

21 世纪化学丛书

计算化学

郭纯孝 编著



Chemical Industry Press



化学工业出版社
化学与应用化学出版中心

华北水利水电学院图书馆



2010388474

O6-04

G787

21世纪化学丛书

计算化学

郭纯孝 编著



005/4/05



化学工业出版社

化学与应用化学出版中心

· 北京 ·

1038847

(京)新登字 039 号

图书在版编目(CIP)数据

计算化学/郭纯孝编著. —北京:化学工业出版社,
2004.3

(21 世纪化学丛书)

ISBN 7-5025-5235-9

I. 计… II. 郭… III. 计算化学 IV. 06-04

中国版本图书馆 CIP 数据核字(2004)第 016822 号

21 世纪化学丛书

计算化学

郭纯孝 编著

责任编辑:陈丽 刘俊之

责任校对:顾淑云

封面设计:蒋艳君

化学工业出版社 出版发行
化学与应用化学出版中心

(北京市朝阳区惠新里 3 号 邮政编码 100029)

发行电话 (010) 64989530

<http://www.cip.com.cn>

*

新华书店北京发行所经销

聚鑫印刷有限责任公司印刷

三河市延风装订厂装订

开本 720 毫米×1000 毫米 1/16 印张 24 $\frac{3}{4}$ 字数 447 千字

2004 年 4 月第 1 版 2004 年 4 月北京第 1 次印刷

ISBN 7-5025-5235-9/O · 41

定 价: 42.00 元

版权所有 违者必究

该书如有缺页、倒页、脱页者,本社发行部负责退换

序

化学作为一门基础学科，其专业门类日臻完善；学科内、外渗透融合更趋紧密；新型交叉学科层出不穷。当今化学的研究内容不断突破传统化学的范畴，涉及范围日益广泛；方法手段愈加先进。从应用层面看，化学已深入到国民经济和人民生活的各个方面，成为改善产业结构，推动经济发展，提高人民生活质量和满足社会多元化需要的重要因素。与此同时，化学领域的发展和突破使其凸显出核心科学的地位。

为了更好地把握发展机遇，适应科教兴国、科技兴国发展战略的要求，适应我国经济建设的需要，造就具有基础扎实、知识广博、富有创新精神的复合型人才，化学工业出版社计划组织出版一套《21世纪化学丛书》，并专门成立编委会，确定撰写这套丛书的宗旨和选题方向，并负责推荐各书撰写的专家。这套丛书内容包括材料化学、能源化学、催化化学、配位化学、纳米化学、胶体化学、界面化学、光化学、电化学、天然有机化学、药物化学、高分子化学、凝胶化学、计算化学、组合化学、环境化学、生物化学、食品化学等方面，着重介绍化学各分支学科领域的发展前沿，遵循学科继承、发展与创新的原则，突出反映其中的新知识、新成果、新应用和新趋势的“四新”内容，兼收并蓄，不拘一格，旨在体现化学科学的前瞻性、创见性、科学性和代表性以及学科间的交叉与渗透，力图使其成为一套内容丰富、体系完善、结构严谨、取材新颖的系列丛书。本丛书将由化学工业出版社陆续出版。

参与该丛书编撰的作者在各自的分支领域均具有丰富的科研积累与教学经验，他们创新的科学精神和认真的求实态度，将使该丛书各分册特色鲜明，各具风采。该丛书既介绍学科前沿的研究成果和进展，又反映学科的系统性和覆盖面，务使提高与普及兼顾，基础与实用并重。我们期望该丛书成为化学、化工、材料、环境、生命、医药卫生等相关领域的大专院校师生，科研院所的技术人员，政府和企业部门的管理人员以及其他各领域的化学爱好者有实用价值的综合参考书。

