


Mathcad 笔记

-----王定略

学 mathcad 有个很重要的工具，就是帮助，一定要多用。

1. 加减乘除，和 excel 一样，除直接按“/”是分号，按“Ctrl+/"是算数里面的除号。
2. 平方根是“√”
3. 有下标的定义变量例如 L_n ， $L + \text{“.”} + n$ ，还有一种是 $L + [+ n$ ，但是后面一种不能定义变量，它是矩阵里面的下标。
4. $\text{Ctrl}+9 \leq$ ， $\text{Ctrl}+0 \geq$ 。
5. 分几种情况关系运算，“}”，或者 add line 命令，在编程工具栏  里面。Eg:

$$\begin{cases} a=3 & \text{if } x > 0 \\ a=0 & \text{if } x = 0 \\ a=-3 & \text{if } x < 0 \end{cases}$$

（这个是程序的标志符，有 add line 包着的，就是一个程序）。

6. 变量的下标可以使用中文。
7. $\text{Ctrl}+Z$ 撤销上一步操作。（很多软件都是这样的，比如 office）
8. Mathcad 变量定义顺序是从上到下、从左到右。
9. 一般短变量名没有长变量名方便理解，所以定义变量的时候尽量定义清楚点。
10. 虚数 i 、 j 不能单独使用，前面必须加数字，因为直接用会被认为是变量。 $1i$ 表示 i 。Mathcad 会显示为 i 的，但输入必须输入 $1i$ 。
11. 向量中的区间变量输入， $X:1;8$ 显示： $X:=1\cdots 8$ ，默认步长是 1。如果步长不为 1，则先输入， $X: 1,3;9$ 显示： $X: =1,3\cdots 9$ ，步长是 2。("; "表示 \cdots)
12. 常用的几个内置函数

矩阵函数

函数名	函数意义
Rows(W)	矩阵 M 的行数
Cols(W)	矩阵 M 的列数
Identity(n)	产生 n 阶单位矩阵
tr(W)	矩阵 M 的迹
augment(M1,M2)	由矩阵 M1，M2 生成的增广矩阵

对数和指数函数

函数名	函数意义
exp(Z)	指数函数
ln(x)	自然对数
Log(x)	以 10 为底的对数函数

统计函数

函数名	函数意义
Mean(V)	数据向量 V 各元素的均值
Stdev(V)	数据向量 V 的标准方差
Var(v)	数据向量 V 的方差

求解方程函数

函数名	函数意义
Root(exp,var)	求解表达式等于 0 的解
find(var1,var2)	在求解模块中求满足约束条件的解
minerr(var1,var2)	在求解模块中求使约束条件极小化误差的解


13. 自定义函数很简单,就是自己在 mathcad 里面写上去就行了,例如 $f(x,y)=x^2+y^2$ (函数参数要用括弧括在里面,参数之间用逗号隔开),不过有一点要注意,就是参数的参数,就是括弧里面的参数之前不用定义,除此以外的参数,在函数前都必须定义。
14. 如果输入表达式太长,可以输入 Ctrl+Enter 换行,不过要注意的是,Ctrl+Enter 换行后是加号,也就是说必须是加号的位置才能换行。
15. 在 mathcad 中,以 b、o 和 h 作为后缀的数值分别表示为二进制数、八进制数和十进制数。所以,以 b, o, h 作为变量名的时候,数值和它相乘,一定要输入“*”乘号。
16. 乘号的形状,默认是个小点,可以修改,在“数学-选项-显示”里面。(2001 里面这样设置,13 以后可以直接右键里面选择乘号的样子,也可以“工具-工具表选项-显示”"tools-worksheet options-display"里面设置乘号的样子。)
17. 一个数组连乘或连加,先把数组定义成区间变量 $X:=2,4\cdots 10$,然后 $\sum_x x = 30$ 或

$$\prod_x x = 3.84 \times 10^3。$$

18. 布尔运算一共有十种，分别是 大于(>)，小于(<)，等于(=)，不等于(\neq)，大于或等于(\geq)，小于或等于(\leq)，与运算(\wedge)，或运算(\vee)，异或运算(\oplus)，非运算（取反运算）(\neg)。（方程，不等式的符号都是布尔运算符）

