和 1 之间取值。有关归一化坐标的更多内容请参见“约定和定义”一章中的

版权声明: 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有 (本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的进行学术交流, 如果您有任何意见和看法, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

“归一化的视场和光瞳坐标”部分。表达式 wavelength 的值必须是 1 到最大定义波长数之间的整数。表达式 vec 的值必须是 1 到 4 之间的整数,包括 1 和 4。表达式 surf 的值必须是 1 到最大表面数之间的整数,包括这两个数。

光线输入的偏振状态由关键字 POLDEFINE 定义。

某条光线一旦被追迹,这条光线的偏振数据将被存放在由表达式 vec 指定的向量变量中。例如,如果命令“POLTRACE Hx, Hy, Px, Py, w, 2, n”被执行,那么数据将被存放在 VEC2 中。数据以下面所述的格式储存,这里每行的第一个数字是指数组位置:

- 0: n, 输入数组的数据个数
- 1: 在通过指定表面后光线的强度
- 2: 电场强度 E 的 x 分量, 实部
- 3: 电场强度 E 的 y 分量, 实部
- 4: 电场强度 E 的 z 分量, 实部
- 5: 电场强度 E 的 x 分量, 实部
- 6: 电场强度 E 的 y 分量, 实部
- 7: 电场强度 E 的 z 分量, 实部
- 8: 反射光中 S 偏振光的振幅, 实部
- 9: 反射光中 S 偏振光的振幅, 虚部
- 10: 透射光中 S 偏振光的振幅, 实部
- 11: 透射光中 S 偏振光的振幅, 虚部
- 12: 反射光中 P 偏振光的振幅, 实部
- 13: 反射光中 P 偏振光的振幅, 虚部
- 14: 透射光中 P 偏振光的振幅, 实部
- 15: 透射光中 P 偏振光的振幅, 虚部

如果数组位置 0 的值为 0 , 则将产生一个错误 , 而且偏振数据是无效的。当指定的光线不能被追迹时将会产生这种情况。要摘录扩展错误信息可参见 RAYTRACE 命令。

例子

```
POLDEFINE 0, 1, 0, 0
```

```
POLTRACE 0, 1, 0, 0, PWAV( ), 1, NSUR( )
```

```
PRINT "Transmission of chief ray at primary wavelength is ",  
vec1(1)
```

相关关键字

```
POLDEFINE , RAYTRACE
```

```
PRINT
```

用途

PRINT 用来输出常量文本和变量数据到屏幕上或者文件里 , 这依赖于由关键字 OUTPUT 决定的当前状态。

语法结构

```
PRINT
```

```
PRINT x
```

```
PRINT "The value of x is "; x
```

```
PRINT "x= ", x, "x+y = ", x+y
```

说明

单独的 PRINT 语句将打印一个空行。带有一列文本自变量和表达式的 PRINT 语句将打印每个文本字符串 (在双引号范围内) 和每个表达式的数值。PRINT 语句使用由 FORMAT 指定的数值输出格式。如果列表的最后一项后面跟着一个逗号 , 那么 PRINT 将不用回车号结束这一行。

例子

版权申明 : 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有 (本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的进行学术性质的交流, 如果您有任何意见和看法, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

```
x = 3
```

```
PRINT X equals ;x
```

```
PRINTFILE
```

用途

打印一个文本文件。

语法结构

```
PRINTFILE filename "
```

```
PRINTFILE NAME$
```

说明

filename 必须是有效范围内的文件名或者包含有效文件名的字符串变量。这个文件必须是 ASCII 码文件(由 ZPL 中的 OUTPUT 和 PRINT 语句创建的),同时必须在当前目录下。即使没有执行 CLOSE 语句,PRINTFILE 也将关闭这个文件。

例子

```
OUTPUT test.txt "
```

```
PRINT Print this to the printer. "
```

```
PRINTFILE test.txt "
```

