

线性及非线性规划计算软件

L I N G O

Lingo是一个目前求解非线性规划的常用软件包，同时它也能够求解线性规划，但是，如果将Lingo用于求解线性规划，则其计算速度要比只能求解线性规划的软件Lindo慢得多。因此，如果专门求解线性规划，而且自变量个数或者限定条件较多，请使用Lindo，这两个软件都是同一个公司开发的。或者使用mathematica，它也是计算线性规划的较好的软件包。

Lingo和Lindo可从<http://www.lindo.com>下载，并且下载后即可安装使用，不过安装后都是demo版本，demo版本对变量的个数、限定条件的个数等等都有限制，因此，实际使用价值不大。

安装完成后，启动Lingo, 你会看到如下窗口：



将求解内容填入窗口后，按求解按钮，则得到计算结果如下图：



Solution Report - LINGO1



Rows= 2 Vars= 2 No. integer vars= 0 (all are linear)
Nonzeros= 5 Constraint nonz= 2 (2 are +- 1) Density=0.833
Smallest and largest elements in abs value= 1.00000 100.000
No. < : 1 No. =: 0 No. > : 0, Obj=MAX, GUBs <= 1
Single cols= 0

Global optimal solution found at step: 0
Objective value: 300.0000

Variable	Value	Reduced Cost
X	0.0000000	1.000000
Y	100.0000	0.0000000

Row	Slack or Surplus	Dual Price
1	300.0000	1.000000
2	0.0000000	3.000000

最后计算结果是： $x=0$, $y=100$

可见，LINGO是一个简便的线性规划与非线性规划计算程序，下面用一个例子来说明LINGO的基本概念，启动LINGO后，在命令窗口中输入：

$$\text{MAX} = 2 * X + 3 * Y;$$

$$X + Y \leq 100;$$

这就是一个最简单的线性规划，输入后，按求解键，LINGO就会给出计算结果。其中结果用求解窗口的形式给出，具体参见LINGO的帮助文件给出的详细说明。此处， $2X + 3Y$ 称为目标函数，本例中求最大值。 $X + Y \leq 100$ 为约束条件。而 X 、 Y 为变量。请注意，LINGO中的每一行都以分号结束。如果有多个约束条件，就从上面的第三行一直往下写。与LINDO一样，LINGO中 $<$ 与 \leq 均代表 \leq ， $>$ 与 \geq 也均代表 \geq 。LINGO中的注解命令也是符号！

若一个命令或语句一行写不下，可以分多行写，但是，最后结束命令时，一定别忘了打个分号“;”，一行内也可写多个命令，只要每个用分号分开即可，也就是说，分号是LINGO的分隔符。LINGO的命令从来不区分大小写，当你在LINGO中定义变量时，每个变量都要以26个字母开始，后面可跟数字或者下划线，最多可以32个字符长。下面是2个例子。

例1 下面是一个最简单的非线性规划，它实质上是求 $(x-y)^2+(z-2)^2+4$ 的最小值。在LINGO中输入下式后，按求解图标，看一看LINGO会给出什么结果。

```
MIN=x^2-2*x*y+y^2+z^2-4*z+8;
```

```
x>=0; y>=0; z>=0;
```



LINGO Model - LINGO1

```
MIN=x^2-2*x*y+y^2+z^2-4*z+8;
x>=0; y>=0; z>=0;
```

Solution Report - LINGO1

```
Rows=      4 Vars=      3 No. integer vars=      0
Nonlinear rows=      1 Nonlinear vars=      3 Nonlinear constraints=      0
Nonzeros=      7 Constraint nonz=      3 Density=0.438
No. < :      0 No. =:      0 No. > :      3, Obj=MIN Single cols=      0
```

```
Local optimal solution found at step:      15
Objective value:      4.000000
```

Variable	Value	Reduced Cost
X	0.0000000	0.0000000
Y	0.0000000	0.0000000
Z	2.000000	-0.4440892E-08

Row	Slack or Surplus	Dual Price
1	4.000000	1.000000
2	0.000000	0.000000
3	0.000000	0.000000
4	2.000000	0.000000

