

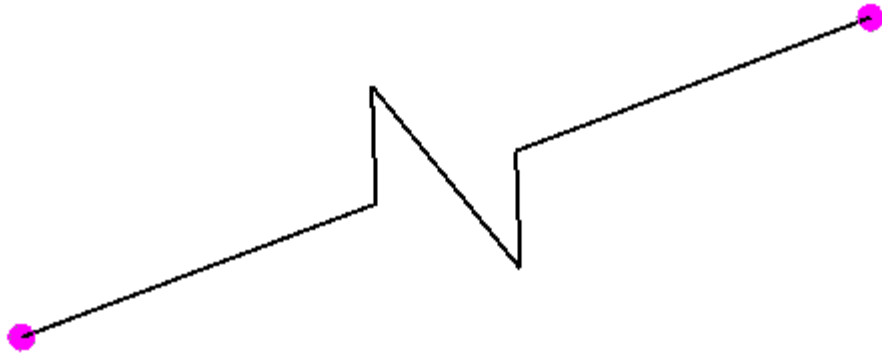
GDL高级培训教程

（剖析实例部件）

- 对已有GDL部件实例的剖析，是学习GDL最快捷的途径。通过解读剖析内部参数设定、算法思路、GDL语句等，能得到很好的借鉴，从而快速提高GDL实战水平。
- 以下几个例子是本人编写的典型GDL部件，有2D符号和3D部件，有的编写年代比较早，算法和语句不一定合理，但应该还有借鉴作用。

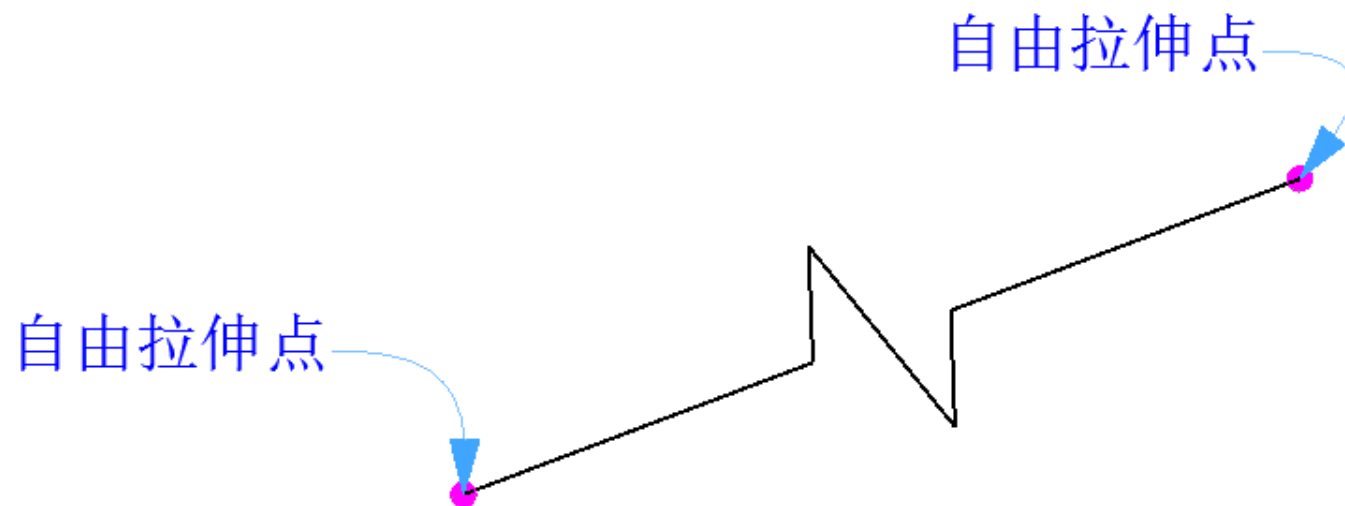
- 尝试做一个剖断线（2D）
- 做一个智能的标高符号（2D）
- 自动布置的餐桌 (2D)
- 向3D进阶——室外踏步(3D)
- 非常复杂的螺旋欧式栏杆(3D)

PART.1 尝试做一个剖断线(2D)



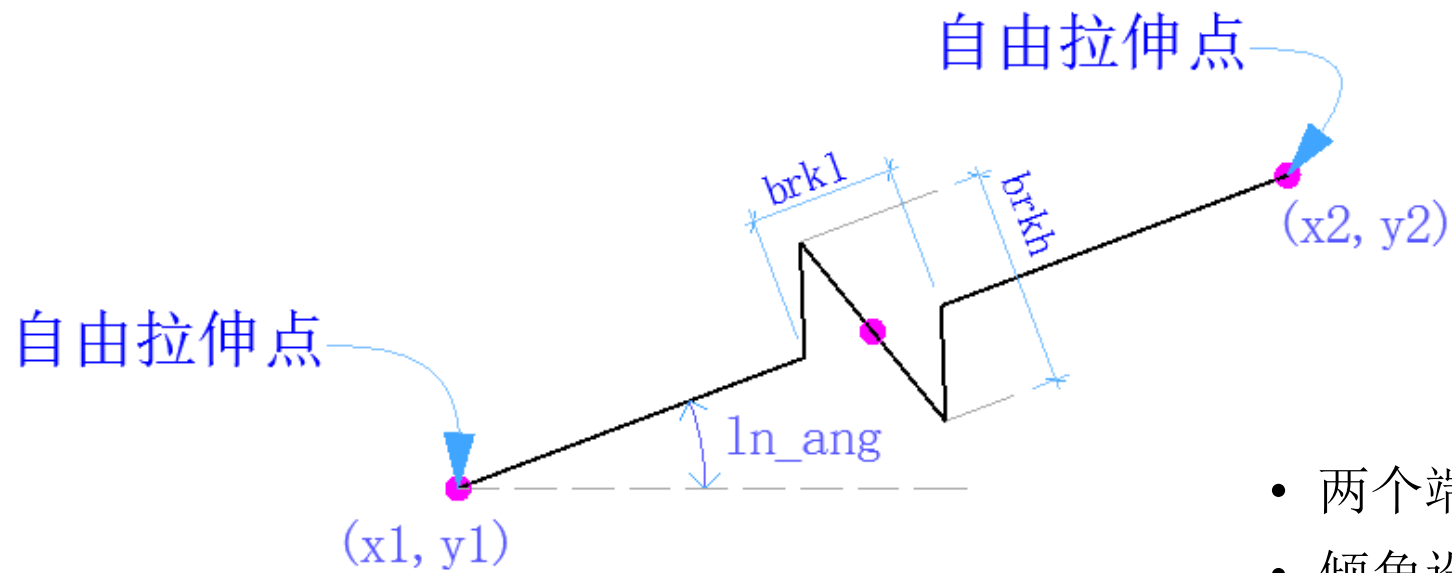
- 希望能自由拉伸两个端点。
- 整个符号能随着端点的位置变化而变化，而无需旋转。
- 剖断符号大小可以调整。