布尔运算	输入方法	返回值
大于	>	$X > Y$, 返回 1; 否则为 0
小于	<	$X < Y$, 返回 1; 否则为 0
等于	Ctrl+=	$X = Y$, 返回 1; 否则为 0
不等	Ctrl+3	$X \neq Y$, 返回 0; 否则为 1
大于等于	Ctrl+0	$X \geq Y$, 返回 1; 否则为 0
小于等于	Ctrl+9	$X \leq Y$, 返回 1; 否则为 0
与运算	Ctrl+Shift+7	X 和 Y 都为真返回 1; 否则为 0
或运算	Ctrl+Shift+6	X 或 Y 为真返回 1; 否则为 0
异或运算	Ctrl+Shift+5	X 或 Y 有且只有一个为真返回 1; 否则为 0
取反运算	Ctrl+shift+1	X 为真返回 0; 否则为 1

Eg: $x:=1$ $y:=2$ $Z:=x=y$ 则 $Z=0$

19. 布尔运算在除了判断，在实际运算中也有很大的作用，比如， $z=x*y$ ， $x>y$ 时， z 取 $x*y$ ，否则 z 取 0。就可以写成 $z:=(x>y)*x*y$ 。
20. 矩阵的下标，默认从 0 开始。比如 $X:=(1,2,3,4,5)$ ，那么 $X_0=1, X_1=2, \dots$ 。这个默认起始值可以调整，2001 版中在“数学-选项-内建变量-数值初始值”里面，13,14 版在“工具-工作表选项-内置变量-数组原点” “tools-worksheet options-built-in variables=array origin”里面。
21. Mathcad 中矩阵下标， $A_{x,y}$ ，第一个表示行，第二个表示列。
22. 矩阵计算中，如果加了一个数值，比如 $A+4$ ，则这个数值被认为是所有元素都为 4 的矩阵。
23. Mathcad 矩阵计算中，提供了一种比较怪异的功能，矩阵向量化 ，就是两个矩阵运算时，一旦用了矩阵向量化运算符后，就会变成矩阵每个元素对应另一个矩阵的元素间的运算，然后在组成矩阵。例如，

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, B = \begin{pmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 1 & 2 \end{pmatrix}, \text{ 则 } \overline{A \cdot B} = \begin{pmatrix} 3 & 8 & 15 \\ 24 & 35 & 48 \\ 63 & 8 & 18 \end{pmatrix}.$$

24. Mathcad 可以通过“插入-数据-文件输入”从外界文档中获得数据，一般是 txt 和 excel 文档，从而形成矩阵。
25. 多项式的根，也就是一元多次方程的根。Mathcad 提供了两种函数， $\text{root}(f(\text{var1}, \text{var2}, \dots), \text{var1}, a, b)$ 和 $\text{polyroots}(v)$ 。（root 函数默认 $f(x)=0$ ，你只用输入 $f(x)$ 的式子就行了。）

用简单的例子说明

$\text{root}(f(\text{var1}, \text{var2}, \dots), \text{var1}, a, b)$ ，第一个参数 $f(\text{var1}, \text{var2}, \dots)$ 是多项式；第二个参数 var1 是要求的未知数；后两个参数 a, b 是我们给根的一个范围， $a < \text{var1} < b$ ，要记住的是 $f(a)$ 和 $f(b)$ 的符号必须相反，否则方程无解。（root 只能求一个未知数的一个根）

举个例子 eg:

$$f(x) := x^2 + 2x - 8 \quad \text{root}(f(x), x, 1, 3) = 2$$

$$f(x, y) := x^2 + y^2 + 2x - 9 \quad y := 1 \quad \text{root}(f(x, y), x, 1, 3) = 2$$