用LINGO求解规划问题，那真是方便极了！

例 2 下面目标函数的原型是

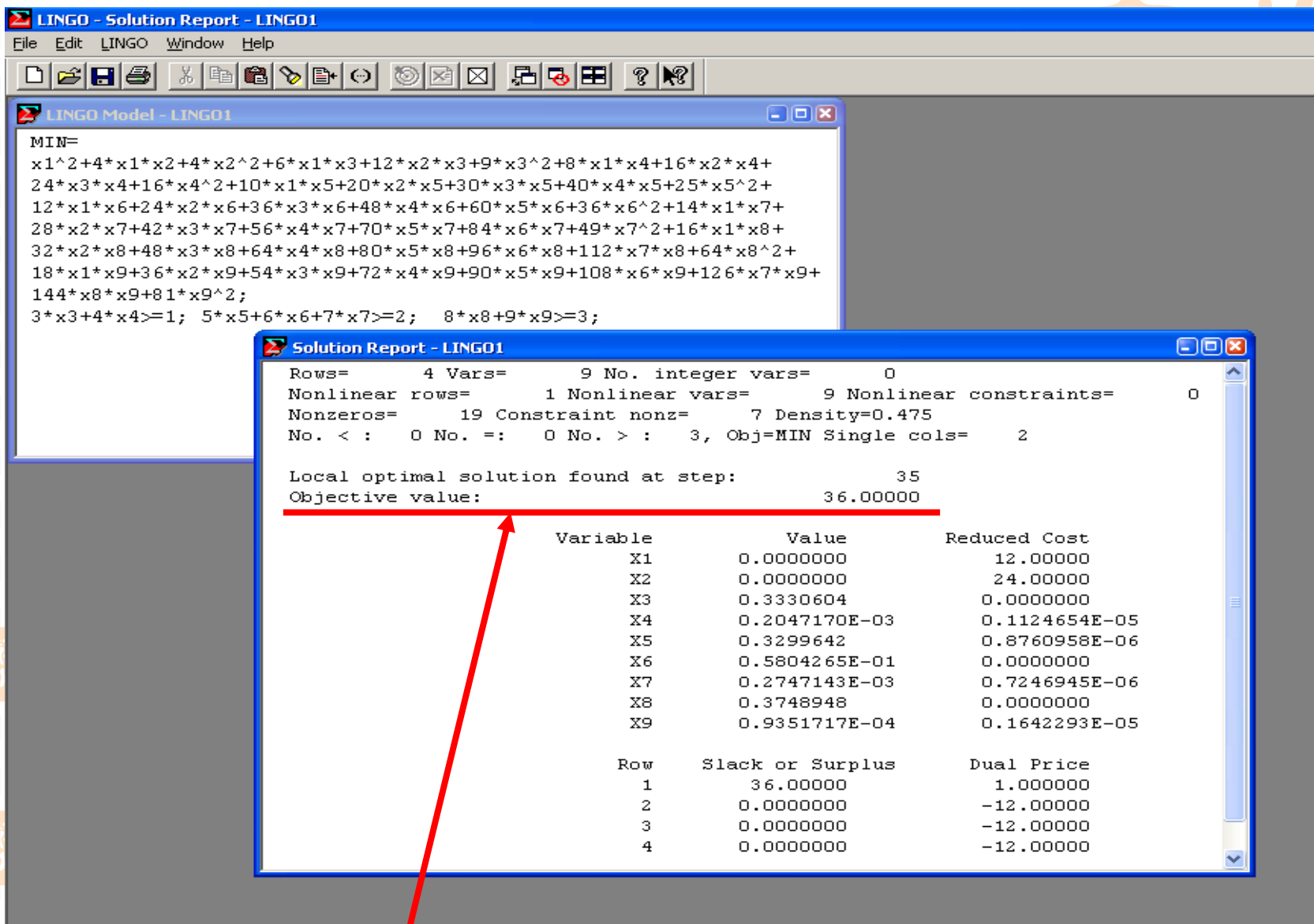
$$(x_1 + 2x_2 + 3x_3 + 4x_4 + \dots + 8x_8 + 9x_9)^2$$