2. 我们希望这个部件是怎么样子的？



- 希望能自由拉伸两个端点。
- 整个符号能随着端点的位置变化而变化，而无需旋转。
- 剖断符号大小可以调整。

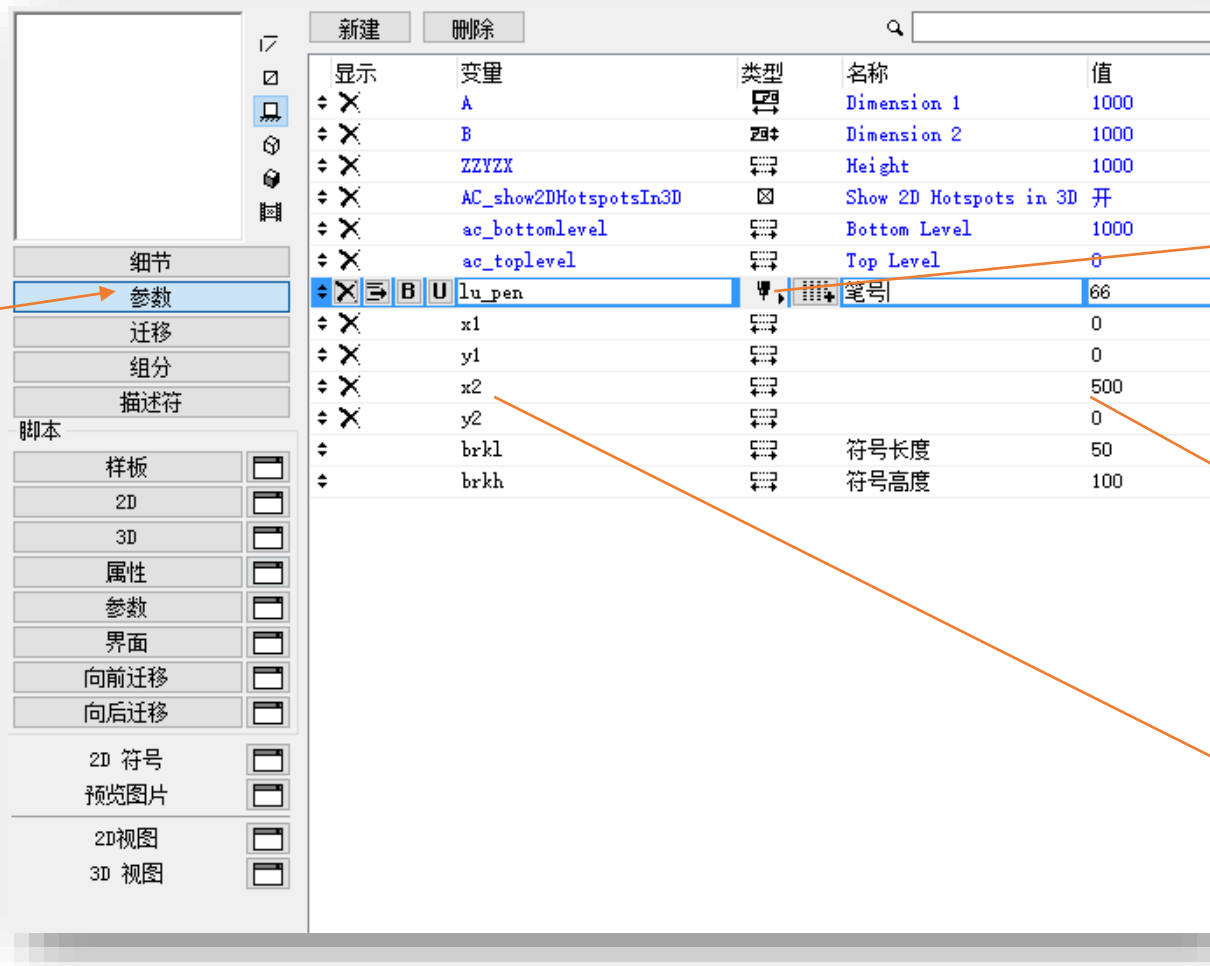
3. 分析这个剖断线可能有哪些参数



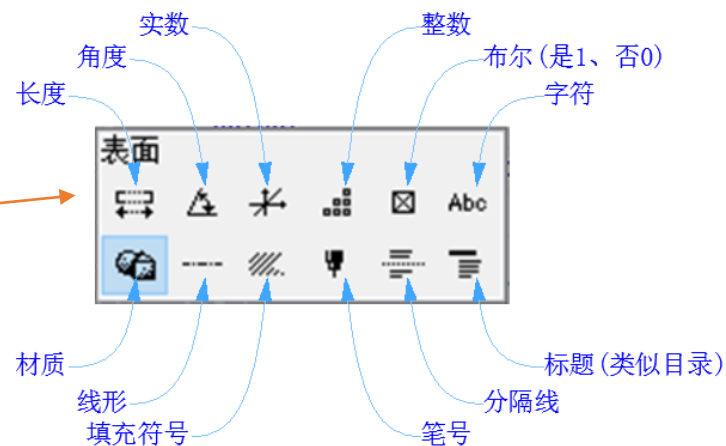
- 两个端点: $(X1,Y1)$, $(X2,Y2)$ 。
- 倾角设为: ln_ang 。
- 符号高: $brkh$, 符号长: $brkl$ 。
- 另外, 线的笔号设为: lu_pen

4.现在开始.....

点击
这里



- 10种参数类型，2种参数框效果控制



- 参数初始值

- 两个端点: (X1,Y1) , (X2,Y2)。
- 符号高: brkh , 符号长: brkl。
- 线的笔号设为: lu_pen

- 倾角需要计算: ln_ang。

!!! 以下编辑两端点的X1, Y1, X2, Y2的坐标位置

Unid = 1 !! 设编辑热点id的初始值

!!! X1

HOTSPOT2 0, y1, unid, x1, 1+128 : unid = unid +1 !基点
HOTSPOT2 x1, y1, unid, x1, 2 : unid = unid +1 !移动点
HOTSPOT2 -1, y1, unid, x1, 3 : unid = unid +1 !向量

!!! y1

HOTSPOT2 x1, 0, unid, y1, 1+128 : unid = unid +1
HOTSPOT2 x1, y1, unid, y1, 2 : unid = unid +1
HOTSPOT2 x1, -1, unid, y1, 3 : unid = unid +1

!!!以上这X1,Y1的语句组合, 才能有自由移动的可编辑热点

!!! X2

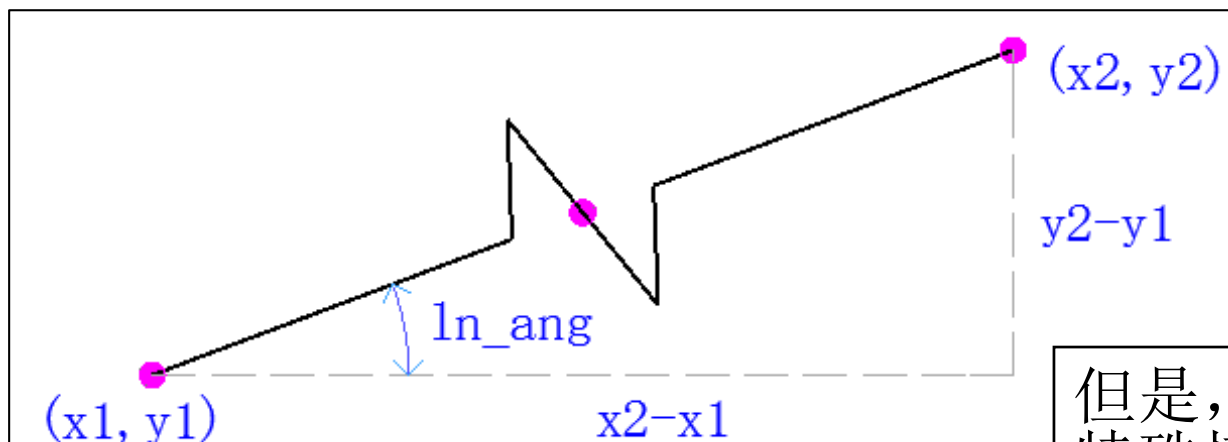
HOTSPOT2 0, y2, unid, x2, 1+128 : unid = unid +1
HOTSPOT2 x2, y2, unid, x2, 2 : unid = unid +1
HOTSPOT2 -1, y2, unid, x2, 3 : unid = unid +1

!!! y2

HOTSPOT2 x2, 0, unid, y2, 1+128 : unid = unid +1
HOTSPOT2 x2, y2, unid, y2, 2 : unid = unid +1
HOTSPOT2 x2, -1, unid, y2, 3 : unid = unid +1

- 控制点(x1,y1)自由移动的标准语句

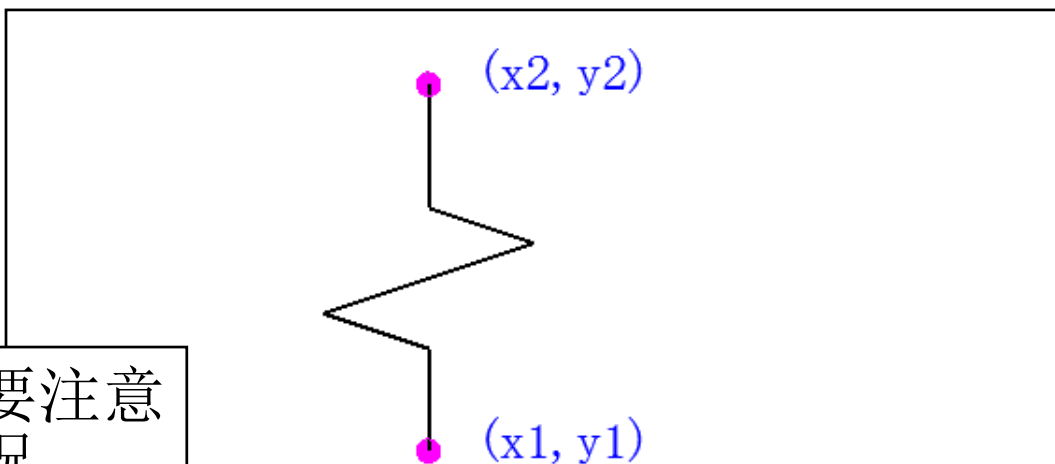
- 控制点(x2,y2)自由移动的标准语句格式



用反正切函数求出倾角：

$$\ln_ang = \text{atn} ((y2 - y1) / (x2 - x1))$$

但是，要注意
特殊情况... ..



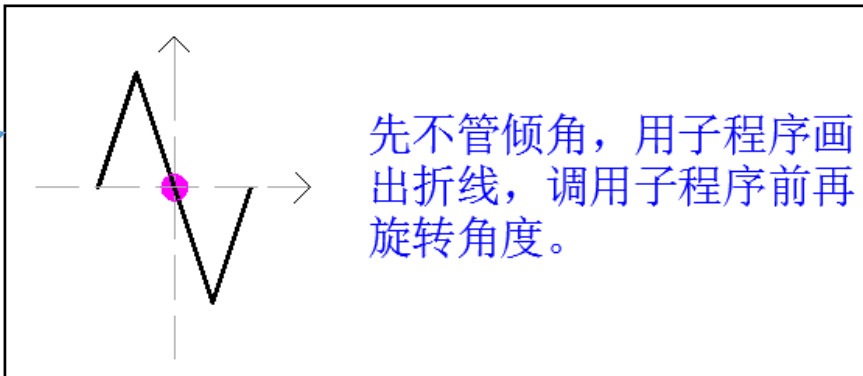
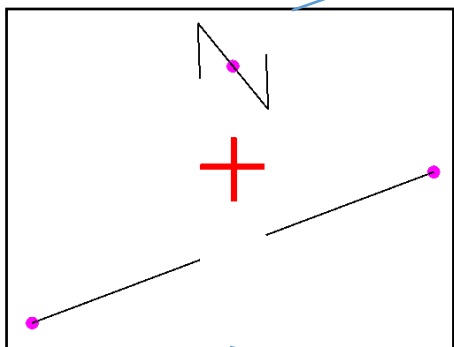
$(x2 - x1) = 0.0$ 时, $(y2 - y1) / (x2 - x1)$ 无意义
因此需要排除这个情况, 直接认定 $\ln_ang = 90$ 度