相关关键字

```
OPEN , OUTPUT , CLOSE , PRINT , PRINTFILE
```

```
PRINTWINDOW
```

用途

打印任何打开的图表和文本窗口。

语法结构

```
PRINTWINDOW winnum
```

版权申明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者),“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的进行学术交流,如果您有任何意见和看法,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

说明

winnum 的值必须是一个整数或者一个值为整数的表达式。整数 winnum 对应于要打印的窗口编号。当窗口被打开时,ZEMAX 将从 1 开始给它们顺序编号。任何关闭的窗口将从窗口列表中删除,但余下的窗口不再重新编号。在一个窗口被关闭后打开的窗口将使用可用的最小的窗口编号。

例子

PRINTWINDOW 5

PWAV

用途

设定主波长。

语法结构

PWAV(wavelength number)

说明

求出表达式(wavelength number)的值,把主波长设成这个指定的编号。

例子

PWAV 1

相关函数

WAVL , WWGT , PWAV

RADI

用途

RADI 用来设定表面的曲率半径。

语法结构

RADI(surface) = (new_value)

版权声明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者)。“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的进行学术交流,如果您有任何意见和看法,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

说明

这个关键字需要两个关键字，一个用来指定表面编号，另一个用来定义一个新的值。求出表达式(surface)的值，取整后用来得到表面编号。如果这个表面编号的值小于 0 或者大于最大的表面数，那么这条命令将被略过。否则，求出表达式(new_value)的值，并将之赋值。

例子

RADI i+m = -1200

相关关键字

UPDATE

RAYTRACE

用途

调用 ZEMAX 光线追迹程序去追迹通过当前系统的一条特殊光线。

语法结构

RAYTRACE (Hx), (Hy), (Px), (Py), (wavelength)

说明

表达式 Hx 和 Hy 必须在-1 到 1 之间取值，象征归一化的对象坐标。瞳孔坐标由表达式 Px 和 Py 指定，它们也必须在-1 到 1 之间取值。有关归一化坐标的更多内容请参见“约定和定义”一章中的“归一化的视场和光瞳坐标”部分。表达式 wavelength 为随意值，默认为主波长，但如果提供了这个值，这个值必须是 1 到最大定义波长数之间的整数。

一旦光线被追迹，那么这条光线的截取坐标和方向余弦将使用 ZPL 函数 RAYX, RAYY, RAYZ, RAYL, RAYM 和 RAYN 来确定。如果在光线追迹过程中产生了错误，那么函数 RAYE (对于 RAY 错误) 将返回一个不是 0 的数。如果 RAYE 为负数，则表明在编号为返回值的绝对值的表面上产生了全反射。如果 RAYE 大于 0，则光线

没有到达该表面。虽然对 RAYE 的检查是随意的，但如果 RAYE 不为 0，则函数 RAYX、RAYY、.....，将返回无效的数值。函数 RANX、RANY 和 RANZ 将返回截止表面的法线的方向余弦，函数 OPDC 返回光线的光程差。函数 RAYV 返回光线被挡住的表面的编号，如果光线没被挡住则返回 0。返回的那些通过渐晕面的表面值不可能是准确的。

例子

```
PRINT "Tracing the marginal ray at primary wavelength! "
```

```
n = NSUR( )
```

```
RAYTRACE 0, 0, 0, 1
```

```
y = RAYY(n)
```

```
PRINT "The ray intercept is ",y
```

```
PRINT "Tracing the chief ray at maximum wavelength! "
```

```
RAYTRACE 0, 1, 0, 0, NWAV( )
```

```
y = RAYY(n)
```

```
PRINT "The ray intercept is ",y
```

相关关键字

RAYTRACEX

RAYTRACEX

用途

调用 ZEMAX 光线追迹程序去追迹通过当前系统的一条特殊光线。

语法结构

```
RAYTRACEX (x), (y), (z), (l), (m), (n), (surf), (wavelength)
```

说明

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁵⁶

表达式 x, y, z, l, m, n 定义了输入光线的位置和方向余弦。表达式 `surface` 必须是 0 和表面数减 1 之间的整数，包括这两个数。表达式 `wavelength` 为随意值，默认为主波长，但如果提供了这个值，则必须是 1 到最大定义波长数之间的整数。

