

张卫 主编

计算机 在 化学中的应用


东华大学出版社

计算机在化学中的应用

张 卫 主编

谢少艾 陈虹锦 副主编

东华大学出版社



图书在版编目(CIP)数据

计算机在化学中的应用/张卫主编. —上海:东华大学出版社, 2002. 12

ISBN: 7-81038-574-7

I. 计… II. 张… III. 计算机应用—化学 IV. 06-39

中国版本图书馆 CIP 数据核字(2003)第 006790 号

责任编辑:紫 仪

封面设计:旭 日

计算机在化学中的应用

张卫 主编

东华大学出版社出版

上海延安西路 1882 号 邮政编码:200051

新华书店上海发行所发行 常熟市大宏印刷有限公司印刷

开本:850×1168mm 1/32 印张:7.25 字数:171000

2002 年 12 月第 1 版 2002 年 12 月第 1 次印刷

印数:001-1000

ISBN 7-81038-574-7/TQ·02

定价:14.00 元

前 言

《计算机在化学中的应用》是化学类专业学生的基础课程之一，随着计算机技术的普及与发展，对学生计算机综合素质的培养十分重要。在学习了《计算机文化基础》、《程序设计与软件基础》等计算机基本知识的基础上，本课程旨在结合本专业特点，进一步提高学生的计算机理论与应用能力。

本教材的基本框架由三部分组成，第一部分为化学中常用的数值方法，第二部分是数学应用软件Mathcad入门及其编程基础以及部分化学应用软件的使用，第三部分是Internet上的化学资源的查找方法与应用。化学中常用的数值方法包括非线性方程求根、线性代数方程组的求解、实验观测数据的拟和及模型参数的确定、数值积分与微分方程的数值解等，同时列举了许多与化学学科相关的实例，力图达到学以致用目的。数学软件Mathcad由于其友好的界面和强大的计算和编程等功能，已经被广泛的应用在化学化工、物理学、电子工程学、机械工程等领域，本教材引入该软件，一方面利用其内置的M++编程语言来编写数值方法中相应的程序，另一方面利用其本身的函数解决实际问题，化学软件Chem Office提供了“Chem Draw”、“Chem 3D”和“Chem Finder”等组件，可以利用其绘制化学反应方程式、化学物质的三维立体结构、以及使用和创建Chem Finder化学数据库等，Origin软件在化学实验数据处理中有着最广泛的应用。同时，随着网络的普及与发展，如何更好的利用Internet上的化学资源也很重要，第三部分即是帮助学生掌握这一运用网络的技术。

本教材由张卫、谢少艾、陈虹锦、范勋培编写，上海交通大学化学化工学院的童晨骅同学为部分程序编写了程序框图，同时也得到了学院许多其他许多老师和同学的帮助，在此向他们表示衷心的感谢，在本教材的编写过程中，编者参考了已出版的相关教材，主要参考书目列于书后，在此说明并致谢。

限于水平和时间的仓促，书中有错误和不当之处，恳切希望读者批评指正。

编者

2002年12月于上海

目 录

第一章 MathCad 软件的使用基础	1
1.1 MathCad 2000 的功能与特点	1
1.2 MathCad 2000 的操作界面	3
1.3 MathCad 2000 的使用基础	3
1.4 使用 MathCad 编写程序	7
1.5 Mathcad 的解析计算	12
第二章 方程求根及其在化学平衡计算中的应用	18
2.1 二分法	18
2.2 迭代法	25
2.3 牛顿法	32
习题2	40
第三章 线性代数方程组的求解及其在分析化学中的应用	41
3.1 高斯(Gauss)-约当(Jordan)消去法	42
3.2 列主元高斯-约当消去法	49
3.3 全主元-约当高斯消去法	55
3.4 高斯-赛德尔迭代法	60
3.5 病态方程组和条件数	63
习题3	65
第四章 实验观测数据的拟合及模型参数的确定	66
4.1 一元线性拟合	67
4.2 多元线性拟合	75
4.3 化非线性拟合为线性拟合	82
4.4 非线性拟合	87
习题4	98
第五章 数值积分与微分方程的数值解	100
5.1 梯形法求积	101