!实际上，我们可以设一个非常小的数，当 $(x2 - x1)$ 的绝对值小于这个数的时候，认定为0

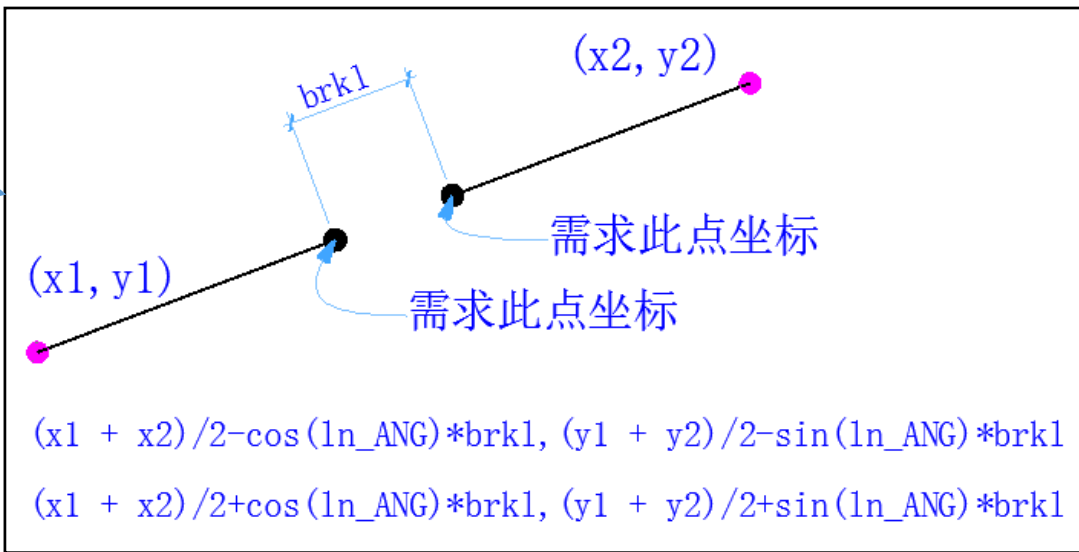
$\text{eps} = 0.0001$

IF $X2 < X1$ and $\text{ABS}(x2 - X1) \leq \text{eps}$ then $\ln_ANG = 90.0$ ELSE $\ln_ANG = \text{atn}((y2 - y1) / (x2 - x1))$

可以分解为： 直线+中间折线 +直线：



先不管倾角，用子程序画出折线，调用子程序前再旋转角度。



$(x1 + x2) / 2 - \cos(\ln_ANG) * brkl$, $(y1 + y2) / 2 - \sin(\ln_ANG) * brkl$
 $(x1 + x2) / 2 + \cos(\ln_ANG) * brkl$, $(y1 + y2) / 2 + \sin(\ln_ANG) * brkl$

```
100: !!折线 子程序
POLY2 4, 1,
      -brkl, 0,
      -brkl / 2, brkh,
      brkl / 2, -brkh,
      brkl, 0
RETURN
```

```
ADD2 (x1 + x2) / 2, (y1 + y2) / 2 !坐标移到线的中间点
ROT2 ln_ANG
      GOSUB 100
DEL 1
DEL 1
```

```
PEN lu_pen

LINE2 x1, y1, (x1 + x2) / 2 - cos(ln_ANG) * brkl,
      (y1 + y2) / 2 - sin(ln_ANG) * brkl

LINE2 (x1 + x2) / 2 + cos(ln_ANG) * brkl,
      (y1 + y2) / 2 + sin(ln_ANG) * brkl, x2, y2
```

完整的代码:

```
!! LU_BreakLine.gsm
!! FOR ArchiCAD 15
!! By 无忧 2014.6.20
```

```
unid = 1
eps = 0.0001
!!! 以下编辑两端点的X1, Y1, X2, Y2的坐标位置
HOTSPOT2 0, y1, unid, x1, 1+128 : unid = unid +1 !基点
HOTSPOT2 x1, y1, unid, x1, 2 : unid = unid +1 !移动点
HOTSPOT2 -1, y1, unid, x1, 3 : unid = unid +1 !向量

HOTSPOT2 x1, 0, unid, y1, 1+128 : unid = unid +1
HOTSPOT2 x1, y1, unid, y1, 2 : unid = unid +1
HOTSPOT2 x1, -1, unid, y1, 3 : unid = unid +1

HOTSPOT2 0, y2, unid, x2, 1+128 : unid = unid +1
HOTSPOT2 x2, y2, unid, x2, 2 : unid = unid +1
HOTSPOT2 -1, y2, unid, x2, 3 : unid = unid +1

HOTSPOT2 x2, 0, unid, y2, 1+128 : unid = unid +1
HOTSPOT2 x2, y2, unid, y2, 2 : unid = unid +1
HOTSPOT2 x2, -1, unid, y2, 3 : unid = unid +1
```

```
!!!! TAN的反三角函数,求线的角度,此处考虑X2-X1约等于0的情况,。
IF X2 > X1 and ABS (x2 - X1) <= eps then ln_ANG = 270.0 ELSE ln_ANG = atn ( ( y2 -y1 ) / (x2 - x1) )
```

```
HOTSPOT2 (x1 + x2) / 2, ( y1 + y2 ) / 2 !!加中点热点
```

```
!!! 以下画线
PEN lu_pen
LINE2 x1, y1, (x1 + x2) / 2 -cos(ln_ANG) * brkl, ( y1 + y2 ) / 2 - sin(ln_ANG) * brkl
LINE2 (x1 + x2) / 2 + cos(ln_ANG) * brkl, ( y1 + y2 ) / 2 + sin(ln_ANG) * brkl, x2, y2

ADD2 (x1 + x2) / 2, ( y1 + y2 ) / 2 !坐标移到线的中间点
ROT2 ln_ANG
GOSUB 100

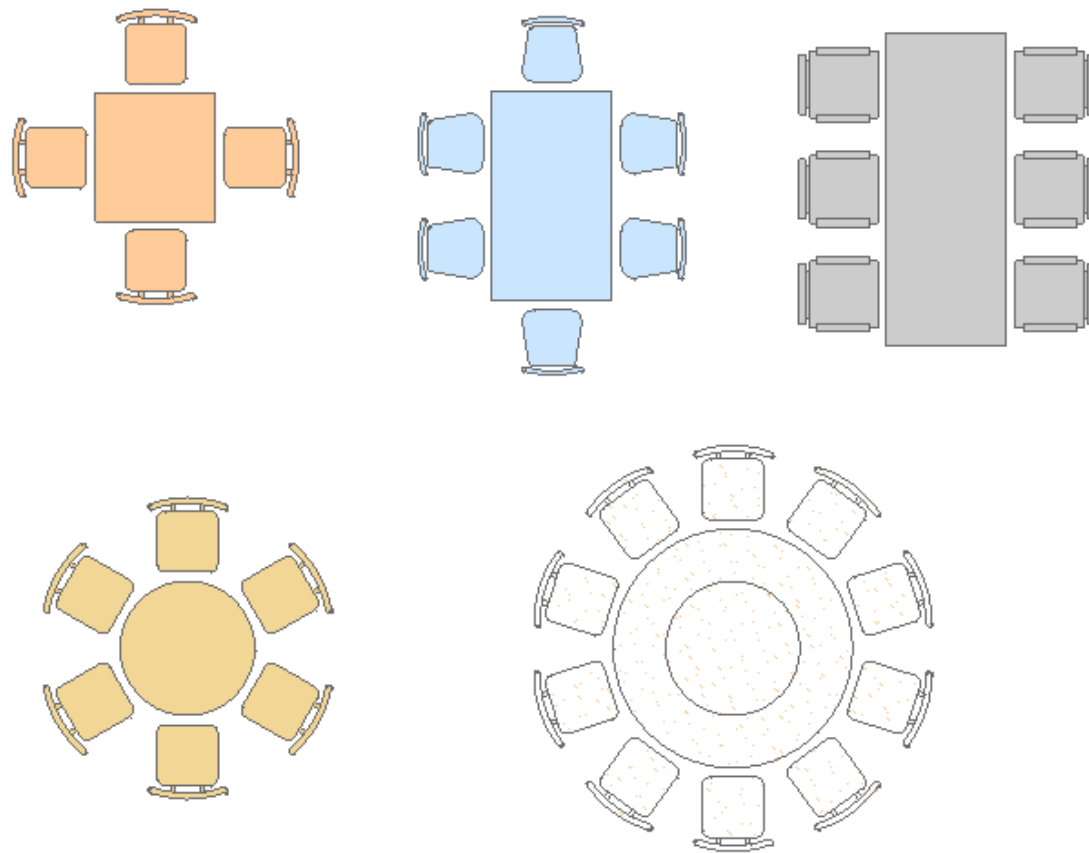
DEL 1
DEL 1
END !!!主程序结束
!!!!=====

100: !!折线 子程序
POLY2 4, 1,
      -brkl, 0,
      -brkl / 2, brkh,
      brkl / 2, -brkh,
      brkl, 0
RETURN
```