➡ 如果对象有无穷大的厚度，且参数 `surf` 为 0，那么虽然光线仍在对象介质空间中定义，但是要假定输入坐标与第一面有关，而不是对象面。否则 ZEMAX 将不作任何更改而使用指定的坐标

一旦光线被追迹，那么这条光线的截取坐标和方向余弦将使用 ZPL 函数 `RAYX`，`RAYY`，`RAYZ`，`RAYL`，`RAYM` 和 `RAYN` 来确定。注意只有那些从第 “`surf`” 面后的表面中得到才有效。

一旦光线被追迹，那么这条光线的截取坐标和方向余弦将使用 ZPL 函数 `RAYX`，`RAYY`，`RAYZ`，`RAYL`，`RAYM` 和 `RAYN` 来确定。如果在光线追迹过程中产生了错误，那么函数 `RAYE`（对于 `RAY` 错误）将返回一个不是 0 的数。如果 `RAYE` 为负数，则表明在编号为返回值的绝对值的表面上产生了全反射。如果 `RAYE` 大于 0，则光线没有到达该表面。虽然对 `RAYE` 的检查是随意的，但如果 `RAYE` 不为 0，则函数 `RAYX`、`RAYY`、.....，将返回无效的数值。函数 `RANX`、`RANY` 和 `RANZ` 将返回截止表面的法线的方向余弦，函数 `OPDC` 返回光线的光程差。函数 `RAYV` 返回光线被挡住的表面的编号，如果光线没被挡住则返回 0。返回的那些通过渐晕面的表面值不可能是准确的。

例子

```
n = NSUR()
```

```
RAYTRACEX 0, 1, 0, 0, 0, 1, 0, NWAV()
```

```
y = RAYY(n)
```

```
PRINT 'The intercept is ',y
```

相关关键字

RAYTRACE

READ

用途

从一个由 OPEN 命令读入的现已打开的 ASCII 码数值文本文件中读取数据。

语法结构

READ x

READ x, y

READ x, y, z, a, b, c, q

说明

ASCII 码文件必须已经打开，详细内容参见关键字 OPEN。每个 READ 命令从文件中读入一行数据。这行的第一个位置的数据存放到列出的第一个变量中，第二个位置的数据存放到列出的第二个变量中，依次类推。因此，在 READ 语句中列出变量的个数应该与文本文件中的列数相配对。文件中的数值数据应由空格隔开。数据的类型是任意的，并在内部将其提升为双精度。在一行中最多可以读入 2000 个字母。列出的变量必须是有效的 ZPL 变量名。

通常在所有的数据都被读入后关闭文件。参见函数 EOFF。

例子

```
PRINT Reading the double-column file TEST.DAT! "
```

```
OPEN TEST.DAT
```

```
READ x1, y1
```

```
READ x2, y2
```

```
READ x3, y3
```

```
CLOSE
```

READSTRING

用途

从一个由 OPEN 命令读入的现已打开的 ASCII 码文本文件中读取数据。

语法结构

```
READSTRING A$
```

说明

这个 ASCII 码文件必须已经打开，详细内容参见关键字 OPEN。每个 READ 命令从文件中读入一行。读入的整行数据被存放在列出的变量中。这个列出的变量必须是一个有效的 ZPL 字符串变量名，这个变量名不必已被说明。通常在所有的数据都被读入后关闭文件。参见函数 EOFF。

例子

```
PRINT Reading the contents of file TEST.DAT! "
```

```
OPEN TEST.DAT
```

```
READSTRING A$
```

```
PRINT A$
```

```
CLOSE
```

```
REM.!
```

说明

REM 用来指出这行的其余部分是注释。

语法结构

```
REM text
```

```
! text
```

说明

这个惊叹号也用来象征注释。如果命令 REM 和符号 “！”在一行

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的旨在进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！

的开端,在任何空格、制表符或者其他字母之前,那么它们仅被认为是注释指示器。任何其他用法将在运行时产生语法错误。

例子

REM any text can be placed after the REM command.