在 L I N G O 中输入下列命令：

MIN=

$$\begin{aligned} & x_1^2 + 4x_1x_2 + 4x_2^2 + 6x_1x_3 + 12x_2x_3 + 9x_3^2 + 8x_1x_4 \\ & + 16x_2x_4 + 24x_3x_4 + 16x_4^2 + 10x_1x_5 + 20x_2x_5 + 30x_3x_5 \\ & + 40x_4x_5 + 25x_5^2 + 12x_1x_6 + 24x_2x_6 + 36x_3x_6 + 48x_4x_6 \\ & + 60x_5x_6 + 36x_6^2 + 14x_1x_7 + 28x_2x_7 + 42x_3x_7 + 56x_4x_7 \\ & + 70x_5x_7 + 84x_6x_7 + 49x_7^2 + 16x_1x_8 + 32x_2x_8 + 48x_3x_8 \\ & + 64x_4x_8 + 80x_5x_8 + 96x_6x_8 + 112x_7x_8 + 64x_8^2 + 18x_1x_9 \\ & + 36x_2x_9 + 54x_3x_9 + 72x_4x_9 + 90x_5x_9 + 108x_6x_9 \\ & + 126x_7x_9 + 144x_8x_9 + 81x_9^2; \end{aligned}$$

$$3x_3 + 4x_4 \geq 1; \quad 5x_5 + 6x_6 + 7x_7 \geq 2; \quad 8x_8 + 9x_9 \geq 3;$$



最后得到最小值36

在LINGO中，输入限制条件时，如果限制条件很多，那么如果某个限制条件出错，LINGO给出的错误信息是一个行号，告诉你某行出现错误，但如果你在限制条件前面加上用方括号括起来的某个字符串，参见LINGO目录下的SAMPLE\BOX程序，那么此时限制条件出错，LINGO给出的错误信息会一目了然，而且LINGO最后给出的分析结果你也会看得清清楚楚，不过这个字符串应该符合变量的定义规则，建议你用这种方法输入限制条件。通过以上介绍，就完全可以计算了，下面我们介绍LINGO的语法。

算术运算符：+，-，*，/，^，用括号括起来的式子优先级最高。

逻辑运算符：#NOT#，#EQ#，#NE#，#GT#，#GE#，#LT#，#LE#，#AND#，#OR#。这与FORTRAN一致，主要用在：@MIN，@MAX，@SUM，@FOR中。

关系运算符：=，<=，>=。请注意，LINGO中没有严格的>或<，如果你真要用到的话，比如 $A < B$ ，请将它转化为 $A + e < B$ ，其中 e 是一个近似于零的数。

数学函数: @ABS(X), @COS(X), @EXP(X), @FLOOR(X)(返回X的整数部分,若 $X>0$,它返回小于X的最大整数,若 $X<0$,它返回大于X的最大整数), @LGM(X)[它是 $(X-1)!$ 的阶乘取自然对数,即 $\ln((X-1)!)$], @LOG(X)(自然对数), @SIGN(X)(符号函数), @SIN(X), @SMAX(X1,X2,...,XN)(求N个数的最大值), @SMIN(X1,X2,...,XN), @TAN(X)(正切函数).

概率函数: 参见LINGO 5.0 USER'S MANUAL--->LINGO'S OPERATORS AND FUNCTIONS--->PROBABILITY FUNCTIONS.

变量域函数: 如果不用下面的函数明确修改变量的取值范围,LINGO认为每个变量都是非负实数,你可用下面的命令修改LINGO的默认值. @BIN(var)限制变量VAR为一个二进制数,即VAR取值为0或1, @BND(A,var,B)限制变量VAR在区间[A,B]内, @FREE(var)取消变量VAR是非负数的限制,即它可以是负数, @GIN(var)限制变量VAR为一个整数.

以上是对LINGO所作的最简单的介绍, LINGO中即提供了一个详细的帮助文件, 这可在LINGO中的HELP菜单得到, 同时它也提供了几十个演示程序, 这可以在LINGO中打开FILE菜单的OPEN选项, 然后选取\LINGO目录中的\SAMPLES子目录, 这个子目录中都是LINGO的例子, 用OPEN装入后即可求解. 下面是LINGO中SAMPLE目录下的演示程序BOX.LG4.

! 程序名: BOX.LG4; !是对程序的注解;

! 每一个规划都以"MODEL:"开始, 又以"END"结束, 也可以省略此结构;

MODEL:

! 下面表示求 $2*(.05*(d*w+d*h)+.1*w*h)$ 的最小值,LINGO中的命令都以分号结束;

! 用方括号"[]"括起来的字符串如[COST]对实际计算没有什么用处, 但它会使LINGO的输出结果一目了然, 具体请看看输出结果, 你就知道这些方括号的用处了;

[COST] min = $2*(.05*(d*w + d*h) + .1*w*h)$;

! 以下都是约束条件;

[SURFACE] $2*(h*d + h*w + d*w) \geq 888$;