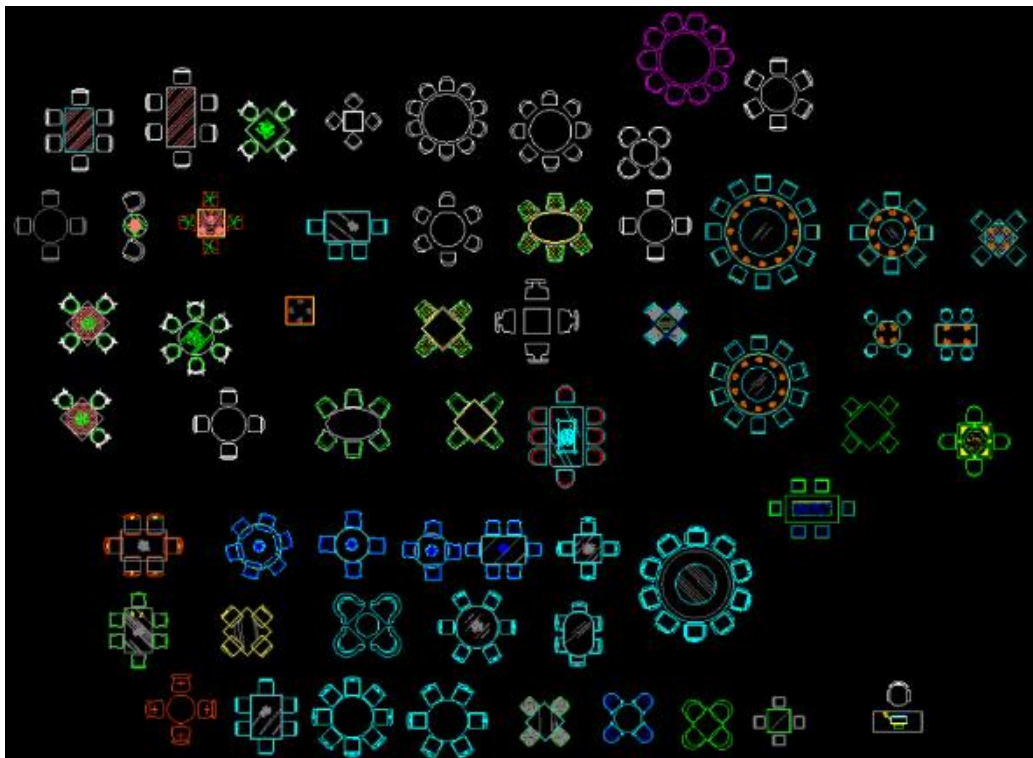
小结：

- 用到了hotspot2 热点编辑。
- 用到了三角函数：反正切函数atn求角，cos 和sin求斜边长。
- 用到了子程序，用GOSUB调用子程序。
- 需要注意三角函数中特殊角的处理。

PART.2 做一个自动布置的餐桌 (2D)

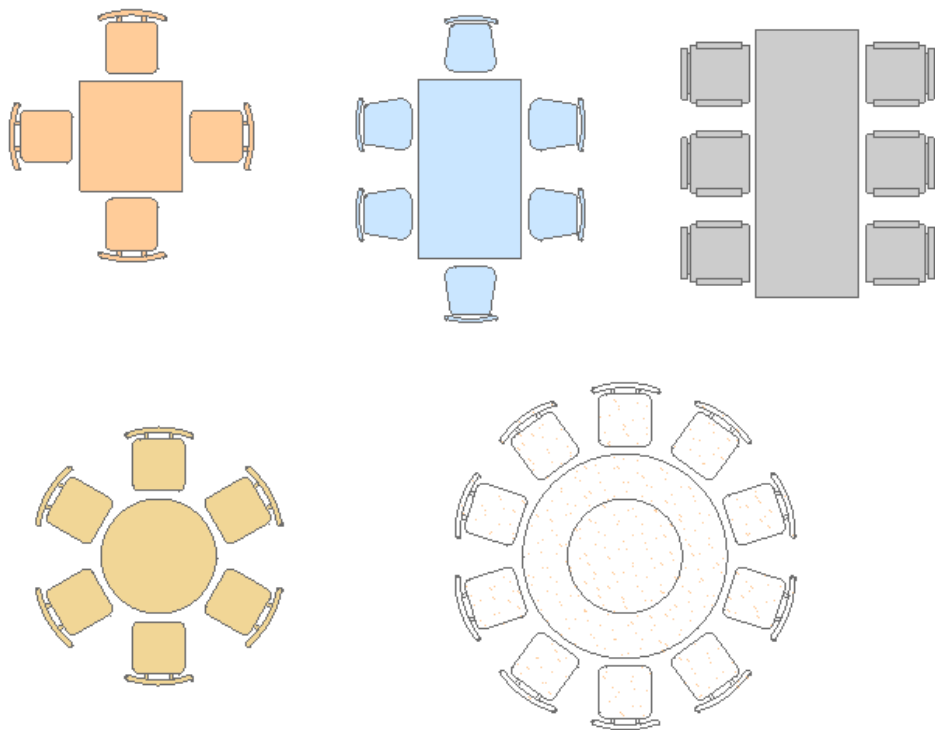


以前的图库是这样的:



- 我们都见过、或收藏过左图这样的CAD图库。
- 圆桌、方桌、不同大小桌子、不同椅子风格..... 都是一个个图块。
- 现在我们想要用一个智能GDL部件，来囊括这些各种变化的餐桌图库。

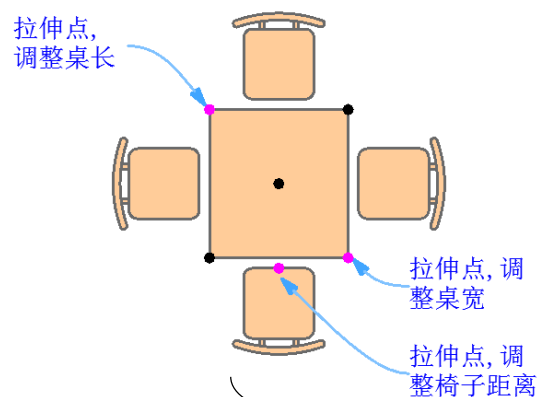
现在我们要什么样的图库？



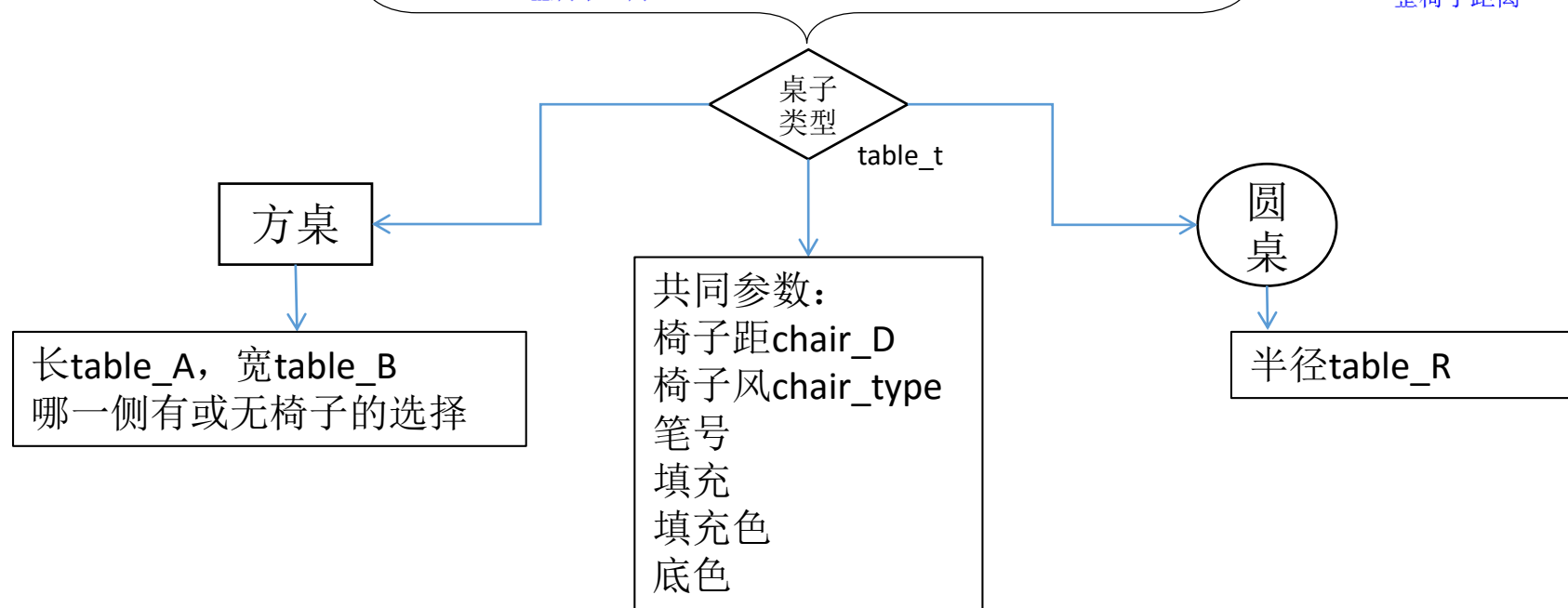
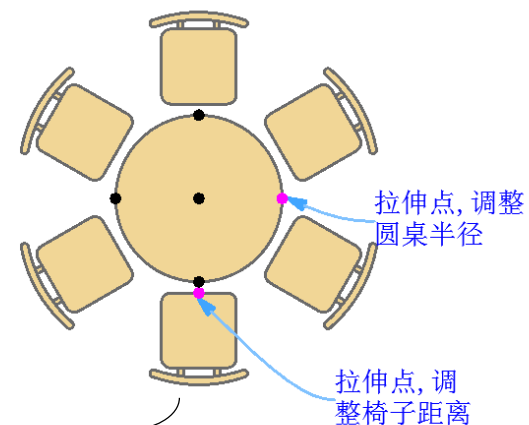
1. 能自由拉伸方桌子的长和宽，圆桌的半径。
2. 拉伸后能自动计算桌子边长，匹配布置椅子。
3. 能更换椅子风格。
3. 能设笔号颜色，能设填充颜色，丰富图面效果
4. 。 。 。 。 。