我们相信，通过编委和作者的共同努力，在广大科研和教学人员的积极参与和支持下，《21世纪化学丛书》的出版，将成为更多青年步入化学领域的桥

梁和阶梯，给广大读者带来有用的知识和有益的启示，为我国化学事业的发展起积极的推动作用。

由于编者的水平和时间所限，本丛书各册中不妥之处在所难免，敬请广大读者批评指正并积极建议，以使后续的分册越出越好。

编委会

2004 年 2 月

前 言

应《21 世纪化学丛书》主编陈洪渊院士和化学工业出版社的邀请编写了《计算化学》一书。

现代的化学研究中, 计算机已成为不可缺少的有力工具。学习使用计算机解决化学领域中遇到的各种数值计算, 已成为化学中不可缺少的部分。计算化学的领域非常丰富, 但作为基础还是数值计算的编制程序能力, 解决实际问题的基本技能。

本书采用当今世界上最流行的 C 程序设计语言。让读者进一步学会用 C 程序设计语言解决化学领域中的各种数值计算问题, 重点在于培养读者编制程序的能力。

主要内容包括: 数值计算的基本概念、计算误差、方程求根、函数插值、数值积分、常微分方程求解、线性方程组的求解、非线性拟合、一元和多元回归。

本书的后三章是属于专题科学研究性质, 是计算化学中的高级应用, 也是计算化学的新成果、新趋势。本书最大的特点是不仅给出数值计算方法, 而且给出了大量的科研上很实用的计算程序。

本书力图做到选材合适、概念准确、简明扼要、条理清楚、通俗易懂、着重于应用, 不过多地进行理论推导, 使本书成为宜教宜学的教材。给出算例较丰富, 书中程序均在 TurboC 2.0 编译通过。已经进行过实际应用, 具有可靠性和通用性。输入输出采用文件系统, 方便简单。每章后附有一定量的习题, 供研究生和科技工作者使用。

本书第 10 章由吉林大学张涛博士编写。第 11 章由吉林大学郭慧云教授编写。作者编制通用的正交试验极差、方差分析计算程序, 并统编全书。在本书的写作过程中, 作者受聘长春大学光华学院, 得到了康中集团董事长和长春大学光华学院董事长康井山先生和长春大学光华学院于福院长的大力资助, 在此表示衷心的感谢!

由于时间紧和作者水平所限, 在内容安排、材料选取上难免存在缺点和错误, 恳切希望读者予以批评指正。

郭纯孝

2004 年 2 月

内 容 提 要

本书是《21 世纪化学丛书》中的一册。

本书全面介绍了计算化学的基本内容, 包括: 计算机数值计算的基本概念、计算误差、方程求根、函数插值、数值积分、常微分方程求解、线性和非线性方程组的求解、非线性拟合、一元和多元回归等。同时还介绍了计算化学发展的前沿, 如正交试验法、多元统计校正理论、蒙特卡罗方法等。

本书选材合适、概念准确、条理清楚, 对于希望了解计算化学的高等院校师生、科研人员均具有参考价值。

目 录

第 1 章 绪论	1
1.1 微机在化学中的应用	1
1.2 计算机解题一般步骤	1
1.3 计算机语言和源程序	2
1.4 算法简介	2
1.5 数值计算误差简介	2
第 2 章 代数方程求根	4
2.1 引言	4
2.2 多项式的求值	5
2.3 二分法	9
2.4 迭代法	13
2.5 迭代过程的加速	17
2.6 牛顿法	21
2.7 弦截法	26
2.8 多项式方程求全部根	28
2.9 多元弱酸缓冲溶液的 pH 值和各组分浓度的确定	33
习题	41
第 3 章 函数插值	43
3.1 引言	43
3.2 线性插值	43
3.3 拉格郎日三点插值	47
3.4 拉格郎日 n 点插值	52
3.5 插值余项	57
3.6 埃特金插值	60
习题	65
第 4 章 数值积分	67
4.1 引言	67
4.2 梯形法求积	68
4.3 辛普生求积	72
4.4 求积公式的误差	79