! any text can also be placed

! after the exclamation symbol.

RETURN

参见 GOSUB

SAVELENS

用途

保存当前的镜头文件。

语法结构

SAVELENS [filename]

SAVELENS NEW\$

说明

SAVELENS 将把当前镜头文件保存到磁盘中。当前内存中的镜头名也将被改变。如果文件名已存在,那么镜头数据将被保存在当前文件名中。

例子

SAVELENS

SAVELENS NEWCOPY.ZMX "

SAVELENS NEW\$

相关关键字

版权声明: 本文由“光学在线”收集及整理,所有版权归文章原始作者所有(本站均注明出处和作者),“光学在线”是一个介绍光学及相关科学的专业网站,其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展,不为任何赢利目的,所以传播此文的目的进行学术交流,如果您有任何意见和看法,请与我们联系(info@photics.net),我们会在最短的时间内给您答复并采取相应的补救措施!

LOADLENS

SDIA

用途

SDIA 用来设定表面的半口径。

语法结构

SDIA(surface) = (new_value)

说明

这个关键字需要两个表达式，一个用来指定表面编号，另一个用来定义一个新的值。求出表达式(surface)的值，取整后得到表面编号。如果表面编号小于 0 或者大于表面数，则这条命令将被略过。否则，求出表达式(new_value)的值，再将之赋值。

如果 new_value 的值是一个正数，则不管原来半口径的状态是什么，其状态都将变成“固定”。如果 new_value 的值为负数，那么其状态将设成“自动”，它将再下次调用“UPDATE”时被更新。

例子

SDIA(5) = SDIA(4)

相关关键字

UPDATE

SETAIM

用途

设定光线定位功能的状态。

语法结构

SETAIM state

说明

这个关键字需要一个数值表达式，它必须是 0 到 2 之间的整数。

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！

表达式 state 是一个代码, 0 代表无光线定位(关闭), 1 代表近轴光线参考, 2 代表实际光线参考。

例子

SETAIM 1

SETCONFIG

用途

为多重结构 (变焦) 系统设置当前结构。

语法结构

SETCONFIG (confignumber)

说明

这个关键字需要一个数值表达式, 它必须是 1 到最大结构数之间的整数。求出表达式的值, 取整后得到结构编号。

例子

SETCONFIG 4

相关函数

CONF, NCON

SETNSCPARAMETER

用途

设置非连续元件系统编辑界面中的任何对象的参数值。

语法结构

SETNSCPARAMETER surface, object, parameter, value

说明

这个关键字需要 3 个数值表达式, 它们的值是一个整数, 分别指

定非连续元件系统的表面编号、对象编号和参数编号。第四个自变量是为指定参数而定义的新值。

例子

SETNSCPARAMETER 4, 2, 15, newp15value

相关函数

NPOS , NPAR

相关关键字

SETNSCPOSITION

SETNSCPOSITION

用途

设置非连续元件系统编辑界面中任何对象位置的 x 坐标,y 坐标,z 坐标,或者 x 倾斜,y 倾斜, z 倾斜。

语法结构

SETNSCPOSITION surface, object, code, value

说明

这个关键字需要 3 个数值表达式, 它们的值是一个整数, 分别指定非连续元件系统的表面编号、对象编号和一个代码。这个代码为 1 到 6, 分别代表 x 坐标,y 坐标,z 坐标,或者 x 倾斜,y 倾斜, z 倾斜。第四个自变量是为指定位置而定义的新值。

例子

SETNSCPOSITION 4, 2, 2, newyvalue

相关函数

NPOS , NPAR

相关关键字

SETNSCPARAMETER

版权声明: 本文由“光学在线”收集及整理, 所有版权归文章原始作者所有(本站均注明出处和作者), “光学在线”是一个介绍光学及相关科学的专业网站, 其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展, 不为任何赢利目的, 所以传播此文的目的进行学术交流, 如果您有任何意见和看法, 请与我们联系 (info@photics.net), 我们会在最短的时间内给您答复并采取相应的补救措施!