分析需要哪些参数

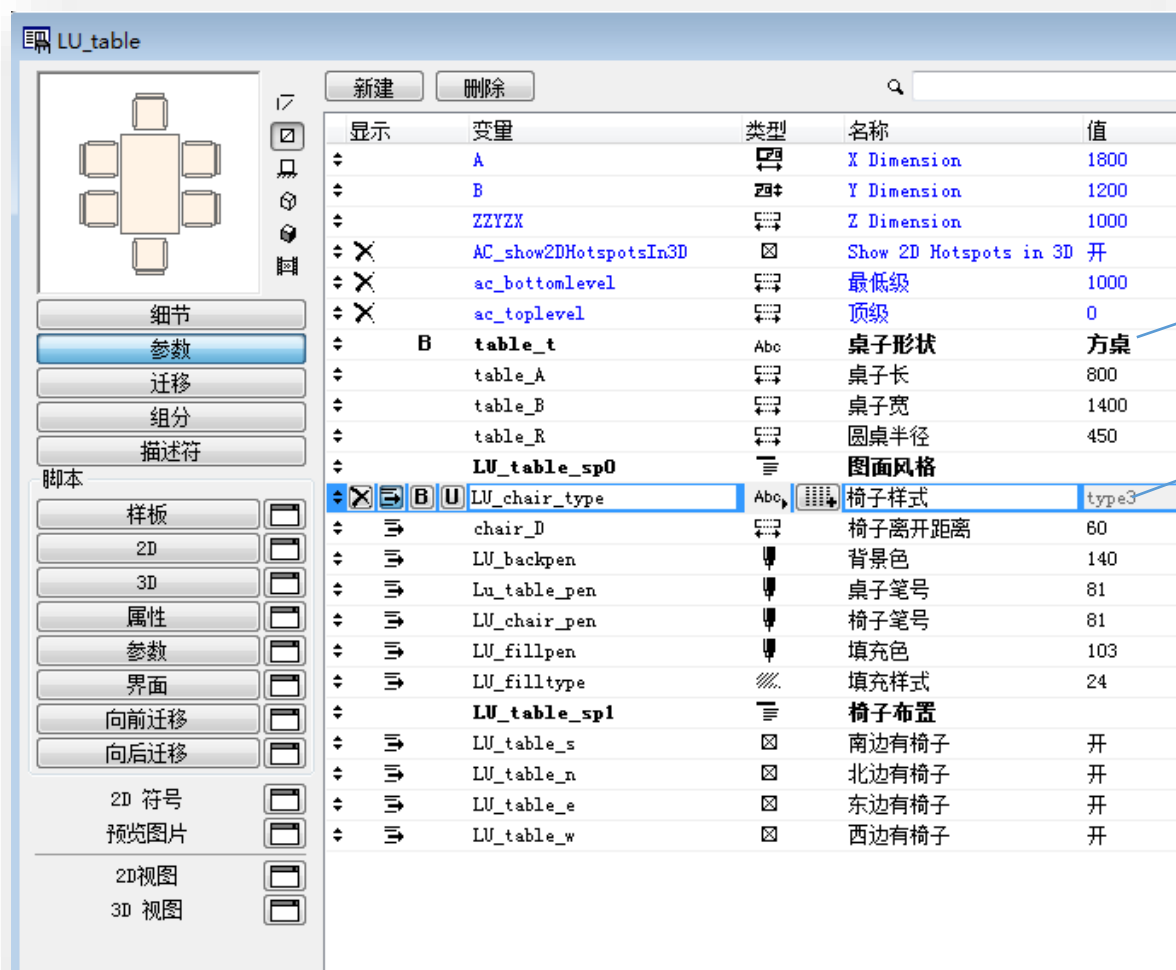
1. 对于矩形桌:



2. 对于圆桌:



完整的参数设置:



选择桌子类型：方桌、圆桌

选择椅子类型：方桌、圆桌

图面风格设置

对于方桌，椅子布置选择


```

if table_t="方桌" then
    !!!以下是动态拉伸桌子的长度: table_A
    HOTSPOT2 0, 0, unid, table_A, 1+128 : unid = unid +1
    HOTSPOT2 -1, 0, unid, table_A, 3      : unid = unid +1
    HOTSPOT2 table_A, 0, unid,table_A, 2 : unid = unid +1

    !!!以下是动态拉伸桌子的宽度: table_A
    HOTSPOT2 0, 0, unid, table_B, 1+128 : unid = unid +1
    HOTSPOT2 -1, 0, unid, table_B, 3      : unid = unid +1
    HOTSPOT2 0,table_B, unid,table_B, 2   : unid = unid +1

    !!!以下是动态拉伸椅子与桌子的距离: chair_D
    HOTSPOT2 0.35 +(table_A - n*0.7 )/2, 0, unid, chair_D, 1+128 : unid = unid +1
    HOTSPOT2 -1, 0, unid, chair_D, 3   : unid = unid +1
    HOTSPOT2 0.35 +(table_A - n*0.7 )/2, -chair_D , unid ,chair_D, 2 : unid = unid +1

else

    !!!以下是动态拉伸桌子的半径: r
    HOTSPOT2 0, 0, unid, table_R, 1+128 : unid = unid +1
    HOTSPOT2 -1, 0, unid, table_R, 3      : unid = unid +1
    HOTSPOT2 table_R, 0, unid,table_R, 2 : unid = unid +1

    !!!以下是动态拉伸椅子与桌子的距离: chair_D
    HOTSPOT2 0, -table_R, unid, chair_D, 1+128 : unid = unid +1
    HOTSPOT2 -1, 0, unid, chair_D, 3           : unid = unid +1
    HOTSPOT2 0, -table_R-chair_D , unid,chair_D, 2 : unid = unid +1

endif

```

用 if ... then

...

else

...

endif

区分圆桌与方桌，并
分别处理热点拉伸

还要解决的问题是布置多少个椅子？

$n = \text{int} (\text{table_A} / 0.70 + 0.5)$!方桌长边布置n个椅子
$m = \text{int} (\text{table_B} / 0.70 + 0.5)$!方桌宽边布置m个椅子
$s = \text{int} (2 * \text{PI} * \text{table_R} / 0.7 + 0.5)$!圆桌边布置s个椅子

1. 这里设座位宽度为0.70米，这个数也可以改为变量。
2. Int 函数是取整数，而+0.5是“四舍五入”的一个技巧，当时的GDL无四舍五入函数。
3. $2 * \text{PI} * \text{table_R}$ 是求圆桌的周长。

不同椅子的子程序

```
if LU_chair_type = "type1" then ch_tp = 100
```

```
if LU_chair_type = "type2" then ch_tp = 200
```

```
if LU_chair_type = "type3" then ch_tp = 300
```

分别指向不同椅子的子程序

100: !!!以下是椅子，椅子样式1

PEN LU_chair_pen

SET FILL LU_filltype

POLY2_B {2} 7, 3, LU_fillpen, LU_backpen,

0, 0, 0,

-0.1122619903377, -0.405504, 1,

-0.08307140786825, -0.405504, 1,

-0.08307140786825, -0.4515783590453, 1,

-0.0, 0.2939904000001, 900,

0, -2.248743628779, 4001,

-0.1122619903377, -0.4477446462227, 1,

-0.1122619903377, -0.405504, -1

POLY2_B {2} 13, 3, LU_fillpen, LU_backpen,

0, 0, 0,

0.24330240, -0.456192, 1,

0.2433024, -0.4359168, 900,

0, 180, 4001,

0.2433024, -0.4156416, 1,

-0.0, 0.29399040, 900,

0, -37.84928883206, 4001,

-0.2433024, -0.4156416, 1,

-0.2433024, -0.4359168, 900,

0, 180, 4001,

-0.24330240, -0.456192, 1,

-0.0, 0.25344, 900,

0, 37.84928883211, 4001,

0.24330240, -0.456192, -1

POLY2_B {2} 7, 3, LU_fillpen, LU_backpen,

0, 0, 0,

0.1125040922132, -0.405504, 1,

0.08477320943716, -0.405504, 1,

0.08477320943716, -0.4513867771602, 1,

0.0, 0.29399040, 900,

0, 2.136647893087, 4001,

0.1125040922132, -0.4477079635582, 1,

0.1125040922132, -0.405504, -1

POLY2_B {2} 17, 3, LU_fillpen, LU_backpen,

0, 0, 0,

0.202752, -0.3548160000001, 1,

0.1520640000001, -0.354816, 900,

0, -90, 4001,

0.152064, -0.405504, 1,

-0.152064, -0.405504, 1,

-0.152064, -0.354816, 900,

0, -90, 4001,

-0.202752, -0.354816, 1,

-0.202752, -0.050688, 1,

-0.1520640, -0.050688, 900,

0, -90, 4001,

-0.152064, 0, 1,

0.152064, 0, 1,

0.1520640, -0.050688, 900,

0, -90, 4001,

0.202752, -0.050688, 1,

0.202752, -0.3548160, -1

return

当时是用笨办法，没有使用**FRAGMENT2**，而是先画好三种椅子，再转换成GDL脚本作为子程序插进来。

因此，学习GDL不要怕做不好，要勇于尝试，哪怕是笨拙的办法也是提高的过程。

下个例子将用到**FRAGMENT2**

布置椅子:

用嵌套的if ... then ... else .. Endif 语句, 区分方圆, 且在方桌 东\南\西\北 区分开关

用嵌套的for...next 和add2, rot2 语句, 重复变化坐标, 进行布置。

```
if table_t ="方桌" then
  if LU_table_s = 1 then
    !!以下是南的椅子
    add2 0.35 +(table_A - n*0.7 )/2, -chair_D
    for i=1 to n
      gosub ch_tp
      add2 0.7 ,0
    next i
    del n+1
  endif
  ....
  ....
  ....
else !圆桌布置椅子
  for i = 1 to s
    rot2 360 /s
    add2 0, -table_R-chair_D
    gosub ch_tp
    del 1
  next i
  del top
endif
```

调用椅子
子程序

如果“南风家”为是, 则布置椅子

重复, 东\西\北

调用椅子
子程序

布置圆桌的椅子

画桌面： 用嵌套的if ... then ... else .. Endif 语句，区分方圆

```
PEN Lu_table_pen
SET FILL LU_filltype

if table_t="方桌" then !方桌
    POLY2_B {2} 5, 3, LU_fillpen, LU_backpen,
                0, 0, 0, 0,
                table_A, table_b, 1,
                table_A, 0, 1,
                0, 0, 1,
                0, table_b, 1,
                table_A, table_b, -1

    HOTSPOT2 table_A, table_b
    HOTSPOT2 table_A, 0
    HOTSPOT2 0, 0
    HOTSPOT2 0, table_b
    HOTSPOT2 table_A/2, table_B/2

else !!!圆桌
    POLY2_B {2} 7, 3, LU_fillpen, LU_backpen,
                0, 0, 0, 0,
                -table_R, 0, 1,
                0, 0, 900,
                0, 180, 4001,
                table_R, 0, 1,
                0, 0, 900,
                0, 180, 4001,
                -table_R, 0, -1

    HOTSPOT2 -table_R, 0
    HOTSPOT2 table_R, 0
    HOTSPOT2 0, table_R
    HOTSPOT2 0, -table_R
    HOTSPOT2 0, 0

    if table_R >= 0.650 then CIRCLE2 0, 0, table_R-0.35

endif

endif
```

画方桌

加热点，便于选择物件

画圆桌

加热点，便于选择物件

当桌子比较大时，中间加转盘 ☺

此外，还要在参数脚本处加限制条件：

The screenshot displays the GraphISOFT software interface, specifically the 'LU_table' parameter script editor and the '自定义设置' (Custom Settings) panel.