4.5	离散点数据的求积	80
4.6	龙贝格求积	88
4.7	辛普生方法求二重积分	92
	习题	94
第 5 章	常微分方程的数值解	97
5.1	引言	97
5.2	欧拉法及改进	98
5.3	龙格-库塔法	102
5.4	积分步长的自动选取	108
5.5	一阶常微分方程组	110
5.6	高阶常微分方程的解	117
	习题	120
第 6 章	线性方程组的解法	122
6.1	引言	122
6.2	高斯消去法	123
6.3	迭代法	127
6.4	逆矩阵法求解线性方程组	142
6.5	三对角线性方程组的追赶解法	145
6.6	矩阵分解法求线性方程组	149
	习题	162
第 7 章	非线性方程组的解法	164
7.1	引言	164
7.2	迭代法	164
7.3	牛顿-雷扶生法	172
7.4	最速下降法	182
7.5	非线性函数参数的拟合	185
	习题	191
第 8 章	回归分析	194
8.1	引言	194
8.2	一元线性回归	195
8.3	加权回归	209
8.4	一元非线性回归	214
8.5	多元线性回归	221
8.6	数据的标准化	228

8.7	多元线性回归数据的标准化处理	230
8.8	多元线性回归的显著性检验	235
8.9	最优回归方程的选择	241
8.10	可化为多元线性回归的问题	243
8.11	多项式回归	247
	习题	253
	附表	255
第9章	蒙特卡罗方法	256
9.1	引言	256
9.2	蒙特卡罗方法的基本原理	256
9.3	MC 方法应用实例	258
9.4	随机数与伪随机数	262
9.5	MC 方法计算积分	267
9.6	MC 方法的综合应用	275
	习题	285
第10章	多元统计校正理论	286
10.1	化学量测数据的矩阵表示	286
10.2	多元线性回归校正	288
10.3	主成分回归	297
10.4	偏最小二乘法	306
第11章	正交试验法	315
11.1	引言	315
11.2	正交试验法的基础知识	316
11.3	正交试验结果的方差分析	327
11.4	考虑交互作用的正交试验	331
11.5	有混合水平的正交试验	340
11.6	正交试验结果的灵活处理	348
11.7	正交试验法的综合应用举例	357
11.8	正交试验的计算程序	361
	习题	371
	习题答案	372
	附表	373
附录		381
参考文献		384

第1章 绪 论

1.1 微机在化学中的应用

自1946年美国宾夕法尼亚大学发明第一台电子管计算机以来,近50年计算机经历了电子管、晶体管、集成电路和大规模集成电路四代。1980年第一代微机“Personal Computer”型出现以来,微机的发展速度非常迅速。目前常用微机的内存可达256M, CPU主频达到1.8GHz以上,硬盘存储已达80G。多媒体已经普遍应用。网络计算机的时代已经开始。

计算机的发展已深入到社会各个领域,引起了巨大变革。计算机在化学中的应用也日益广泛,更加深入。我国高等学校和科研单位的实验室中已经普遍配备了各种型号性能优良的微机。

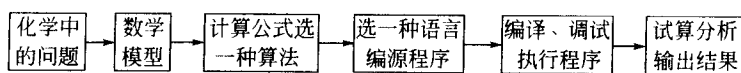
计算机在化学中的应用,归纳起来有以下几个方面:

- (1) 化学中的数值计算问题;
- (2) 化学中的红外、质谱、核磁、晶体结构,以及各学科试题等大型数据库的建立;
- (3) 化学中的文献、情报检索查询;
- (4) 化学中的图形学,显示化学中的二维及三维空间图形;
- (5) 化学中的专家系统,如有机化合物的解析、合成路线专家系统等;
- (6) 化学中的教学与辅导系统,如计算机的远程教学,以及化学各学科的辅导系统。

本书的主要目的是介绍化学中的数值计算问题,也就是用微机来解决化学中遇到的各种各样的数值计算问题。

1.2 计算机解题一般步骤

用计算机解化学中数值计算问题首先要把化学问题模型化,转变为数学公式,选择一种计算方法(包括算术运算和逻辑运算)。这种计算方法常称为算法。然后选择一种高级语言,把上述的算法书写成源程序。这项工作称为程序设计。再把源程序进行编程,调试得到执行程序。最后用已知的算例去试算检查。分析结果正确无误,才能用于未知的算例。归纳起来如下图所示:



1.3 计算机语言和源程序

目前在微机上使用的高级语言种类很多，适于科学计算的 language 有 FORTRAN, BASIC, PASCAL 以及 C 语言。本书使用目前流行的 C 程序设计语言编程。用全屏幕编辑软件(如 EDIT 或 QEDIT 等)，把计算公式书写成源程序，也称为源代码，使用 TurboC 2.0 编译系统编译，书中全部程序都在 586 微机上 TurboC 2.0 调试通过，形成 EXE 文件可以单独使用。

1.4 算法简介

算法是我们事前选定的计算方案和规划计算步骤。对同一个数值问题有很多算法可选用。首先要选择一种计算量尽量小、占用机器内存储量小、逻辑结构简单简单的算法。

在算法上尽量采用递推化公式。递推的基本思想是将一个复杂的问题简化为简单的过程多次重复。这种重复在程序上表现为循环，描述很容易。

描述算法方法有很多，其中使用框图方法可以形象描述算法。每个框表示要做的内容，用箭头表示各框执行的顺序。按照框图编制一个源程序更加容易。

1.5 数值计算误差简介

数值计算总是存在各种误差，计算结果总是近似的。误差是不可避免的，根本不存在绝对严格和精确的结果。但在数值计算中我们要求满足给定的精度总是可以实现的。

误差的来源首先决定你选用的数学模型。将化学问题归结为数学问题时，总要加上很多限制，总要忽略一些次要的因素，这样建立的理想化数学模型只是客观化学问题的一种近似。这种近似必然会产生误差，称为模型误差。描述问题的数学模型是近似的，所以要求解法的绝对准确是没有意义的。

在计算数值解中，常常存在对某种无穷过程的“截断”，由此产生的误差称为截断误差。

例如，计算指数函数 e^x 的值，常常用泰勒(Taylor)定理展开成幂级数形式：

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots$$

实际计算中，我们只能取有限项之和，而忽略其余各项，这时必然产生截断误差。根据泰勒的余项定理， e^x 的截断误差为：

$$R_n(x) = \frac{x^{n+1}}{(n+1)!} e^{\theta x}, \quad 0 < \theta < 1$$

假定 $|x| \leq 1$ ，要求截断误差小于 5×10^{-3} ，注意到 $e^{\theta x} \leq e \leq 3$ ，

$$|R_n(x)| < \frac{3}{(n+1)!}$$

于是，只要 $n=5$ 即满足要求。

数值计算中，另一种误差为舍入误差。受机器字长的限制，用机器代码表示数据的位数有一定限制，由此产生的误差称为舍入误差。少量的运算产生的舍入误差是微不足道的，但千百万次的舍入误差有时是很惊人的。在微机数值计算中的实数尽量采用双精度字长，以减少舍入误差。总之，对任何一项计算，首先必须保证精度。在满足给定的精度下，选择一种好的算法。

第2章 代数方程求根

2.1 引言

在化学问题中常常遇到代数方程的求根问题。这些问题有时是复杂化学问题的组成部分，有时自身是一个完整的问题。

例 2-1 化学中真实气体服从范德华(van der Waals)方程

$$\left(p + \frac{a}{V^2}\right)(V - b) = RT$$

其中： a ， b 是常数； R 是气体常数； p ， V ， T 三个变量的关系是一个三次方程。已知 p ， T ，求它的体积 V 。

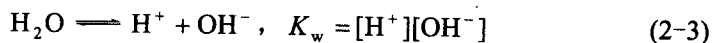
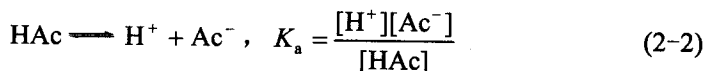
原方程变为