第一章 MathCad 软件的使用基础

MathCad 软件是由 MathSoft 公司推出的一种交互式数学系统, MathCad 从 DOS 版本发展到今天的 MathCad 2000, 一直以它的高效、易用而得到广大用户的好评。MathCad 交互性很强, 他不像一般的编程语言那样晦涩难懂, 而是所见即所得。MathCad 的屏幕就好像一张电子便笺, 用户可以在其中进行公式的书写、推导、计算、数值显示等操作。同时, MathCad 的运算速度非常快, 用户改变任何一个数据、变量或数学表达式后, 其运算结果可以在瞬间做出相应的改变。

MathCad 是一个功能强大的工具软件, 它支持各种科学问题的分析、求解和显示技术。目前, 它已经被各门工程技术学科的科研人员广泛使用, 包括工程力学和热力学、数字信号处理、物理学和天文学、化学和化学工程、数学和统计学等。另外, MathCad 软件提供了一种内部编程语言 M++, 它用于编写用户自定义的函数, 这样就使得一些原来看起来是不可能实现的, 或者是用其他方法处理起来非常不方便的任务, 通过 MathCad 程序可以使这一切变得轻而易举。

1.1 MathCad 2000 的功能与特点

MathCad 具有独特的、功能强大地处理方程、数值、符号、文本以及图形的能力。

(1) 界面友好

MathCad 的使用操作十分简单, 可以按照自己的方式进行计算, 按照在稿纸上书写的方式, 写好公式、方程、图形数据或函数以及带文本的注释, MathCad 就会随即显示出计算结果或绘出图形。由于可以直接使用数学语言, 因而形象、直观。例如, 在编程语言中, 求二次代数方程的根的公式需要写成: $X = (-B \pm \text{SQRT}(B^2 - 4 * A * C)) / (2 * A)$ 而在 MathCad 中, 这一等式可以书写为:

$$x = \frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

(2) 强大的编程功能

内置的M++编程语言扩充了MathCad处理复杂问题的能力，可利用程序化的算子来生成功能程序；可定义局部变量、字符串变量；组合数据结构和嵌套数组。利用循环、递归、条件分枝以及return和continue语句来方便地编制出可以解决复杂问题的程序，并且提供On error语句处理运行时的错误。

(3) 优越的数学功能

能方便的解决求导、积分、求和以及三角函数、曲线函数和表达式的扩展、简化、因式分解以及曲线拟合等各种数学问题，

(4) 智能的可视化工具

可以把等式和数据作二维和三维图形，使等式和数据更加直观和形象化，并且MathCad的等式和图形处于活动状态，如果改变任何一个数据、变量或等式，就会立即对数学公式进行计算，并重新描绘图形。MathCad还可以创建和播放一些较短的动画。

(5) 全面支持 OLE2

如果用户想在技术报告中插入计算结果，只需要在MathCad中选中需要的部分，将其拖到目标文件中即可，无论是客户方还是服务方，MathCad都能全面支持OLE2，提供数据交换的拖放功能。

(6) MathCad 资源中心

提供了一个内容全面的MathCad电子书，其中包含了使用指南、举例和参考信息，简单的拖动其中的信息就可以放在MathCad工作表中使用。

1.2 MathCad 2000 的操作界面

MathCad 的工作界面是 Windows 95/98/NT 的常用用户界面，当启动 MathCad 时，可以看到以下几个部分：

- ◇ 无标题的 MathCad 工作表
- ◇ 资源中心窗口
- ◇ 菜单条
- ◇ 数学选项板
- ◇ 常用工具栏
- ◇ 格式栏

1.3 MathCad 2000 的使用基础

1.3.1 表达式、变量和函数

(1) 表达式的输入与编辑

表达式的创建是通过键入一串字符来实现的，这些字符包括字母和数字组成的各种变量、常量和函数的名字和数值，以及“+”和“-”等的算符。算符可以通过数学选项板中算术、矩阵、积分、程序等选项板中相应的按钮来输入。表达式的编辑也很简单，若是需要改变一个变量或函数的名字，以及插入、删除一个运算符或用另一个运算符来替代另一个运算符，只要将编辑光标移动至该处，用“Backspace”键或“Delete”键来删除字符，再键入相应的字符即可。

(2) 变量

若要定义一个变量或函数，首先输入变量名或函数名，然后输入“：”，再输入这个变量或函数所要赋予的值或表达式。变量名或函数名可以是任意长度、包含多种字符，可利用的字符如下：