最后两个参数也是可以缺省的，不过在求根之前必须先给 x 赋个值才能计算。Eg:

$$f(x) := x^2 - 2x + 1 \quad x := 0 \quad \text{root}(f(x), x) = 1$$

有一点要注意， x 赋值不同，根的误差也不同， x 赋值离真值太远，误差可能很大。第一种方法则不存在这个问题。

$\text{polyroots}(v)$ 这个函数就没那么复杂，就是这个参数 v 是个列向量，用矩

阵命令做的。以上个例子为例， $f(x) = x^2 - 2x + 1$ $v = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$

$$\text{polyroots}(v) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

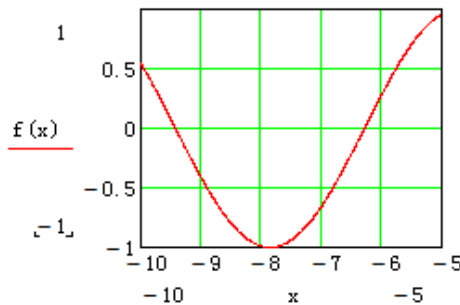
可以发现 $\text{polyroots}(v)$ 比 $\text{root}(f(\text{var1}, \text{var2}, \dots), \text{var1}, a, b)$ 函数要好用，但是 root 函数除了解多项式也能解别的函数，用法是一样的。

26. Mathcad 有比较强大的绘图功能，最常用的就是 X-Y 二维图。“插入-图表-X-Y

绘图” "insert-Graph-X-Y plot"。在图形里面，横坐标位置输入 x 坐标的参数，纵坐标输入 y 坐标的参数即可。这个自己一试就知道怎么用了。它配合 `root` 函数一起使用求值很好，因为 `root` 函数先要估一个根的范围，先把函数的图画出来，大抵的范围一看就知道了。

27. X-Y 二维图中，默认有 6 个参数可以填，分别是 x 轴参数， y 轴参数， x 轴显示范围， y 轴显示范围，

Eg:



x 轴参数是 x , y 轴参数是 $f(x)$, x 轴范围是 $(-10, -5)$, y 轴范围是 $(-1, 1)$ 。

这个 x 轴， y 轴范围可以用来调整图形显示的精度，可以用来判断根的范围。其他参数可以通过设置调出来。

28. mathcad X-Y 二维图除了能做函数的图形以外，还可以做离散点的函数， x 轴， y 轴坐标都用列向量提供，它会把对应的点连起来。（嫌占位置可以用横向量加转置）
29. 解方程组，用 `given`...`find`, `minerr`, `maximize`, `minimize` 函数计算，最常用的是 `given-find` 函数。

举个例子 EG:

$$X:=1 \quad y:=0 \quad z:=3$$

given

$$x+y+z=1$$

$$x-e^{y+z}=0$$

$$x-y+z=1$$

$$\text{find}(x,y,z)=\begin{pmatrix} 1 \\ 0 \\ -6.346 \times 10^{-6} \end{pmatrix}$$

可见，解方程组和 `root` 函数解方程一样，要先给未知数赋值。**Mathcad** 里面解方程是用的迭代法，需要一个初始值进行迭代。`given` 后给出方程组，`find` 函数求得答案。这里有几点要注意：

- 1、解方程前必须赋值；
- 2、`given` 函数必须单独占一行；
- 3、方程中的等号是布尔运算中的等号 (`ctrl+=`)，而不是等号 (`=`)。
- 4、初始赋值不同，误差也不一样，这个要小心；

5、赋初始值迭代我们可以人工干预，开始并不知道赋值多少比较好，就随便赋个值，然后按照计算的结果再进行赋值，这样结果会比较接近真值。值得注意的是，像 -6.346×10^{-6} 这种超级小的数，一般真值就是 0，我们按 0 重新赋初始值。

`minerr`，`maximize`，`minimize` 函数与 `find` 函数有什么不同呢，这个可以查看 `mathcad` 的帮助。`minerr` 函数与 `find` 函数的区别就是，不管方程有没有解，`minerr` 函数都会把迭代的结果给出来，而 `find` 函数则会提示方程无解。`maximize`，`minimize` 两个函数则是解不定方程的，求极值的。这里的 `given` 函数一般就给约束条件比如 `x>0` 之类的。`Minimize(f, var1, var2, ...)`、`Maximize(f, var1, var2, ...)`。举个例子：