SETOPERAND

用途

将评价函数编辑界面中任意一行或者一列设成某一数值。

语法结构

SETOPERAND row, col, value

说明

这个关键字需要 2 个数值表达式，它们的值是整数，分别指定评价函数编辑界面中的行和列。整数 col 是 1 则代表操作数类型；2 代表 int1；3 代表 int2；4-7 代表 Hx-Py；8 代表目标值；9 代表权重。注意实际值和贡献百分数不能直接设定，只能计算得出。

例子

SETOPERAND 1, 8, tarvalue

相关函数

MFCN, OPER

SETTEXTSIZE

用途

改变由命令 GTEXT 写下的字符的大小。

语法结构

SETTEXTSIZE xsize, ysize

说明

这个两个由代表字符大小的自变量是指图表屏幕宽度的分数。例如，默认的文本尺寸是 70，40，这意味着每个字符大小是图表屏幕宽度的 1/70，以及屏幕高度的 1/40。不带自变量则使文本尺寸恢复回默认值。

例子

！使文本尺寸是默认值的两倍

SETTEXTSIZE 35, 20

！使文本尺寸恢复回默认值

SETTEXTSIZE

SETTITLE

用途

设置镜头的标题，它通常将在所有的图中显示出来。

语法结构

SETTITLE A\$

SETTITLE Here is the lens title ”

SETUNITS

用途

设置当前的镜头长度单位

语法结构

SETUNITS code

说明

这个关键字需要一个数值表达式，它的值必须是 0 到 3 之间的一个整数。这个代码是 0 则代表毫米,1 代表厘米,2 代表英寸,3 代表米。这个功能不能以任何方式缩放或者转变镜头数据，它只能改变镜头的指示数据是如何表达的。

例子

SETUNITS 0

SETVAR

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！

用途

改变用来优化的变量的状态。

语法结构

SETVAR (surface_expression), VARCODE,
(status_expression)

说明

自变量 surface_expression 的值必须是 0 到最大表面数之间的整数，否则将产生一个错误。surface_expression 决定了哪个表面将被设成变量。VARCODE 必须是下列 ASCII 记忆码之一：

R 代表曲率半径

T 代表厚度

C 代表二次曲线系数

Pn 代表参数 n

En 代表特殊数据值 n

如果 status_expression 的值是 0，那么这个表面的变量状态被设成与优化无关。否则，这个值被设成可变的。

例子

SETVAR j+3, R, 1

SETVAR 5, P6, 0

SETVAR surfk+2, E06, status

SETVECSIZE

用途

改变数组 VEC1、VEC2、VEC3、VEC4 的最大容量。

语法结构

SETVECSIZE (expression)

说明

自变量 expression 的值必须是 1 到 100,000 之间的整数。四个向量组变量总是有相同的容量。

SETVIG

用途

设置镜头的渐晕因子。

语法结构

SETVIG

说明

参见“约定和定义”一章中的关于渐晕因子的描述。

SHOWFILE

说明

使用 ZEMAX 文件阅览器在屏幕上显示文本文件。

语法结构

SHOWFILE filename 'saveflag

SHOWFILE NAME\$ saveflag

说明

filename 必须是有效范围内的文件名，或者包含有效文件名的字符串变量。这个文件必须是一个 ASCII 码文件（是在 ZPL 中由 OUTPUT 和 PRINT 语句创建的），且必须在当前目录下。这个文件一旦被显示，它将和其他文本文件一样可以上下滚动以及被打印出来。可以滚动和打印数据的能力是用 OUTPUT 和 SHOWFILE 语句代替单一的 PRINT 语句的主要优点。即使没有执行 CLOSE 语句，