- ◇ 大小写的字母

- ◇ 0- 9 十个数字
- ◇ 无穷大符号 (Ctrl+z)
- ◇ 希腊字母
- ◇ 下划线、百分号
- ◇ 下标

对变量名命名的制约如下：

- * 名字不能以数字、下划线、“'”和百分号为下标
- * 名字中所有字符必须具有相同的字体和风格
- * MathCad 不能区分变量名和函数名
- * MathCad 的内置常量，比如 π 、%、e、TOL=10-3...等不能用
- * MathCad 区分大小写
- * MathCad 区分不同字体的名字

(3) 函数

MathCad 提供了自定义函数的功能，同时还具有丰富的内置函数库，主要的函数库包括：

超越函数：基本的三角函数、指数函数、双曲函数等；

截断和舍入函数：这些函数从一个数值中取得某些信息，包括实部、虚部和取模等；

矩阵和向量函数：包括所有的矩阵、向量、数组的变换和运算；

排序函数：对向量和矩阵元素进行排序；

字符串函数：操作字符串，在字符串和数字、向量之间转换，创建自定义的错误信息；

离散变换函数：包括离散傅立叶变换函数和小波变换函数。

如果需要插入一个内置函数，在工作表的空白区域或在一个占位符上单击鼠标，选择“Insert”菜单下的“Function”命令，在打开的“Insert Function”对话框中选择所需要的函数，直接按下“Insert”按钮即可。

1.3.2 文本

文本区域在工作表中作为注释来使用，解释各个计算和程序的功能和结构。MathCad 中的文本可以包含各种字体、大小、风格和段落。可自定义文本区域的排列方式。

(1) 创建一个文本区域

利用工作表中的“Insert”菜单中的“Text Region”命令可在十字光标所在的位置创建一个文本区域，该文本区域的大小可调，且可以拖动至任意位置。

(2) 在文本中插入表达式

选择“Insert”菜单中的“Math Region”命令，就可以在文本区域中当前光标的位置上，插入一个数学区域，在该数学区域中可以输入数学表达式，文本中的数学表达式也一样会影响工作表中的计算结果，若只用作显示，则可以通过选中该表达式，选择“Format”菜单中的“Properties”命令，在“Calculation”标签中，选中其中的禁止选择框 (Disable Evaluatic)，从而文本中的该数学表达式将不会影响工作表中其他表达式的计算结果。

1.3.3 计算特点及计算的控制

MathCad 在表达式中发现一个错误，会把错误的运算符用另一种颜色（通常是红色）显示出来，同时还会出现一条错误信息，有错误的表达式是不能进行运算的。计算的控制分为自动模式和手动模式两种，自动模式下，计算结果将会自动更新，即工作表中的计算结果和图形在任何时候都是最新的。自动模式的特征是在状态条显示“Auto”字样，同时菜单“Math”中的“Automatic Calculation”旁边会显示一个选中的符号。手动模式下，可以手动的控制计算及计算结果的更新，如果只想更新能够看得见的表达式，只需要在菜单“Math”中选择“Calculate”命令，若向下移动工作表，计算结果还要再次手动更新，若是想要更新工作表中

的所有表达式，就需要使用菜单“Math”中的“Calculate Worksheet”命令了。处于手动模式的特征是在状态条不再显示“Auto”字样，同时菜单“Math”中的“Automatic Calculation”旁边也不会显示选中的符号。

1.3.4 向量和矩阵

在MathCad中单个的变量叫做标量，一系列变量叫做向量，一个矩形的多列变量叫做矩阵，向量和矩阵合称为数组。

(1) 创建向量和矩阵

创建数组可以使用菜单“insert”中的“matrix”命令，也可以使用数学选项板中的矩阵选项板，确定行、列数后，工作表中的光标处会出现矩阵算式，在占位符上输入各个元素的值，即成功的创建了一个数组，这种方法用于不是太大的数组。另外，当元素和数组之间存在一定的数学关系时，可以使用范围变量来填充数组的元素的方法来创建数组，如：

$$i := 0..20$$

$$j := 0..9$$

$$A_{i,j} := i + j$$

从而创建了一个21行10列的一个数组A

(2) 向量与矩阵常用运算符

- ◇ V_n - 返回指定的向量元素
- ◇ $A_{m,n}$ - 返回指定的矩阵元素
- ◇ $A^{>n}$ - 提取矩阵的第 $n+1$ 列，返回一个向量
- ◇ A^T - 把一个矩阵的行和列颠倒转置
- ◇ $|M|$ - 返回矩阵的行列式，返回一个标量
- ◇ M^{-1} - 对矩阵求逆，返回一个矩阵
- ◇ $\sum v$ - 对向量 V 求和，返回一个标量