LU_table Script Editor:

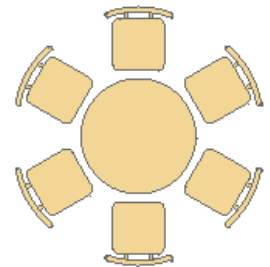
- Values "table_t" "方桌", "圆桌"** (Values "table_t" "Square Table", "Round Table")
- Values "LU_chair_type" "type1", "type2", "type3"** (Values "LU_chair_type" "type1", "type2", "type3")
- if table_t = "方桌" then hideparameter "table_R"** (if table_t = "Square Table" then hideparameter "table_R")
- if table_t = "圆桌" then** (if table_t = "Round Table" then)
- hideparameter "table_A"** (hideparameter "table_A")
- hideparameter "table_B"** (hideparameter "table_B")
- endif** (endif)
- Values "chair_D" range (-0.150 , 0.150)** (Values "chair_D" range (-0.150 , 0.150))
- Values "table_A" range (0.4 ,)** (Values "table_A" range (0.4 ,))
- Values "table_B" range (0.4 ,)** (Values "table_B" range (0.4 ,))
- Values "table_R" range (0.18 , 1.90)** (Values "table_R" range (0.18 , 1.90))

自定义设置 (Custom Settings) Panel:

- 桌子形状 (Table Shape):** 方桌 (Square Table) is selected. The value is 800.
- 桌子长 (Table Length):** 800
- 桌子宽 (Table Width):** 856
- 图面风格 (Drawing Style):** type1 is selected. The value is type1.
- 椅子样式 (Chair Style):** type1 is selected. The value is type1.
- 椅子离开距离 (Chair Distance):** 60
- 背景色 (Background Color):** 视窗背景 (Window Background) -1
- 桌子笔号 (Table Line Number):** 0.20 mm 61

Annotations:

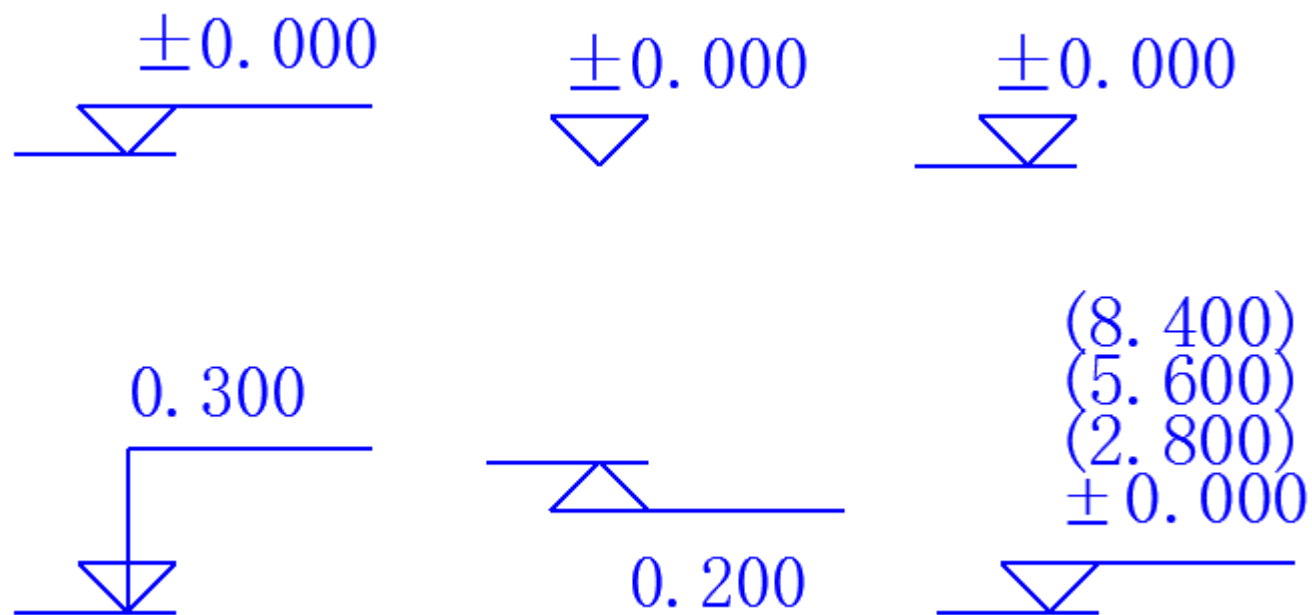
- Red arrow from **hideparameter "table_A"** to **隐藏无关的差数选项** (Hide irrelevant difference options).
- Red arrow from **Values "chair_D" range (-0.150 , 0.150)** to **限制椅子离开桌子的距离-150~ 150** (Restrict chair distance from table -150~ 150).
- Red arrow from **Values "table_A" range (0.4 ,)** to **限制桌子的最小值** (Restrict table minimum value).

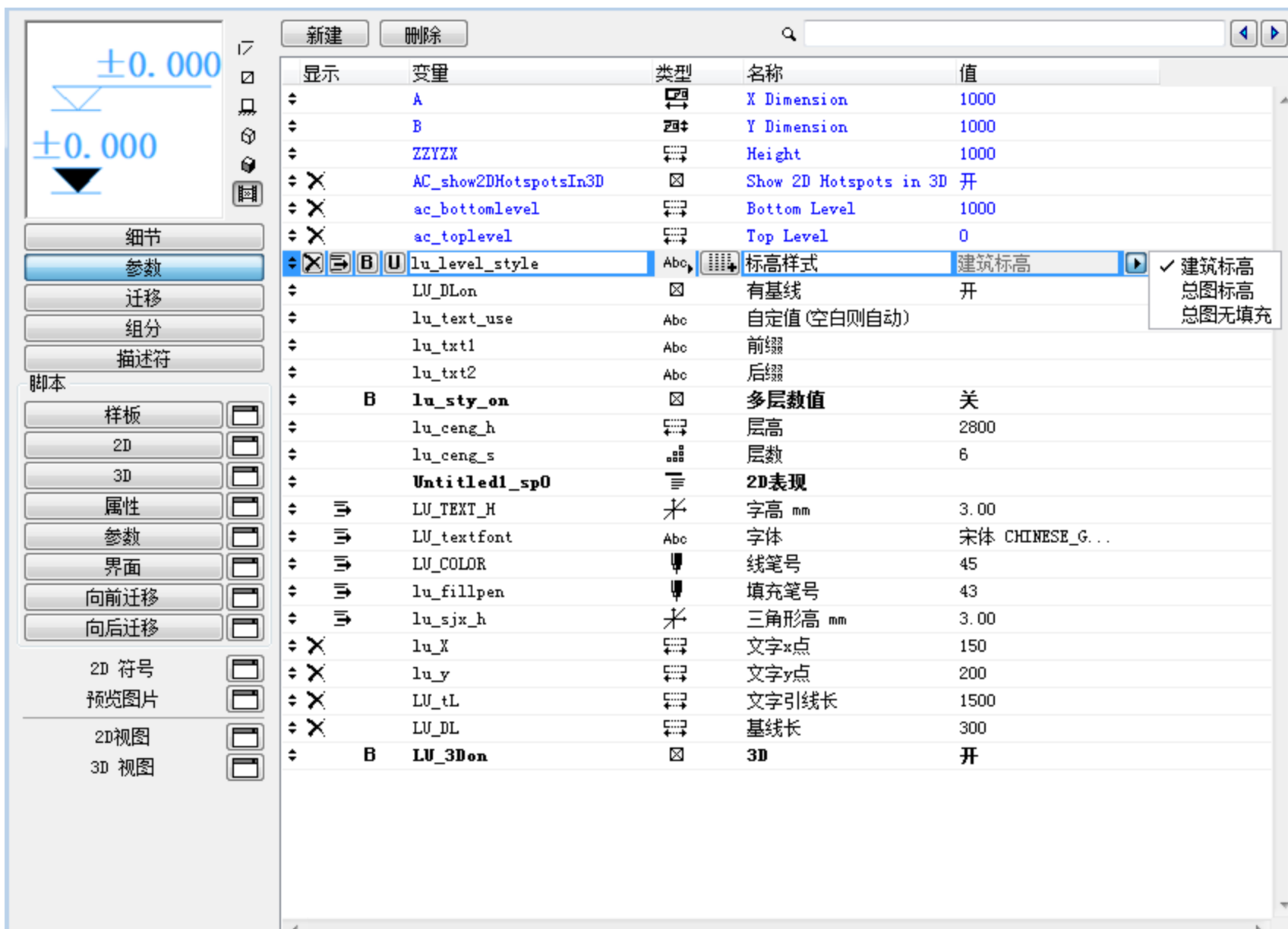


小结:

- 记得这是AC8.1的时候做的GDL，当时hotspot2 热点编辑是第一个出现的版本。
- 注意排列个数的问题，长度/间距，四舍五入后取整数就是个数。
- If ...then...else...endif 语句的嵌套使用。
- For ... next 语句里有add2, rot2的等坐标变换的，要注意del 的个数。
- 若到fragment2则程序更加简洁，但笨方法也能解决问题。

PART.2 做一个智能标高





样板脚本

```
! Name   : lu_level_new.gsm
! Date    : 2013年5月30日
! Version : 15
! Written by 无忧

IF GLOB_CONTEXT = 2 or GLOB_CONTEXT = 3 THEN
    lu_bg = STR ( "%m", GLOB_hstory_ELEV + GLOB_ELEVATION)
else
    lu_bg = STR ( "%m", SYMB_POS_Y)
endif
if LU_3Don and GLOB_CONTEXT = 4 then lu_bg = STR ( "%m", SYMB_POS_Y)

IF lu_bg = "0.000" THEN lu_bg = "±" + lu_bg
lu_bg = lu_txt1 + lu_bg + lu_txt2      !!!前缀和后缀
IF lu_text_use <> "" THEN lu_bg = lu_text_use !!!自定义数值

unid = 1
mirr_x = 1 : mirr_y = 1
IF lu_level_style = "总图标高" THEN fill_on = 3 else fill_on = 1

HOTSPOT2 0,0
```

GLOB_CONTEXT 显示外观的关键字

1- 图库部件编辑器, 2- 平面图, 3-3D 视图, 4- 剖面/立面图, 5- 设置对话框, 6- 列表, 7- 详图, 8- 布图, 22- 平面图的反馈模式, 23- 3D 视图的反馈模式, 24- 剖面/立面图的反馈模式, 28- 布图的反馈模式, 43- 从3D 视图生成为一个运算符, 44- 从剖面/立面图中生成为一个运算符, 46- 从列表中生成为一个运算符

→ 楼层高度的全局变量，标高的全局变量

→ 换算为m单位。

```

PEN LU_COLOR
HOTSPOT2 lu_X, 0, unid, lu_Y, 1+128 : unid = unid + 1
HOTSPOT2 lu_X, -1, unid, lu_Y, 3 : unid = unid + 1
HOTSPOT2 lu_X, lu_Y, unid, lu_Y, 2 : unid = unid + 1

HOTSPOT2 0, lu_Y, unid, lu_X, 1+128 : unid = unid + 1
HOTSPOT2 -1, lu_Y, unid, lu_X, 3 : unid = unid + 1
HOTSPOT2 lu_X, lu_Y, unid, lu_X, 2 : unid = unid + 1

if lu_X >= 0 then mirr_x = 1 else mirr_x = -1
if lu_y >= 0 then mirr_y = 1 else mirr_y = -1

IF mirr_x = 1 and mirr_y = 1 THEN txt_con = 7
IF mirr_x = 1 and mirr_y = -1 THEN txt_con = 1
IF mirr_x = -1 and mirr_y = 1 THEN txt_con = 9
IF mirr_x = -1 and mirr_y = -1 THEN txt_con = 3
IF lu_level_style = "总图标高" or lu_level_style = "总图无填充" THEN txt_con = 8

DEFINE STYLE "AC_STYLE_1" LU_textfont,
    LU_TEXT_H, txt_con, 0

IF lu_level_style = "建筑标高" THEN
MUL2 mirr_x * ( GLOB_SCALE * 0.01 ) , mirr_y * ( GLOB_SCALE * 0.01 )
!!LU_tL !! 文字底线
HOTSPOT2 0, 0.3, unid, LU_tL, 1+128 : unid = unid + 1
HOTSPOT2 -1, LU_tL, unid, LU_tL, 3 : unid = unid + 1
HOTSPOT2 LU_tL, 0.3, unid, LU_tL, 2 : unid = unid + 1
del 1
ENDIF

parameters : LU_X = ( GLOB_SCALE * 0.01 ) * LU_X
parameters : LU_Y = LU_Y * ( GLOB_SCALE * 0.01 )

GOSUB 99 !!!文字

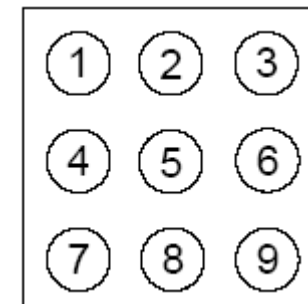
MUL2 mirr_x * GLOB_SCALE * lu_sjx_h / 300 , mirr_y * GLOB_SCALE * lu_sjx_h / 300
!!! mirr_x x方向镜像系数, mirr_y y方向镜像系数
!ADD2 0.3, 0
GOSUB 100
IF lu_level_style = "建筑标高" THEN GOSUB 200 !!! 文字线

IF LU_DLon = 1 THEN
GOSUB 500 !!!基线
ENDIF
del 1

END

```

变换文字的锚固点



当前绘图比全局变量例

```

99: !!!文字
SET STYLE "AC_STYLE_1"

if LU_Y < 0 then mulx -1
    if lu_sty_on = 1 then
        TEXT2 LU_X, LU_Y, lu_bg
        for i = 1 to lu_ceng_s
            lu_bg = "(" + STR( "%m", lu_ceng_h * i ) + ")"
            text2 LU_X, LU_Y + LU_TEXT_H * i * 0.12 , lu_bg
        next i
    else
        TEXT2 LU_X, LU_Y, lu_bg
    endif
RETURN
!!!!=====

100: !!!三角形
set fill "Foreground"
POLY2_B {2} 4, fill_on , lu_fillpen , 0,
0, 0, 0,
0, 0, 1,
0.3, 0.3, 1,
-0.3, 0.3, 1,
0, 0, -1
RETURN
!!!!=====

200: !!! 文字线
if abs( lu_y ) > 0.8 then
line2 0, 0, 0, abs(lu_y) - 0.05
line2 0, abs(lu_y) - 0.05 , LU_tL, abs(lu_y) - 0.05 !!拉出的文字线
else
LINE2 0, 0.3, LU_tL, 0.3 !!普通的文字线
endif
RETURN
!!!!=====

500: !!!基线
HOTSPOT2 0, 0, unid, LU_DL, 1+128 : unid = unid + 1
HOTSPOT2 1, LU_DL, unid, LU_DL, 3 : unid = unid + 1
HOTSPOT2 -LU_DL, 0, unid, LU_DL, 2 : unid = unid + 1
if LU_DL > 0 then line2 -LU_DL, 0, 0.3, 0 else line2 -LU_DL, 0, -0.3, 0 !!基线
RETURN

```



小结:

- 记得这是最早做的一个GDL，当时的AC没有这样的标高符号，所以尝试制作。
- 用到了全局变量有： GLOB_CONTEXT ; GLOB_hstory_ELEV ; GLOB_ELEVATION , GLOB_SCALE 等。
- 拉动符号，导致符号镜像，带来文字位置变化，注意锚固点的设定。
- STR () 有很丰富的内容，需要注意。

STR{2}(格式_字符串, 数值_表达式 [, 高_精度_字符串])

%[0或更多标志][字段_宽度][.精度] 换算_指定符

标志 (用于m、mm、cm、e、df、di、sqm、sqcm、sqf、sqi、dd、gr、rad、cum、l、cucm、cumm、cuf、cui、cuy、gal) :

举例:

h = 23

nr = 0.345678

TEXT2 0, h, STR ("%m", nr) !0.346

TEXT2 0, h-1, STR ("%#10.2m", nr) !35

TEXT2 0, h-2, STR ("%4cm", nr) !34.5678

TEXT2 0, h-3, STR ("%12.4cm", nr)!34.5678

TEXT2 0, h-4, STR ("%6mm", nr)!345.678000

TEXT2 0, h-5, STR ("%15e", nr)!+3.456780e-01

TEXT2 0, h-6, STR ("%ffi", nr) !1'-2"

TEXT2 0, h-7, STR ("%0.16ffi", nr) !1'-1 5/8"

TEXT2 0, h-8, STR ("%3fdi", nr) ! 1'-1.609"

TEXT2 0, h-9, STR ("%10.4df", nr) ! 1.1341'

TEXT2 0, h-10, STR ("%0.64fi", nr) !13 39/64"

TEXT2 0, h-11, STR ("%12.4di", nr)!+13.6094"

TEXT2 0, h-12, STR ("%3sqm", nr) ! 346

TEXT2 0, h-13, STR ("%sqcm", nr) !+3,456.78

TEXT2 0, h-14, STR ("%2sqmm", nr)! 345,678.00

TEXT2 0, h-15, STR ("%12sqf", nr) !3.72

TEXT2 0, h-16, STR ("%10sqi", nr)! 535.80

alpha = 88.657

TEXT2 0, h-17, STR ("%10.3dd", alpha)!+88.657°

TEXT2 0, h-18, STR ("%1dms", alpha)!88°39'

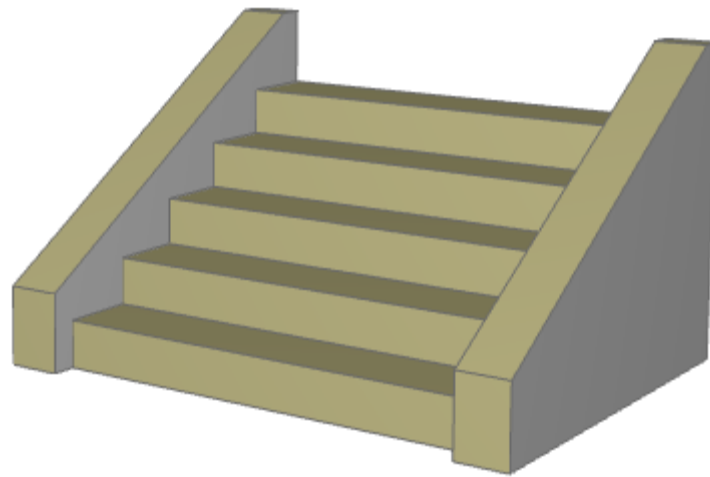
TEXT2 0, h-19, STR ("%2dms", alpha) !88°39'25"