SHOWFILE 也能关闭文件。如果自变量 saveflag 为 0 或者被省略, 那么当窗口被关闭时这个也将被删除。如果 saveflag 是一个非 0 值, 那么即使在窗口关闭之后文件仍将保留下来。

例子

```
OUTPUT test.txt "
```

```
PRINT Print this to a file. "
```

```
SHOWFILE test.txt "
```

相关关键字

OPEN , OUTPUT , CLOSE , PRINT , PRINTFILE

SOLVETYPE

用途

改变一个给定表面和值的求解状态。这里仅支持有一部分求解类型, 如要有关设置其他求解类型的信息请与 FSI 联系。

语法结构

```
SOLVETYPE (surf_expression), CODE, (arg1), (arg2), (arg3)
```

说明

自变量 surf_expression 的值必须是 0 到最大表面数之间的一个整数, 否则将产生错误。自变量 CODE 必须是下表列出的一个 ASCII 记忆码。表达式 arg1、arg2、arg3 的值分别是指在“求解”一章中规定的第一个、第二个、第三个求解参数。

关键字 SOLVETYPE 中的代码

求解类型	代码
曲率拾取 (Curvature Pickup)	CP
厚度拾取(Thickness Pickup)	TP
厚度边缘光高(Thickness Marginal Ray Height)	TM
玻璃补偿(Glass Offset)	GO
玻璃拾取(Glass Pickup)	GP
锥形拾取(Conic Pickup)	KP
参数 1-8 拾取(Parameter 1-8 Pickup)	P1, P2 ... P8

例子

！下一行将在表面 7 上加一个玻璃拾取求解类型，从第 5 表面拾取

SOLVETYPE 7, GP, 5

！加一个比例因子为-1 的厚度拾取求解类型

SOLVETYPE 7, TP, 5, -1

STOPSURF

用途

STOPSURF 用来设置用编号确定的当前光栏面的位置。

语法结构

STOPSURF (surface)

说明

这个关键字需要一个表达式，它的值必须是 1 到表面数减 1 之间

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的旨在进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁶⁹

的一个整数。

例子

STOPSURF n+2

相关关键字

UPDATE

SUB

参见 GOSUB

SURFTYPE

用途

改变表面类型。

语法结构

SURFTYPE (surf_expression), CODE

说明

自变量 surf_expression 的值必须是 0 到表面数之间的一个整数。自变量 CODE 是一个代表表面类型的整数值。这个整数代码可以在运行时由函数 SCOD 确定。命令 SURFTYPE 不能用来设置光栅漂移、光栅相位或者用户自定义表面。

例子

SURFTYPE j+1, SCOD(STANDARD)

B\$ = "EVENASPH"

SURFTYPE 5, SCOD(B\$)

THIC

用途

THIC 用来设置表面的厚度。

语法结构

THIC (surface) = (new_value)

说明

这个关键字需要两个表达式，一个用来指定表面编号，另一个用来定义一个新的值。求出表达式(surface)的值，取整后得到表面编号。如果这个表面编号小于 0 或者大于表面数，那么这条命令被略过。否则求出表达式(new_value)的值，再将之赋值。

例子

THIC i+m = THIC(i+n)-1

相关关键字

UPDATE

TIMER

用途

重新设置内部时钟。这一特点与 ZPL 函数 ETIM()联合使用可用来测量自最后一个 TIMER 命令以来所经过的时间。

语法结构

TIMER

说明

TIMER 和 ETIM()主要用来测试 ZPL 编译器和各种程序结构的运行效率。

例子

i = 0

TIMER

LABEL 1

```
x = RAND(1000)
```

```
i = i+1
```

```
IF i<10000 THEN GOTO 1
```

```
FORMAT .1
```

```
RPINT Elapsed time: ;ETIM( ), Seconds "
```