[VOLUME] $h*d*w \geq 1512$;

[NOTNARRO] $h/w \leq .718$;

[NOTHIGH] $h/w \geq .518$;

[FOOTPRNT] $d*w \leq 252$;

END

在按下求解键后,LINGO输出计算结果为(其中汉字是我加入的,使你能看明白):

! 总共6行, 3个变量, 整数变量没有

! Rows= 6 Vars= 3 No. integer vars= 0

! 非线性行数 6行 非线性变量 3个 非线性限制条件 5个

Nonlinear rows=6 Nonlinear vars=3 Nonlinear constraints=5

Nonzeros= 20 Constraint nonz=12 Density=0.833

• No. < : 2 No. =: 0 No. > : 3, Obj=MIN Single cols= 0

! 进行到第9次求出解来, 其解为50.96508

Local optimal solution found at step: 9

Objectiv value: 50.96508

! 以下是具体数值

Variable	Value	Reduced Cost
D	23.03096	0.0000000
W	9.562196	0.0000000
H	6.865657	0.0000000

! 下面是约束条件的误差限

Row	Slack or Surplus	Dual Price
COST	50.96508	1.000000
SURFACE	-0.1247204E-05	-0.2342588E-01
VOLUME	-0.1937612E-05	-0.1329933E-01
NOTNARRO	0.1185074E-10	2.298546
NOTHIGH	0.2000000	0.0000000
FOOTPRNT	31.77343	0.0000000

上面计算得出, 在最小值处 $D=23.03096$, $W=9.562196$, $H=6.865657$, 下面我们将 D 限制在1.5到10.5之间, 取消 W 是非负值的限制, 让 H 取一个整数值, 则改成如下:

!程序名: BOX.LG4, 已经改动

MODEL:

[COST] min = 2*(.05*($d*w$ + $d*h$) +.1* $w*h$);

@BND(1.5, d ,10.5); @FREE(W); @GIN(H);

[SURFACE] 2*($h*d$ + $h*w$ + $d*w$) >= 888;

[VOLUME] $h*d*w$ >= 1512;

[NOTNARRO] h/w <= .718;

[NOTHIGH] h/w >= .518;

[FOOTPRNT] $d*w$ <= 252;

END

其实, @BND(1.5,d,10.5)可以在限制条件中, 用 $D \geq 1.5$ 及 $D \leq 10.5$ 来代替, 但@FREE(W)与 @GIN(H)是不能用别的来代替的, @GIN()可用于求整数规划. 将此程序运行一下, 看看会有什么结果? 下面我们再提出一个问题: 如果目标函数在不同的区域中, 取不同的值, 那么, 怎么计算, 你可能注意到, 这里的关键是目标函数的表达方式, 可以用LINGO中的逻辑运算符: #NOT#, #EQ#, #NE#, #GT#, #GE#, #LT#, #LE#, #AND#, #OR#. 这些运算符, 当条件满足时, 返回数值1, 否则返回数值0, 比如对(D #LT# 1), 当 $D < 1$ 时, 其值为1, 否则为0, 因此, 如果将BOX.LG4的目标函数改一下, 改成当 $D < 1, W < 2, H < 3$ 时, 还是原来的那个函数, 而当D,H,W取其它值时, 目标函数变成 $3 \times (D + W + H)$, 则目标函数就变成如下形式:

MODEL:

[COST] min = ((d #LT# 1) #AND# (W #LT# 2) #AND# (H #LT# 3))*2*(.05*(d*w + d*h) +.1*w*h) + #NOT#((d #LT# 1) #AND# (W #LT# 2) #AND# (H #LT# 3))*3*(W+D+H);

[SURFACE] 2*(h*d + h*w + d*w) >= 888;

[VOLUME] h*d*w >= 1512;

[NOTNARRO] h/w <= .718;

[NOTHIGH] h/w >= .518;

[FOOTPRNT] d*w <= 252;

END

下面是修改后的运行结果：

Local optimal solution found at step: 14

Objective value: 109.9803

Variable	Value	Reduced Cost
D	12.43758	0.1774019E-08
W	14.09926	0.0000000
H	10.12327	0.0000000

Row	Slack or Surplus	Dual Price
-----	------------------	------------

COST	109.9803	1.000000
------	----------	----------

SURFACE	0.1136868E-12	-0.6192584E-01
---------	---------------	----------------

VOLUME	263.2218	0.0000000
--------	----------	-----------

NOTNARRO	0.0000000	4.041292
----------	-----------	----------

NOTHIGH	0.2000000	0.0000000
---------	-----------	-----------

FOOTPRNT	76.63941	0.0000000
----------	----------	-----------

可能你觉得, 目标函数的这种写法太复杂, 都几乎看不清是怎么回事了, 那么, 我们引入2个变量, 你就一目了然了.