$$Z(x,y):=80x+45y$$

$$X:=1 \quad y:=1$$

Given

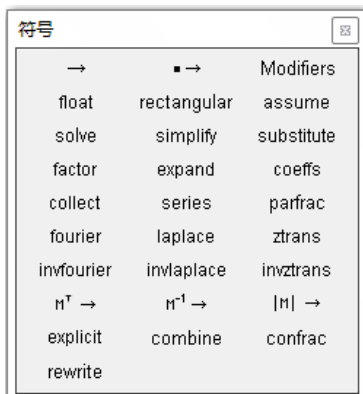
$$20x + 5y \leq 400$$

$$x \geq 0$$

$$\max imize(z, x, y) = \begin{pmatrix} 14 \\ 24 \end{pmatrix}$$

30. 三角函数，要注意的是，默认是用弧度计算，如果想用角度，可以后面加个 `deg`，或者直接乘以 $\pi/180$ 。
31. 求极限，积分，微分和求导要注意的是，最后求结果时不是用等号 (`=`)，而是和符号运算一样用箭头符号 (`→`)。求导数的时候，拐角那个点的导数是不存在的，那个点的附近求出来的导数，误差很大，这个要注意。(eg “0” 是一个奇点，求 0.0001 的导数，误差就很大)

32. 符号运算，功能主要都是在符号工具栏里面。（符号运算的功能都能用到代数计算中）



- ① “→”符号等号，mathcad14 自己是这样叫的，是用的最多的符号运算符。相当于代数运算的等号。其实也能代替等号，在代数式中运用。不过返回的是分数（如果等式里面没小数的话），等号返回的则是小数。求积分，极限，微分，导数都是用“→”。
- ② 符号运算的基本格式是：“表达式+关键字+→”，这个“关键字”在左边的工具栏中，大部分都是。

③ **simplify**关键字是化简运算符，功能是将一个多项式中多余的累赘部分去掉，成为一个最精简的式子。**expand**关键字是展开运算符，功能是将被乘方，相乘之类相连的式子释放成多项式。**factor**关键字是因式分解运算符，功能和展开运算符刚好反过来，将多项式编程连乘（除）的式子。**coeffs**关键字是求多项式系数的运算符，“多项式+coeffs+与要求系数相对应的参数+→”，功能是将多项式某个未知数的系数提出来做成一个列向量。Eg:

$2x^4+3x^2+2x+4$ coeffs, $x \rightarrow (4 \ 2 \ 3 \ 0 \ 2)^T$ ，是从0次幂开始的，如果有缺就补0。

以上四个关键字是除了符号等号用的最多的符号运算符。别的可以查阅帮助。

33. 符号运算允许多个关键字一起作用，以简化运算过程。操作方法就是点完一个关键字以后不按回车而是接着点第二个，第三个关键字就可以了。Eg:

$$x^2 + 11x + 18 \begin{matrix} \text{simplify} \\ \text{expand} \end{matrix} \rightarrow x^2 + 11x + 18.$$

34. 符号运算可以解方程，和不等式。关键字 solve。用法“式子+solve，未知数+→”。

例子： $ax^2 + bx + c = 0$ solve, $x \rightarrow \left(\begin{array}{c} \frac{b}{2} + \frac{\sqrt{b^2 - 4ac}}{2} \\ \frac{b}{2} - \frac{\sqrt{b^2 - 4ac}}{2} \end{array} \right)$ ，注意（ax 中间一定不要忘了

记了输入*乘号）。

$$x^2 > a \text{ solve, } x \rightarrow \begin{cases} \begin{pmatrix} -c1 \\ -c2 \end{pmatrix} & \text{if } \sqrt{a} < -c1 \wedge 0 \leq a \wedge -c2 < -\sqrt{a} \quad (\text{“}\wedge\text{” 是布尔运算}) \\ 0 & \text{if } a < 0 \end{cases}$$

符与， $-c1 > \sqrt{a}$ 且 $a \geq 0$ 且 $-c2 < -\sqrt{a}$ 时， x 的范围就是 $-c1$ 叠加上 $-c2$ 的范围。）