UPDATE

用途

更新瞳孔位置、折射率数据、近轴常量、半口径、最大视场归一化数据和求解。如果自从上一次更新以来指示数据（如半径和厚度）已被改变，那么必须在光线追迹和系统评价之前使用关键字 UPDATE。如果 UPDATE 命令后面跟着其他单词，如“ALL”，那么所有打开的窗口都同时被更新。

语法结构

```
UPDATE
```

```
UPDATE ALL
```

例子

```
THIC 5 = THIC(5)-1
```

```
UPDATE
```

```
VEC1 , VEC2 , VEC3 , VEC4
```

用途

这些关键字用来设置向量组变量 VEC1、VEC2、VEC3 和 VEC4。每个向量组可以储存一个由双精度浮点数组成的数组。

语法结构

```
VEC1 (array_subscript) = (new_value)
```

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的在于进行学术性质的交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁷²

VEC2 (array_subscript) = (new_value)

说明

VEC1...4 用来把一些用户数据存放到一个数组中。表达式 array_subscript 可以是其值被取整的任意一个表达式。这个作为结果的整数表达式必须在 0 和当前最大数组容量之间,数组的起始容量是 100,否则将被告示一个错误。ZPL 功能 VEC1...4 可被用来记录数据。为了改变默认容量使之不同于 100,请使用关键字 SETVECSIZE。

例子

i = 0

LABEL 1

i = i+1

VEC1(i) = i

IF i<10 THEN GOTO 1

j = 0

LABEL 2

j = j+1

VEC2(j) = VEC1(j)*VEC1(j)

IF j<10 THEN GOTO 2

i = 0

LABEL 3

i = i+1

PRINT 'x= ', VEC1(i), 'x*x= ', VEC2(i)

IF i<10 THEN GOTO 3

PRINT

PRINT 'ALL done! "

WAVL , WWGT

用途

WAVL 和 WWGT 是用来改变波长的值和权重。

语法结构

WAVL (wavelength number) = (new_value)

WWGT (wavelength) = (new_value)

说明

这两个关键字需要两个表达式，一个用来指定波长编号，另一个用来定义一个新的值。求出表达式(wavelength number)的值，取整后得到波长编号。如果这个波长编号小于 1 或者大于波长数，那么这条命令将被略过。否则，求出(new_value)的值，再将之赋值。

例子

WAVL 1 = 0.486

WAVL 2 = 0.587

WWGT 1 = 1.0

WWGT 2 = 0.65

相关函数

WAVL , WWGT

范例 1

假设你需要一个程序取打印出每个定义视场角的主光线在像平面的截止坐标。通常这比你想象的更加必须，例如，ZEMAX 的点列图都是对应于主光线的坐标画出来的，因此拿截止坐标来充当这些数据是完全可以的。这对于横向的光线像差也是同样，它们也是以主光线为参考。

版权申明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁷⁴

学习 ZPL 的最好的方法就是学习前面已写好的程序。仔细阅读下面的程序，并尽量理解程序的逻辑结构。

```
nfield = NFLD()
maxfield = MAXF()
n = NSUR()
FOR i=1,nfield,1
    hx = FLDX(i)/maxfield
    hy = FLDY(i)/maxfield
    PRINT  Field number ;i
    RAYRACE hx,hy,0,0,PWAV()
    PRINT  x-field angle: ;FLDX(i), y-field angle: ;FLDY(i)
    PRINT  x-chief ray   : ;RAYX(i), y-chief ray : ;RAYY(i)
    PRINT
NEXT
PRINT  'ALL done! '
```

程序的第一行调用了函数 NFLD()，它将返回定义的视场数，并将它赋值给变量 “ numfield ”，第二行调用了函数 MAX()，它将返回最大的半视场角（以度表示），并将它存放在变量 “ maxfield ” 中。然后通过调用函数 NSUR()将表面编号存放在变量 “ n ” 中。

