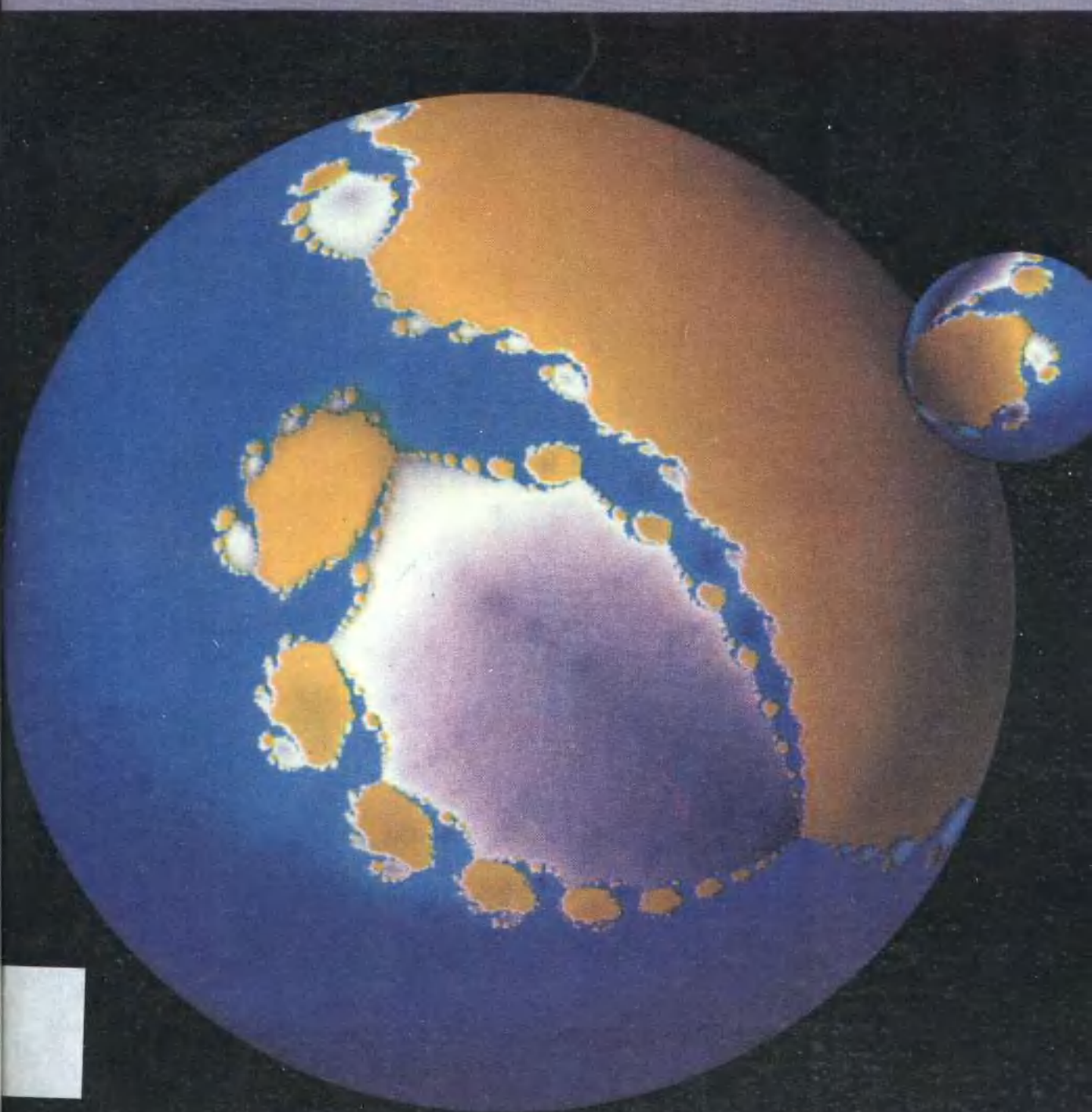


# 计算物理学

[美] Steven E. Koonin 著  
秦克诚 译

高等教育出版社



C411.2

196

K58

# 计算物理学

[美] Steven E. Koonin 著

秦克诚 译



A0919525

高等教育出版社

(京) 112号

## 内 容 简 介

计算物理学是物理学的新的极为重要的分支,它与理论物理和实验物理一起构成现代物理学的整体。目前,计算物理学作为一门大学本科及研究生物理课程已经成熟。1986年在美出版的 Steven E. Koonin 所著《计算物理学》,获得了有关专家的一致好评,并被国外不少著名大学采用作为计算物理课的教材。

本书全面介绍了计算物理中各种常用的计算方法,配置了内容新颖而丰富的遍及现代物理学各分支的例题和课题,书后附有全部例题和课题的程序。程序采用 IBM PC BASIC 语言,具有很强的通用性。

本书可作为高等学校物理专业高年级学生及研究生学习计算物理时的教材,对于广大物理教师和物理学工作者也很有参考价值。

### Computational Physics

Steven E. Koonin

©高等教育出版社,1993。

本书经 Addison-Wesley Publishing Company, Inc.

授权将 ©1986 英文版翻译成中文并在中国出版、销售。

### 计算物理学

[美]Steven E. Koonin 著

秦克诚 译

\*

高等教育出版社出版  
高等教育出版社激光照排技术部照排  
新华书店北京发行所发行  
高等教育出版社印刷厂印装

\*

开本 787×1092 1/16 印张 20.75 字数 490 000

1992年11月第1版 1992年11月第1次印刷

印数 0 001—2 242

ISBN7-04-002748-7/O·872

定价 11.20 元

# 第一章 基本数学运算

在绝大多数用计算机模拟物理系统的问题中,有三种数值运算是非常重要的,那就是微分、求积和求根. 设我们能够计算一个函数 $f(x)$ 在自变量 $x$ 的任何值上的函数值. 微分就是要在给定的 $x$ 值上求 $f$ 的某一阶导数. 求积粗略地说是微分的逆运算,它要求我们计算 $f$ 在两个确定的积分限之间的定积分(我们对“求积”和“积分”两个术语加以区分,把“积分”这个术语保留给第二章讨论的解微分方程的过程). 而求根则是求使 $f$ 为零的 $x$ 值(可能不只一个).