(3) 常用矩阵和向量函数

rows(A)	返回数组 A 行数, 若 A 是一个标量, 则返回 0
cols(A)	返回数组 A 列数, 若 A 是一个标量, 则返回 0
length(V)	返回向量 V 的元素个数
last(v)	返回向量 V 的最后一个元素的索引值
max(A)	返回数组 A 中最大的元素
min(A)	返回数组 A 中最小的元素

(4) 字符串

虽然在 MathCad 中的许多场合用到的表达式和变量都是数字, 但也可以使用字符串构成的表达式和变量, 字符串包括任何能用键盘输入的字符, 包括字母、数字、标点和空格, 也包括各种特殊符号。字符串不同于变量和数字, MathCad 总是在双引号中显示它们。在 MathCad 中, 可以把一个字符串指定为变量名, 或是把一个字符串表达式设置成一个元素, 另外, 字符串不会限制大小, 并总在一行显示, 尤其需要注意的是一个字符串“123”不会被当作数字 123, 而仅仅是由一个字符组成的序列。

1.4 使用 MathCad 编写程序

利用 M++ 编写的程序具有一般程序设计语言的许多属性, 包括条件分枝、循环结构、局部变量、错误处理、迭代递归等, 并且每个 MathCad 程序都可以作为子程序被其他 MathCad 程序调用, 这样就极大的增强了 MathCad 的功能, 使得 MathCad 相对于其他高级语言的短处不复存在, 而 MathCad 相对于其他高级语言的长处就更加突出了。

1.4.1 定义一个程序

MathCad 程序是一个特殊的表达式, 如定义一个函数,

$f(x) = x^2 \cdot \ln(x) + \sin(x)$ 就可以用一个最简单的程序来表达:

$$f(x) := \begin{cases} t \leftarrow x^2 \cdot \ln(x) \\ w \leftarrow \sin(x) \\ t + w \end{cases}$$

MathCad 程序由多条语句组成，每条语句就是一个表达式，这些语句按照一定的顺序组合，同其他表达式一样，MathCad 程序也必须有一个值，由最后一条语句执行返回，可以是字符串、单个数字或一个数组，甚至可以是一个矩阵数据。

在 MathCad 程序中，变量可是设为局部变量和全局变量两种，局部变量的值只有在其所在的表达式或程序中有效，在此范围之外都没有定义，示例如下：

$$x := 25$$

$$\left| \begin{array}{l} x \leftarrow 12 \\ \sqrt{x+1} \\ x-1 \end{array} \right. = 0.328 \quad \text{在语句里 } x \text{ 是局部变量}$$

$$\frac{\sqrt{x+1}}{x-1} = 0.212 \quad \text{在程序之外 } x \text{ 是全局变量}$$

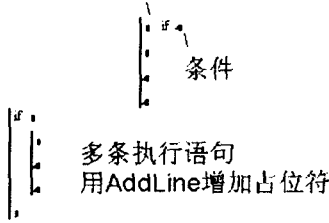
局部变量分配算子 \leftarrow 有两个占位符，左边的占位符只能输入一个变量名（不能是函数名、数组），右边的占位符中可以输入任何的表达式，但是不能是范围变量。

另外，当程序中有许多重复的子表达式时，可以把重复的子表达式定义为一个局部变量，以避免多次计算，造成 CPU 资源浪费。

1.4.2 条件语句

通常情况下，MathCad 从上往下执行程序中的每一个语句，若需要仅当某一个特殊条件出现时，才执行其中的一条语句时，可以通过在程序中使用“if”语句来实现条件的选择。“if”语句的格式如下：