MODEL:

$A = ((d \leq 1) \text{ \#AND\# } (W \leq 2) \text{ \#AND\# } (H \leq 3));$

$B = \text{\#NOT\#}((d \leq 1) \text{ \#AND\# } (W \leq 2) \text{ \#AND\# } (H \leq 3));$

$[\text{COST}] \min = A * 2 * (.05 * (d * w + d * h) + .1 * w * h) + B * 3 * (W + D + H);$

$[\text{SURFACE}] \quad 2 * (h * d + h * w + d * w) \geq 888;$

$[\text{VOLUME}] \quad h * d * w \geq 1512;$

$[\text{NOTNARRO}] \quad h / w \leq .718;$

$[\text{NOTHIGH}] \quad h / w \geq .518;$

$[\text{FOOTPRNT}] \quad d * w \leq 252;$

END

这次的运算结果与上面的一样, 只不过LINGO还会给出A和B的值.

LINGO中有一个语句DATA, 可以利用它来输入数值数据. 请看对BOX.LG4的修改

MODEL:

DATA: ! 此语句必须以"DATA:"开始, 以"ENDDATA"结束;

const1 = 1512; const2 = 0.1; ! 这是定义2个常量;

input1 = ?; ! 程序运行到此处时,会弹出一个对话框,要求你输入input1的值;

input2 = ?; ! 输入input2的值;

! 也可写成 input1 , input2=?, ?;

ENDDATA

[COST] min = input1*(.05*(d*w + d*h) + const2*w*h);

[SURFACE] 2*(h*d + h*w + d*w) >= input2;

[VOLUME] h*d*w >= const1;

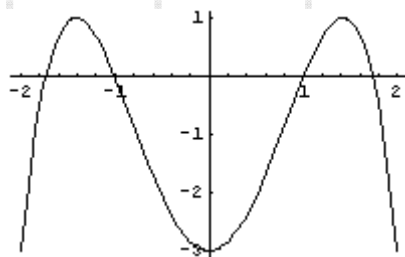
[NOTNARRO] h/w <= .718;

[NOTHIGH] h/w >= .518;

[FOOTPRNT] d*w <= 252;

END

再看看下面的函数， $y=4x^2-x^4-3$ ，其函数图象为：



其最大值点为为 $x = \pm \sqrt{2}$ ，下面是运用LINGO求解的程序：

model:

[MAX] max=4*x^2-x^4-3;

@free(x);

init:

x=2;

endinit

end

其中,“init: ... endinit”命令是给变量设一个初值,不同的初值不仅会影响求解的精度,而且对某些问题,有可能会给出不同的结果,例如,对此问题,初值 $x=2$ 时将会得到最大值点为1.414213,取初值 $x=-2$,会得到-1.414213,但请注意, @free(x)不能忘了写,它表示取消 x 非负值的约定,否则只能得到1.414213.

有了以上的说明,你就可用LINGO进行简单规划问题的求解了,但是,如果你还想用到LINGO中更为复杂的功能,求解更为复杂的规划问题,就得了解LINGO的语法.

下面是LINGO的语法说明:

SET命令: SET是LINGO中的一个最基本的命令,使用它,你可以将相似的限制条件归结到一起,将一个很长的表达式用更为简单的方式表达出来. SET中的每个成员可能有一个或多个属性, SET命令与其它计算机语言中对变量的定义是一样的.

在一个求解模型中,可以有SET命令,可以没有SET命令,也可以有多个SET命令,它的定义如下:

SET:

setname [/ member_list /] [: attribute_list];

.....

ENDSET

SET在模型中的位置要满足:当求解模型用到SET中的成员(即变量)时,它一定被定义. 你可以将上面的SETNAME理解成模型中的一个变量,如果有MEMBER_LIST,则表示该变量是一个数组, MEMBER_LIST为该数组的取值范围,而最后面的ATTRIBUTE_LIST(属性)就是变量的实际值,此值的具体大小一般在DATA中定义. 其中,方括号中的内容都是可选的. 请看下面的例子.