$$pV^3 - (pb + RT)V^2 + aV - ab = 0 \quad (2-1)$$

这是一个求三次方程根的问题。

例 2-2 在分析化学中，已知醋酸 HAc 的浓度为 C ，求溶液的 pH 值。

在 HAc 稀溶液中存在



溶液中 HAc 浓度 $C_{\text{HAc}} = [\text{HAc}] + [\text{Ac}^-]$

由电中性原理 $[\text{H}^+] = [\text{Ac}^-] + [\text{OH}^-]$

把 $[\text{Ac}^-]$ ， $[\text{HAc}]$ 代入(2-2)式，有

$$K_a = \frac{[\text{H}^+]([\text{H}^+] - [\text{OH}^-])}{C_{\text{HAc}} - [\text{H}^+] + [\text{OH}^-]} \quad (2-4)$$

再把 $[\text{OH}^-] = K_w / [\text{H}^+]$ 代入(2-4)展开变为 $[\text{H}^+]$ 的三次方程

$$[\text{H}^+]^3 + K_a[\text{H}^+]^2 - (K_a C_{\text{HAc}} + K_w)[\text{H}^+] - K_a K_w = 0 \quad (2-5)$$

把 K_a ， K_w ， C_{HAc} 代入可得出 $[\text{H}^+]$ 的浓度。溶液的 $\text{pH} = -\lg[\text{H}^+]$ 。

例 2-3 量子化学计算共轭体系的 π 电子能级。

根据 HMO 的基本假定，令

$$x = \frac{\alpha - E}{\beta}$$

其中： α 称为库仑积分， β 称为共振积分， E 为 π 电子能级。

求解下面久期行列式的根得到丁二烯的 π 电子能级。

$$\begin{vmatrix} x & 1 & 0 & 0 \\ 1 & x & 1 & 0 \\ 0 & 1 & x & 1 \\ 0 & 0 & 1 & x \end{vmatrix} = x^4 - 3x^2 + 1 = 0 \quad (2-6)$$

也就变为求解一个多项式的代数方程。

关于代数方程求根问题类型的分类，首先是根据方程的个数，其次是解的个数。代数方程求根的分类如图 2-1 所示。

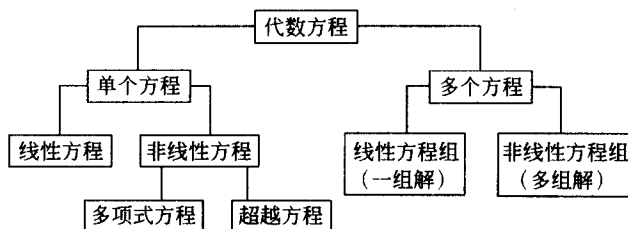


图 2-1

关于多个方程中线性方程组的解在第 6 章讨论，非线性方程组的解在第 7 章讨论，本章主要讨论单个方程的求解问题。

一元非线性方程又分为多项式方程和超越方程。含有 $\sin x$, $\cos x$, $\tan x$ 等三角函数或者含有 $\lg x$, e^x 等特殊函数的非线性方程统称为超越方程。

一元非线性方程求解分直接法和间接法。所谓的直接法是用一组公式得到解，如二次方程的求根。间接法是重复使用某种算法求解，得到的解是近似的，但可以达到任意要求的精度。迭代法是最适宜于计算机的间接解法。

2.2 多项式的求值

形如

$$f(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n = 0 \quad (2-7)$$

称为一元 n 次代数方程， $f(x)$ 称为一元 n 次多项式。

如果我们直接计算多项式的值，使用乘法的次数为 $N + N - 1 + \cdots + 1 = \frac{N(N-1)}{2}$ ，加法的次数为 N 次。如果把上式用代数结合律改写为

$$f(x) = ((\cdots (a_0 \cdot x + a_1) \cdot x + a_2) \cdot x + \cdots + a_{n-1}) \cdot x + a_n \quad (2-8)$$