执行的语句

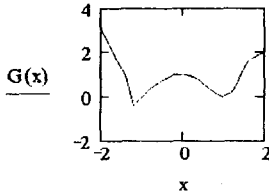


使用“if”语句可以定义分段函数，也可以同 otherwise 语句配合使用，建立多分支的选择语句，示例如下：

$$x := -2, -1.8..2$$

$$F(x) := x^2 - 1$$

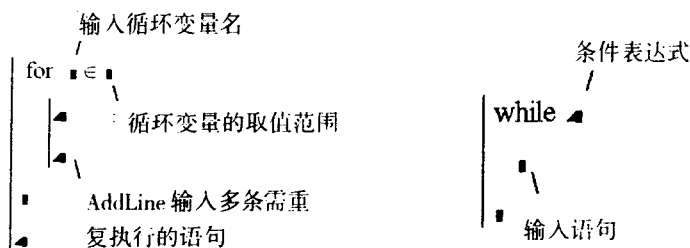
$$G(x) := \begin{cases} F(x) & \text{if } -2 \leq x \leq -1.3 \\ -F(x) & \text{if } -1.3 \leq x \leq 0 \\ (F(x))^2 & \text{if } 0 \leq x \leq 1.4 \\ x & \text{otherwise} \end{cases}$$



在使用“if”语句时需要注意的是：if 语句只能选择程序选项板上的 if 按钮，不能手工输入单词“if”；if 语句只能在程序中使用；otherwise 语句必须与 if 语句搭配使用。

1.4.3 循环语句

MathCad 提供了两种循环语句，若知道循环过程要具体执行多少次，可以使用 for 循环语句，若希望当某一条件出现时终止循环过程，而用户又不知道条件出现的具体时间时，可以使用 while 循环语句。示例如下：



for 循环支持多重嵌套，而且不会出现交叉嵌套；而 while 循环体的条件等于真的时候，while 循环才继续下去，在循环体或者程序中需要有一条语句能够修改 while 循环的条件，并且该条件最终会变成假。同 if 语句一样，循环语句也只能选择程序选项板上的 for 或 while 按钮，不能手工输入单词“for”或“while”。

1.4.4 程序执行的流程控制

MathCad 专业版的程序选项板中提供了 break、continue 和 return 三种语句用来控制程序执行的流程，三种语句功能如下：

break 语句：某特殊条件出现，break 语句终止程序的执行，若 break 语句在循环体中，则退出循环体，接着循环体后面语句执行；

continue 语句：终止当前循环，强迫程序执行下一次循环过程；

return 语句：停止程序执行，并返回一个值，程序不再执行后面语句；

若要插入控制程序语句，需要选择程序选项板中相应的 break、continue 或 return 按钮，而不能通过占位符输入关键字，并且控制程序语句只能用在程序中。

(1) break 语句：

当某一特殊条件出现时，使用 break 语句跳出一个程序或循环体是非常有用的，break 语句在程序中的使用方式有三种情况：

①在程序的主流程中使用

例如：

$a \leftarrow 10 = 10$ break $a \leftarrow 20$	$x \leftarrow 12 = 7$ $y \leftarrow 7$ $\text{break if } x = 12$ x	$a \leftarrow 10 = 20$ $\text{break if } a = 5$ $a \leftarrow 20$
---	---	---

一般情况下，在主程序中使用 break 语句比较少见。

② 在 for 或 while 循环体中使用

一般是通过一个语句来判断是否满足特殊条件，只有特殊条件满足了，才退出循环体，执行循环体后面紧接着的语句。值得注意的是 break 语句只能跳出 break 语句所在的循环体中。例如：

$$\text{fsum}(n) := \left| \begin{array}{l} \text{sum} \leftarrow 0 \\ \text{for } i \in 1..2000000 \\ \quad \left| \begin{array}{l} \text{sum} \leftarrow \text{sum} + i \\ \text{break if } i \geq n \end{array} \right. \\ \text{sum} \leftarrow 5 \cdot \text{sum} \end{array} \right.$$

$\text{fsum}(3) = 30$

③ 在程序子集中

程序子集其实也是一段完整的程序，只是它没有程序名而已，它也包括一条竖线，竖线右边有许多语句，最后一条语句作为程序子集的返回值，也可把程序子集的值赋给一个局部变量。若 break 语句不在循环体中，则 break 语句的执行也将终止整个程序的执行，并且返回该语句之前的计算结果。例如：

$x \leftarrow 3$ $\left \begin{array}{l} 1 \\ 2 \\ \text{break if } x > 0 \\ 4 \end{array} \right.$ $y \leftarrow x$ y	$= 0.5$	$x \leftarrow 3$ $\left \begin{array}{l} 1 \\ 2 \\ \text{break if } x < 0 \\ 4 \end{array} \right.$ $y \leftarrow x$ y	$= 81$
--	---------	--	--------