符号运算符解方程和不等式同样可以用到代数计算中去解方程和不等式。而且不需要赋初始值，虽然计算慢点，但以目前计算机的计算能力，这个时间差可以忽略不计。

$$\text{Eg: } x^2 + 11x + 18 = 0 \text{ solve, } x \rightarrow \begin{pmatrix} -2 \\ -9 \end{pmatrix}.$$

35. 符号运算解方程组，方法和代数运算是一样的，用 `given-find` 函数。只是最后那个等号改成了符号等号 (\rightarrow)。符号运算接方程组也能用到代数计算中。

例如：

$$\begin{array}{l} \text{given} \\ x + y + z = 1 \\ x - e^{y+z} = 0 \\ x - y + z = 1 \\ \text{find}(x, y, z) \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

把数字换成字母也是一样的。

36. `Substitute` 关键字，变量替换运算。EG: $x+y \text{ substitute, } x=a, y=b \rightarrow a+b.$ (=是布尔运算符)
37. `series` 关键字，泰勒级数展开。默认是在 0 点展开 ($a+a_1(x-0)+a_2(x-0)^2+\dots$)，展开到 6 次幂。用法“式子+series,a,b,c \rightarrow ” a: 针对哪个变量展开；b: 在那个点展开，用布尔等号表示；c: 最高展开到几次幂。a, b, c 三个参数可以缺省，b, c 位置可以互换。

$$\text{Eg: } \cos(x) \text{ series, } 3, x=2 \rightarrow \cos(2) - \sin(2)(x-2) - 1/2 \cos(2)(x-2)^2.$$

$$\cos(x+y) \text{ series } x, x=2, 3 \rightarrow -\cos(y)/3 - 14\sin(y)/15 - (x-2)(2/3\cos(y) - 1/3\sin(y)) + \dots$$

38. `parfrac` 关键字，将任意一个有理分式转化成分母为一次或二次的部分分式之和。对复杂的有理分式化简非常有效。

$$\text{Eg: } \frac{2x^2 - 3x + 1}{x^3 + 2x^2 - 9x - 18} \text{ parfrac, } x \rightarrow \frac{1}{3 \cdot (x-3)} - \frac{3}{x+2} + \frac{14}{3 \cdot (x+3)}.$$

39. mathcad 的图形功能也是很强大的。我们用的最多的是二维 X-Y 直角坐标系。方法在前面介绍过了。这里在补充一下。

1、可以用 $f(t)=\sin(t);g(t)=t^2$ 的方式来输入图形。只需要在横坐标位置和纵坐标位置分别输入 $f(t),g(t)$ 即可。

2、可以通过向量来作图。 $i:=1\cdots 50,x_i:=\sin(i),y_i:=i^2$ 。（只要元素用矩阵下标来定义，系统就默认它是矩阵的一个元素）方法和前面一样。

3、可以多个函数在一个 X-Y 坐标系里面作图，只需要在 y 轴位置输入两个函数就行了，它们之间用逗号“,” 隔开。

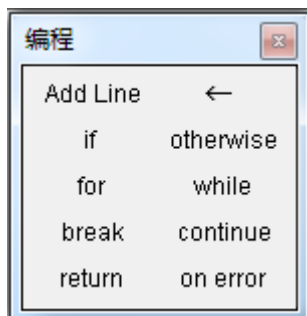
4、可以读取曲线上任意一点坐标。方法是：1、选中图形；2、格式-图标-跟踪，打开 X-Y 跟踪对话框（format-graph-trace, X-Y trace 对话框）。3、在图形上想知道坐标的地方点一下，坐标就在对话框里面出现了。

5、缩放（zoom）功能。选中图形，单击右键选择缩放（zoom），会出来一个对话框。然后在图形里面用鼠标选择一块你想放大的区域。对话框里面的放大和缩小按钮就会变亮，按下放大，你选择的部分会铺满整张图，点缩小，则会还原到原始状态。

6、双击图形会出现格式对话框，里面有好多功能。轨迹选项里面，有个选项叫做“类型”，里面有线条、点、柱状图等选项，我们画直方图就用柱状图类型显示。

40. 极坐标系和直角坐标系的使用方法很类似。就是横坐标参数表示角度（角坐标），纵坐标参数表示长度（线坐标）。最右边有两个值，表示的是长度（线坐标）的范围（ ρ 的范围）。别的操作和直角坐标系是一样的。（注意的是 $\rho:=1$ 这个函数是个圆，虽然函数式中没有角度 θ ，但实际上是有的，别以为它没有角坐标）