SETS:

PRODUCT / A B/;

MACHINE / M N/;

WEEK / 1..10/;

ALLOWED(PRODUCT, MACHINE, WEEK);

X // : X_RANGE;

Y /1 2 3 4 5 6/: Y_RANGE;

ENDSETS

变量PRODUCT是数组,取值于PRODUCT(A)或PRODUCT(B),而WEEK也是数组,取值于WEEK(1)到WEEK(10), ALLOWED是由上面的3个数组变量定义的,它实际上是一个3维数组. X则是一个变量,它的值取自于X_RANGE. Y是一个数组即Y(1)到Y(6),它取值Y_RANGE.而SET中所出现的变量的属性值由DATA语句来设置.

SETS:

SET1 /A, B, C/: X, Y;

ENDSETS

DATA:

X = 1 2 3; Y = 4 5 6;

ENDDATA

实际上，上面的说法有些地方是不确切的，这是因为LINGO中的数组与我们通常所提及的数组还是有点不一样。下面我编了一个演示程序，它不能计算优化问题，它和我们所通常看到的按顺序执行的计算机程序是一样的，这个程序演示了数组、FOR循环、SUM命令的使用方法，对于其它的命令，可以参考此程序，其中，以“！”开始的行是注释行。

如果你想在LINGO中执行此程序，不用重新输入，只要用鼠标从都是“！”的行的下一行开始，一直选中到本文件的结束，然后将这段COPY到LINGO中即可。另外，这段程序在光盘上的LINGO目录下，文件名是DEMO.LG4，你可以装入此程序，直接在LINGO中运行。

!用SETS与ENDSET定义数组;

sets:

!定义数组a,其引用值为value1(1),value1(2),.....,value1(5);

!在FOR,SUM等命令中的引用名称为a(1),a(2),.....,a(6) ;

a/1..5/:value1;

!定义数组b,其引用值为value2(1),value2(2),.....,value2(6);

!在FOR,SUM等命令中的引用名称为b(1),b(2),.....,b(6) ;

b/1..6/:value2;

!定义数组c,其引用值为:

!value3(1,1),value3(1,2),.....,value3(1,6);

!value3(2,1),value3(2,2),.....,value3(2,6);

!.....

!value3(5,1),value3(5,2),.....,value3(5,6)

!在FOR,SUM等命令中的引用名称为c(i,j) ;

c(a,b):value3;

!定义数组d,维数与c一样;

d(a,b):value4;

endsets

!数组的赋值要用DATA与ENDDATA命令;

data:

!为数组a,b,d赋值;

!value1(1)=11,value1(2)=22,value1(3)=33,value1(4)=44,value1(5)=55;

value1=11 22 33 44 55;

!value2(1)=1,value2(2)=2,value2(3)=3,value2(4)=4,value2(5)=5,value2(6)=6;

value2=1 2 3 4 5 6;

!value4(1,1)=1,value4(1,2)=2,value4(1,3)=3,value4(1,4)=4,value4(1,5)=5,value4(1,6)=6;

!value4(2,1)=7,value4(2,2)=8,value4(2,3)=9,value4(2,4)=10,value4(2,5)=11,value4(2,6)=12;

!value4(3,1)=13,value4(3,2)=14,value4(3,3)=15,value4(3,4)=16,value4(3,5)=17,value4(3,6)=18;

!value4(4,1)=19,value4(4,2)=20,value4(4,3)=21,value4(4,4)=22,value4(4,5)=23,value4(4,6)=24;

!value4(5,1)=25,value4(5,2)=26,value4(5,3)=27,value4(5,4)=28,value4(5,5)=29,value4(5,6)=30;

value4= 1 2 3 4 5 6

7 8 9 10 11 12

13 14 15 16 17 18

19 20 21 22 23 24

25 26 27 28 29 30;

enddata

!下面是一个二重循环,即 $C=A.B+D$;

@for(a(i):

@for(b(j):

value3(i,j)=value1(i)*value2(j)+value4(i,j)

)

);

!对数组a求和;

total_a=@sum(a(i):value1(i));

!对数组b,去掉前4个元素求和;

total_b=@sum(b(i)|i #LT# 4:value2(i));

!求数组d的和;

total_d=@sum(d(i,j):value4);

!对数组d,求对角线元素的和;

total_c=@sum(c(i,j)|i #EQ# j:value4(i,j));