计算上式的值仅用 N 次乘法和 N 次加法。若 $N = 10$ ，第一种方法用 45 次乘法

和 10 次加法，而第二种方法仅用 10 次乘法和 10 次加法。

多项式的这种有效算法是我国宋代数学家秦九韶最先提出的，外国文献中称霍纳(Horner)算法。霍纳工作较秦九韶工作晚五六世纪。

秦九韶算法特点是通过一次式反复计算逐步得到高次多项式的值。用一个递推公式表示

$$\left. \begin{array}{l} (1) v_0 = a_0 \\ (2) v_i = v_{i-1} \cdot x + a_i \\ (3) f(x) = v_n \end{array} \right\} \quad (i=1,2,\dots,n) \quad (2-9)$$

n 次多项式的系数有 $n+1$ 个，存放在 $a[n+1]$, x 为给定值，求 $f(x)$ 值的 $\text{poly}(a, m, x)$ 函数框图如图 2-2。 $\text{poly}(a, m, x)$ 函数参数说明: $a[]$ 存放多项式的 $n+1$ 个系数, m : 多项式系数的个数: x : x 设定值。

求多项式及一阶导数值的程序如下:

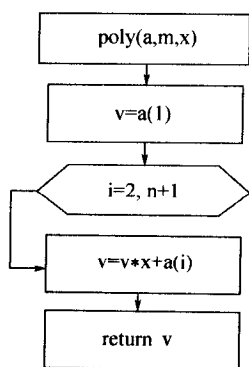


图 2-2

```

/* po.c */
main()
{
    int i,n;float x,p,a[20];
    float poly(float a[],int m,float x);
    printf( "\nInput n x : ");
    scanf( "%d%f", &n, &x);
    printf( "\nPlease input (a[i]i=1,n+1: ");
    for(i=1;i <=n+1;i++)
        scanf( "%f", &a[i]);
    p=poly(a,n+1,x);
    printf( "\n f(x)    =%12.6f\n",p);
    for(i=1;i<=n;i++)
        a[i]*=(n-i+1);
    p=poly(a,n,x);
    printf( " df(x)/dx   =%12.6f\n",p);
    for(i=1;i<=n-1;i++)
        a[i]*=(n-i);
    p=poly(a,n-1,x);
    printf( " df(x) ^ 2/dx ^ 2  =%12.6f\n",p);
    for(i=1;i<=n-2;i++)
        a[i]*=(n-i-1);
    p=poly(a,n-2,x);
    printf( " df(x) ^ 3/dx ^ 3   =%12.6f\n",p);
}
    
```

```

    }
float poly(float a[],int m,float x)
{
    int i;float v;
    v=a[1];
    for(i=2;i<=m;i++)
        v=v*x+a[i];
    return(v);
}

```

计算下面多项式在 $x=2$ 时的值及一阶、二阶、三阶导数值。

$y=5x^5+4x^4+3x^3+2x^2+1x+2.0$
 $y=? \quad y'=? \quad y''=? \quad y'''=?$

运行结果如下:

```

Input n x: 52.✓
Please input (a[i]i=1, n+1: 5.4.3.2.1.2.✓
f(x)                = 260.000000
df(x)/dx             = 573.000000
df(x)^2/dx^2         = 1032.000000
df(x)^3/dx^3         = 1410.000000

```

如果采用文件形式输入和输出, 多项式求值以及求一阶、二阶、三阶导数值的程序和计算例子如下:

```

/* pol.c */
#include <stdio.h>
main(int argc,char*argv[])
{
    FILE*fp1,*fp2;int i,n;float a[20],x,p;
    float poly(float a[],int m,float x);
    if(argc <3)
    {puts( "\nUsage:PO1 input_data_filename output_filename\n ");
      exit(1);}
    if((fp1=fopen(argv[1], "r "))==0){
        printf( "can't open data file\n ");exit(1);}
    if((fp2=fopen(argv[2], "w "))==0){
        printf( "can't open output file\n ");exit(1);}
    fscanf(fp1, "%d%f\n ", &n,&x);
    for (i=1;i<=n+1;i++)
        fscanf(fp1, "%f ", &a[i]);
    fclose(fp1);
}

```