41. Mathcad 的程序编写，所有语句都在编程工具栏里面。



1、Add line “层次线”程序模块的标志；（在 add line 范围内的就是一个程序，出去就程序结束了）

2、“←”局部定义(赋值)，相当于代数运算的“:=”。

3、“if”选择语句；

4、“otherwise”和 if 语句一起使用，选择语句；

- 5、“for”循环语句；“while”循环语句；
- 6、“break”、“continue”中断语句，前者中断整个循环，后者只中断当前进行的一次循环；
- 7、“return”返回语句。一般层次线中执行的最后一行（不执行的不算）是程序的返回值，有时候需要中途返回一个值，就用“return”语句；
- 8、“on error”错误语句。“a+on error+b”当 b 出错时（不能计算），返回 a。
42. 关于程序的返回值，程序做执行的最后一个语句就是返回值所在语句。

Eg: $f(x) := \begin{cases} 5 \\ 2 \end{cases} \text{ if } x < 0$ ，当 $x \geq 0$ 时，5 就是最后一个执行的语句，所以返回值就是 5，当 $x < 0$ 的时候，if 语句就是最后一个句话，返回值就是 2。但如果把两个语句掉个位置就不行了，因为 5 在最后，怎么都会执行，永远是最后执行的语句，返回的结果就永远是 5 了。

43. If 语句里面的嵌套。If 语句的格式是“a+if+b”，a 是返回的值，b 是条件。嵌

套的时候，在 a 的位置键入层次线 add line，就会变成 $f(x) := \begin{cases} \text{if } x > 0 \\ \begin{cases} x \leftarrow 1 \text{ if } x < 2 \\ x \leftarrow 3 \text{ if } x \geq 2 \end{cases} \\ 8 \end{cases}$

的样子，此时 a 的位置就换到下面去了。otherwise 和 if 一起使用。表示出了 if 条件以外的条件。

44. 循环语句。

“for”语句格式:

for $x \in y$

z

x 循环变量，y 值域；z 为“for”语句下执行的语句，会计算值域 y 中每个 x 值对应的 z 值，一般 z 值会随 x 的变化而变化。

Eg:

$f(x) := \begin{cases} i \leftarrow 1 \\ \text{for } i \in 1 \dots 9 \\ x \leftarrow x + i \\ x \end{cases}$ ，i 的值域就是 1 到 9 的整数。这个值域也可以用向量给

出来。

Eg:

$$a := (1 \quad 3 \quad 8 \quad 5)$$

$$f(x) := \begin{cases} \text{for } i \in a \\ x \leftarrow x+i \\ x \end{cases}$$

, 这里 i 的值域就是向量 a , i 取值在值域中按顺序一个一个取。

“while”语句格式:

while x

y

x :条件表达式; y :运算表达式。如果 x 为真 (不为 0), 则执行运算表达式 y 。表达式 y 的计算必须导致 x 的变化。一旦 x 变为假 (=0), 循环结束。不过如果 y 表达式无法使 x 变成假, 则会陷入死循环, 只能按 Esc 键才能停止计算, 这个是要注意的。

Eg:

$$f(x) := \begin{cases} i \leftarrow 1 \\ \text{while } i < 7 \\ \quad \begin{cases} x \leftarrow x+i \\ i \leftarrow i+1 \end{cases} \\ x \end{cases}$$

(注意: 运算表达式 y 如果不止一个表达式, 要用 add

line 嵌套起来。) 这里专门用了一个变量 i 来执行循环变量, 大多数情况下, 函数变量本身就可以承担循环变量的任务。For 语句的循环, 循环次数是固定的, 而 while 语句的循环是不定的。这是二者最大的区别。

45. continue 语句和 break 语句都是用来执行中断循环的语句。二者的区别在于, continue 语句只中断正在执行的一次循环, 中断后会进跳过 continue 语句以后的语句而进入下一次循环; 而 break 语句则是中断整个循环语句, 中断后执行循环语句后面的语句。它们都和条件语句一道使用。