然后定义了一个以 i 作为视场位置的计数器的 FOR 循环，这个计数器从 1 开始，最大值为 nfield，每个循环的增量为 1。定义 “ hx ” 和 “ hy ” 的两个程序行使用了函数 FLDX()和 FLDY()，它们将返回当前的编号为 i 的视场位置的 x-和 y-角度。然后用关键字 RAYTRACE 进行光线追迹。注意主光线通过瞳孔的中心，这就是为什么两个瞳孔坐标都为 0 的原因。函数 PWAV()返回主波长的编号，这通常是针对主光线的。

各样的 PRINT 语句将像面上的主光线坐标输出到屏幕上。如果你喜欢，你可以试着将程序输入到 ZPL 末尾的文件中（如 CHIEF.ZPL），然后在 ZEMAX 内运行它来进行测试。当主光线为参考点时，得到的坐标与那些在点列图中列出的是一样的。

范例 2

这是一个估计当前光学系统的均方根斑点尺寸（轴上）的 ZPL 程序例子。程序追迹许多通过系统的随机光线，并记录下这些光线相对于主波长的主光线的偏离量。当前波长的权重被应用来估计均方根斑点尺寸。

```
PRINT Primary wavelength is number ;
```

```
FORMAT .0
```

```
PRINT PWAV(),
```

```
FORMAT .4
```

```
PRINT 'which is ',WAVL(PWAV()), 'microns. '
```

```
PRINT Estimating RMS spot size for each wavelength. "
```

！追迹多少条随机光线来估算？

```
n = 100
```

！初始化时钟

```
TIMER
```

！储存表面数以备后用

```
ns = NSUR()
```

！从波长 1 开始

```
weightsum = 0
```

```
wwrms = 0
```

```
FOR w = 1,NWAV(),1
```

版权声明：本文由“光学在线”收集及整理，所有版权归文章原始作者所有（本站均注明出处和作者），“光学在线”是一个介绍光学及相关科学的专业网站，其宗旨在于推广光学和相关学科以及光电产业在中国的传播和发展，不为任何赢利目的，所以传播此文的目的进行学术交流，如果您有任何意见和看法，请与我们联系（info@photics.net），我们会在最短的时间内给您答复并采取相应的补救措施！⁴⁷⁶

```
rms = 0
```

```
FOR i = 1,n,1
```

```
    hx = 0
```

```
    hy = 0
```

```
    angle = 6.283185*RAND(1)
```

```
    ! SQRT 可在瞳孔处产生均匀分布
```

```
    radius = SQRT(RAND(1))
```

```
    px = radius*COSI(angle)
```

```
    py = radius*SINE(angle)
```

```
    RAYTRACE hx,hy,px,py,w
```

```
    x = RAYX(ns)
```

```
    y = RAYY(ns)
```

```
    rms = rms+(x*x)+(y*y)
```

```
NEXT
```

```
rms = SQRT(rms/n)
```

```
wwrms = wwrms+(WWGT(w)*rms)
```

```
weightsum = weightsum+WWGT(w)
```

```
FORMAT .4
```

```
PRINT RMS spot size for ',WAVL(w)
```

```
FORMAT .6
```

```
PRINT 'is ',rms
```

```
NEXT
```

```
wwrms = wwrms/weightsum
```

```
PRINT 'Wavelength weighted rms is ',wwrms
```

```
FORMAT .2
```

```
t = ETIM()
```

```
PRINT 'Elapsed time was ',t, "seconds. "
```

注意开始两个 PRINT 语句中后缀逗号的使用。它们允许从开始三个 PRINT 语句打印出来的数据显示在同一行中。中间的 FORMAT 语句改变了要打印的数值输出的格式,即使它们可能是同一行的。字符“!”用来表明一个注释语句,当程序运行时这些语句将被略过。