TEXT2 0, h-20, STR ("%10.4gr", alpha) !98.5078G

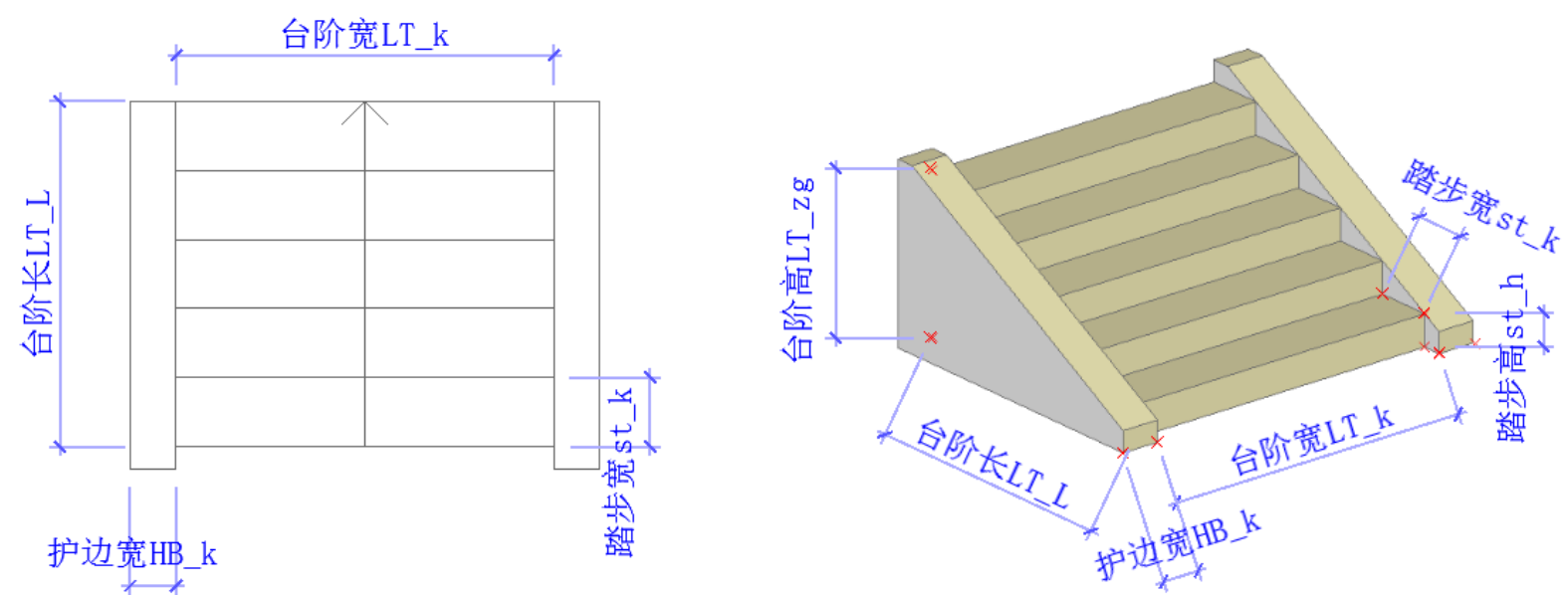
TEXT2 0, h-21, STR ("%rad", alpha) !1.55R

TEXT2 0, h-22, STR ("%2surv", alpha) !N 1°20'35" E

PART.3 向3D进阶——室外踏步



先分析这样的踏步，需要哪些参数？

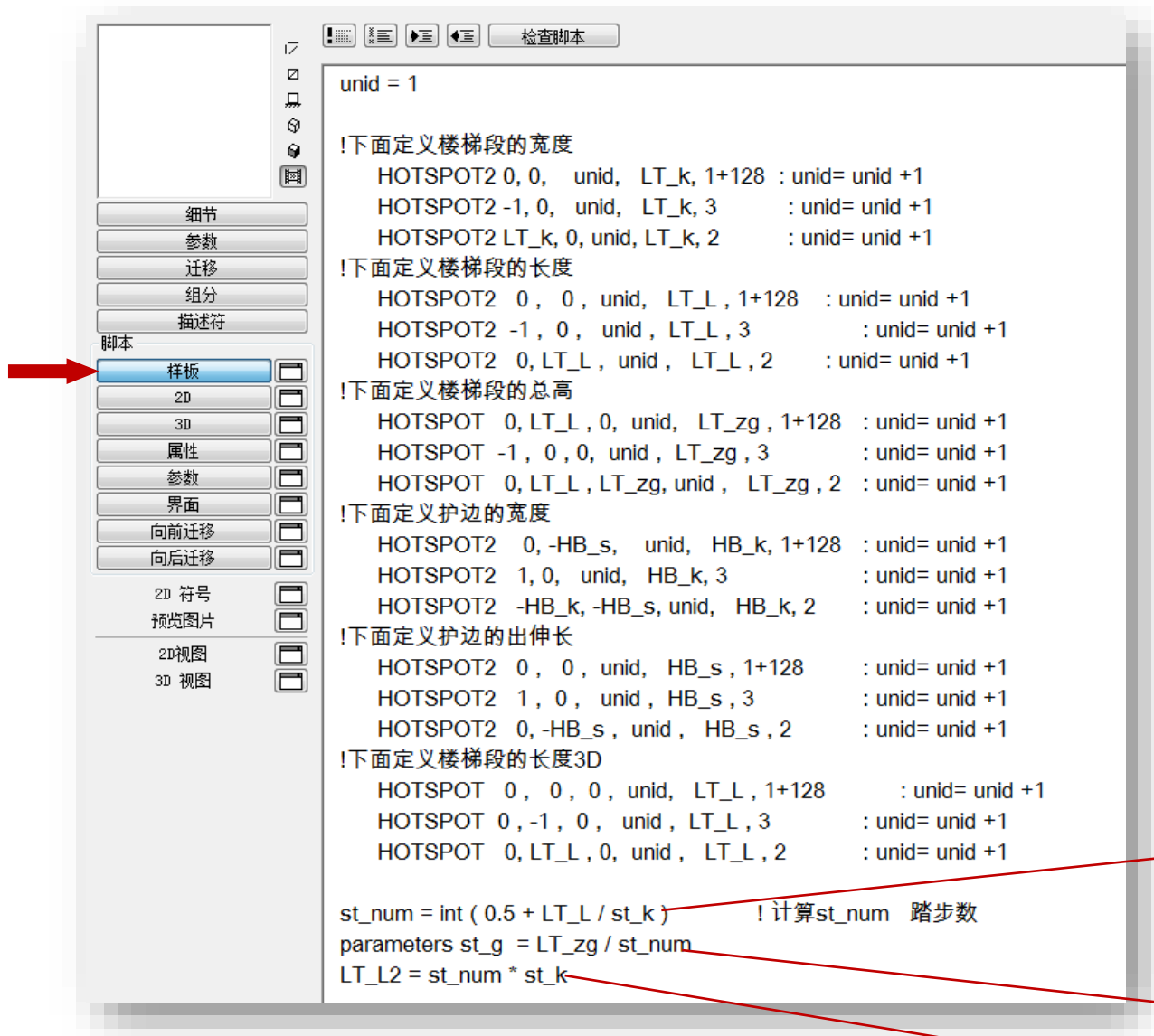


↕	B	LT_zg	↔	台阶总高	900
↕	B	LT_L	↔	台阶长	1500
↕	B	st_k	↔	踏步宽	300
↕		st_g	↔	踏步高	180
↕		LT_k	↔	台阶宽	1000
↕		st_h	↔	踏步板厚	100
✕		st_num	⋮	踏步数	6
↕		_sp0	≡	材质	
↕	↗	LT_mat1	🔲	踏面材料	67
↕	↘	LT_mat2	🔲	侧面材料	60
↕		HB_k	↔	护边宽	200
↕		HB_g	↔	护边高	300
↕		HB_s	↔	护边处伸	100
↕		Hb_z	☑	有左护边	开
↕		Hb_y	☑	有右护边	开

台阶总高

小技巧：命名参数名称的时候，可以拼音首字或英文混合都可以，尤其对英文不好的人来说。这样参数名不至于太长，自己又能看懂。但注意两个问题：

1. 参数名不要与GDL命令，全局变量重名
2. 自己做的各个GDL部件，相同意义的参数名称最好一致，便于自己日后解读。



小技巧：做3D物件的时候，往往先考虑3D程序，2D部分先用project2代替，当调试成熟了再优化2D。

在样板里，可以写2d和3d的可编辑热点，且都有效。但个人不是很建议这样做，容易条理不清晰 ☹

踏步数是根据 初始台阶长度LT_L2/踏步宽st_k 计算出来的。而不是输入的常数，这样符合设计过程。这里也用到了四舍五入和取整数。

根据踏步数，计算总高/踏步数 = 台阶高。更新参数st_g

踏步数* 踏步宽 = 真实的台阶长度。更新台阶长度为LT_L2

从一个台阶做起

```
cprism_    LT_mat2, LT_mat2, LT_mat1,  
6,         LT_k ,          !LT_k = 楼梯宽  
st_k,      st_g,  15,      !st_k = 踏步宽,st_g = 踏步高  
0,         st_g,  15,  
0,         0,   10,  
0,        -st_h,  13,      !st_h = 踏步板厚  
st_k,      st_g-st_h,  8,  
st_k,      st_g,   -1
```



拼接