Eg:

$$f(n) := \begin{cases} s \leftarrow 0 \\ \text{for } k \in 0 \dots n \quad \text{if } n \geq 0 \\ \quad \begin{cases} \text{continue if } \text{mod}(k,2)=0 \\ s \leftarrow s+k \end{cases} \\ \text{for } k \in 0 \dots |n| \quad \text{otherwise} \\ s \leftarrow s+k \\ s \end{cases}$$

, $f(n)$ 的功能是当 $n \geq 0$ 时, 0 到 n 奇数之和返

回给 $f(n)$ ；当 $n < 0$ 时，取膜，0 到 $-n$ 之和。（for 语句执行语句不止一个，都要用 add line 嵌套）。

46. 这里专门强调一次，一个程序中，到底哪一句是最后一次执行的语句，要注意，要小心，特别是条件语句多的函数。

47. return 语句和 on error 语句。

return x ，读到 return 运算符，终止程序，并返回 x。

x on error y ，计算 y，如果 y 发生错误（无意义），则计算 x 并返回 x，否则返回 y 的计算结果。on error 运算符是一种特殊的条件运算符。

48. 程序的递归，也就是函数自己调用自己本身。这在每一种语言中都是非常重要的运算方式。

Eg:

$$f(n) := \begin{cases} n \cdot f(n-1) & \text{if } n \geq 2 \\ 1 & \text{otherwise} \end{cases}, f(n) \text{ 程序功能是返回 } n! \text{。里面自己调用自己 } f(n-1),$$

从而实现循环。

49. mathcad 有大量的函数，比如 $\text{mod}(x,y)$, x/y 返回余数。等，多少要了解一些。以下是几个常用的函数，更多的要去了解，最好能找个 mathcad 函数大全下来查比较好。

$\text{max}(A,B,\dots)$: 返回 A,B, \dots 所有元素中的最大值。

$\text{min}(A,B,\dots)$: 返回 A,B, \dots 所有元素中的最小值。

$\text{mod}(x,y)$: 返回余数；

$\text{round}(x,n)$: 四舍五入到小数点后 n 位，若缺省 n 则默认是 0，四舍五入取整；

$\text{Round}(z,y)$: 返回 $\text{round}(z/y) \cdot y$ ；

$\text{floor}(z)$: 返回 $\leq z$ 的最大整数； $\text{Floor}(z,y)$: 返回 $\leq z$ 的 y 的最大倍数；

$\text{ceil}(z)$: 返回 $\geq z$ 的最小整数； $\text{Ceil}(z,y)$: 返回 $\geq z$ 的 y 的最小倍数；

$\text{trunc}(z)$: 去掉 z 的小数部分，返回整数部分； $\text{Trunc}(z,y)$: 返回 $\text{trunc}(z/y) \cdot y$ 。

50. 几个常用的统计函数：（统计函数的参数都是针对矩阵的）

均值： $\text{mean}(A,B,\dots)$: A,B, \dots 所有矩阵内的所有元素相加求均值。

中位数： $\text{median}(A,B, \dots)$: 返回所有矩阵元素，从小到大排列，正中间的那个元素值，如果是偶数，则去正中两个元素的均值。

方差： $\text{var}(M)$; $\text{Var}(M)$, 前者是矩阵所有元素的最大似然估计方差，后者是所

有元素的无偏估计方差。

$$\text{var}(M) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (M_{i,j} - \text{mean}(M))^2}{m \cdot n}、\text{Var}(M) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (M_{i,j} - \text{mean}(M))^2}{m \cdot n - 1}。$$

标准方差: `stdev(M)`, `Stdev(M)`, 分别表示 $\text{var}^2(M)$; $\text{Var}^2(M)$ 。

协方差: `cvar(A,B)`。相关系数: `corr(A,B)`。

51. 几个矩阵函数:

`augment(A,B,...)`: 矩阵合成函数, 就是将矩阵 (向量) A, B, \dots 按从左到右的顺序合成矩阵。Eg: $A=(1\ 2)$; $B=(3\ 4)$ 则 `augment(A,B)=(1 2 3 4)`。

`stack(A,B,...)`: 矩阵合成函数, 不过它是将矩阵 (向量) A, B, \dots 按从上到下的顺序合成矩阵。

`Submatrix(A,ir,jr,ic,jc)`: 提出子矩阵函数, 表示一个由矩阵 A 的第 ir 行到 jr 行之间和第 ic 列到 jc 列之间的子矩阵。

`rows(M)`, `cols(M)`: 分别是求矩阵 M 行数和列数的函数。

`rank(M)`: 求矩阵 M 的秩的函数。

`tr(M)`: 求矩阵 M 的迹的函数。

`sort(V)`: 将向量 V 中所有的元素按从小到大的顺序进行排序输出一个新向量。

`rref(M)`: 对矩阵 M 进行单位化变换, 将 M 左边方阵范围部分变换成一个单位矩阵。(针对高斯消去法) 如果 M 是个方阵, 则得到一个单位矩阵, 如果 M 是个增广矩阵, 则变换后, 最右边一列则是方程组的解。

`identity(n)`: 生成一个单位矩阵, 行列数为 n 。

`diag(V)`: 生成一个对角矩阵, 对角元素按向量 V 元素的顺序排列。

`eigenvals(M)`: 求矩阵 M 的特征值。

`eigenvec(M,z)`: 求矩阵 M 对应于特征值 Z 的特征向量。

`eigenvecs(M)`: 求矩阵的特征向量。

52. 柱状图:

`hist(intvls,data)` 返回一个列向量, 数据 $data$ (矩阵) 所有元素中, 出现在 $intvls$ 上的频率。($intvls$ 是个整数, 就是把 $data$ 里所有的元素按从小到大的顺序, 均分成 $intvls$ 份; 向量的值就是按从小到大的顺序, 每一个等分区间里面元素的个数。)

$\text{histogram}(\text{intvls}, \text{data})$ 返回具有两列的一个矩阵，第一列包含等长的 intvls 的中点，其范围为 $\min(\text{data}) \leq \text{value} \leq \max(\text{data})$ 。第二列与 $\text{hist}(\text{intvls}, \text{data})$ 相同；结果矩阵具有 intvls 行。柱状图输出的第一列对于绘制柱状图的水平轴很有用。（返回一个两列的矩阵， data 里面的元素按从小到大顺序均分成了 intvls 份，有 intvls 个区间，以区间的中点的值代表这个区间，组成了矩阵第一列；第二列的值就是 $\text{hist}(\text{intvls}, \text{data})$ 返回的值，按从小到大的顺序，每个区间里面 data 分布元素的个数。）

$\text{mode}(\text{A}, \text{B}, \text{C}, \dots)$ 返回元素 A、B、C、... 中出现频率最高的数。

（下面的比较是自己边用的过程中写的，可能很乱，不系统。）

53. 在写函数的时候，如何在一堆语句中插入一条语句。Eg $p(n) := \begin{cases} i \leftarrow 0 \\ \text{while } n < i \\ m \leftarrow m \cdot (n-1) \\ m \end{cases}$,

我要给 m 赋个初值。插入在 $i \leftarrow 0$ 和 while 语句之间。Mathcad 中插入 add line 时，默认如果在一个语句中 add line 就是嵌套。所以在 $i \leftarrow 0$ 后按 add line 无法实现我们的目的。正确的做法是：先删掉“ \leftarrow ”，则第一句变得没有意义，这个时候点 add line 可以实现我们的目的。插入一条语句。

54. 利用“插入-组件-数据导入向导（一般是 excel 文档）”或者“插入-数据-数据导入向导”完成数据的输入；利用“插入-数据-文件输出（推荐输出为 txt 文本文档）”实现数据的输出。其实 mathcad 中输入输出的功能有很多，我比较支持利用上述的方法实现数据输入输出。另外，数据输出的时候，最好输出为 txt 文本，需要弄到 excel 里面去的话，自己在复制粘贴。选中输入输出文件范围，单击右键，属性，可以编辑修改值域，阈值、所选范围、输入输出路径等。