如果 $f$ 的解析形式是已知的,那么几乎总可以(只要有足够的毅力)为 $f$ 的导数推出显式的公式,对 $f$ 的定积分也常常可以推出显式公式. 但是,情况常常是不能用解析方法,尽管我们能够计算 $f(x)$ 本身的值. 这可能是由于计算 $f$ 需要用一些非常复杂的数值过程,我们没有能够对它应用微分规则和求积规则的适当的解析公式;或者更不好办的是,我们用以产生 $f$ 的方法只向我们提供 $f$ 在一组分立的坐标值上的数值. 在这两种情况下,我们都必须使用近似公式,通过我们能够计算的 $f$ 值来表示导数和积分. 此外,除了最简单的函数之外,所有函数的根都不能用解析方法求出,因而必须用数值方法.

本章讨论如何在计算机上实现这三种基本运算. 基本方法是用一个简单的函数(例如一次或二次多项式)来逼近 $f$ ,对这个简单函数很容易实行这三种运算. 我们将只推导最简单和最常用的公式;进一步的讨论可以在许多数值分析教科书上找到.

## 1.1 数值微分

设我们想要求 $x=0$ 处的导数 $f'(0)$ . (通过平移我们推导的公式可以很容易地推广到任意的 $x$ 值.) 假定我们知道 $f$ 在一组等间隔的 $x$ 值格点上的数值:

$f_n = f(x_n); x_n = nh (n=0, \pm 1, \pm 2, \dots)$ ,  
我们的目标是要通过 $f_n$ 算出 $f'(0)$ 的一个近似值(见图 1.1).

首先我们用 Taylor 级数在 $x=0$ 的邻域展开 $f$ :

$$f(x) = f_0 + xf' + \frac{x^2}{2!} f'' + \frac{x^3}{3!} f''' + \dots, \quad (1.1)$$

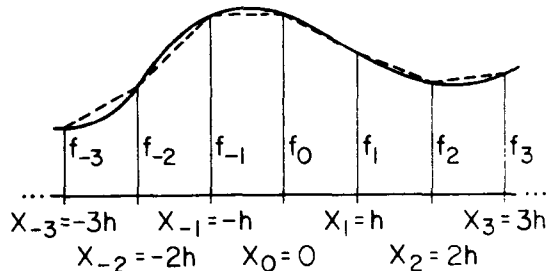


图 1.1 等间隔格点上的 $f$ 值. 虚线表示线性内插.

其中所有各阶导数都取  $x=0$  处之值. 容易验证

$$f_{\pm 1} \equiv f(x=\pm h) = f_0 \pm hf' + \frac{h^2}{2} f'' \pm \frac{h^3}{6} f''' + O(h^4), \quad (1.2a)$$

$$f_{\pm 2} \equiv f(x=\pm 2h) = f_0 \pm 2hf' + 2h^2 f'' \pm \frac{4}{3} h^3 f''' + O(h^4), \quad (1.2b)$$

其中  $O(h^4)$  表示与  $h^4$  同阶或更高阶的一些项. 为了估计这些项的大小, 我们可以假定  $f$  及其各阶导数的大小都是同一数量级, 许多物理学中用到的函数属于这种情况.

从(1.2a)式给出的  $f_1$  减去  $f_{-1}$ , 重新集项, 得

$$f' = \frac{f_1 - f_{-1}}{2h} - \frac{h^2}{6} f'' + O(h^4). \quad (1.3a)$$

含有  $f''$  的项当  $h$  很小时趋于零, 它是只保留第一项所得到的有限差分近似

$$f' \approx \frac{f_1 - f_{-1}}{2h} \quad (1.3b)$$

的主要误差来源. 若  $f$  在这个3点区间  $[-h, h]$  上是一个二次多项式, 那么这个“3点公式”将是精确的, 因为三阶导数和一切高阶导数都变为零. 因此, (1.3b)式的实质是假定函数  $f$  通过三点  $x = \pm h, 0$  的二次多项式内插成立.

回忆初等微积分中定义导数的公式, (1.3b)式是很自然的结果. 通过使  $h$  值越来越小, 原则上可以使和  $h^2$  同阶的误差项任意小. 还要注意, 这里用的是  $x=0$  两侧的对称差分, 它比下面两个前向差分和后向差分公式更精确(相差  $h$  的一阶):

$$f' \approx \frac{f_1 - f_0}{h} + O(h); \quad (1.4a)$$

$$f' \approx \frac{f_0 - f_{-1}}{h} + O(h). \quad (1.4b)$$

这两个“2点”公式的基础是, 假设  $f$  在  $x=0$  和  $x=\pm h$  之间的区间上可以用一个线性函数很好地逼近.

作为一个具体例子, 我们来计算  $f(x) = \sin x$  的  $f'(x=1)$ . 精确的答案当然是  $\cos 1 = 0.540302$ . 下述 BASIC 程序对输入的  $h$  值计算(1.3b)式在这种情形下之值:

```
10 X=1: EXACT=COS(X)
20 INPUT "Enter value of h (<=0 to stop)":H
30 IF H<=0 THEN STOP
40 FPRIME=(SIN(X+H)-SIN(X-H))/(2*H)
50 DIFF=EXACT-FPRIME
60 PRINT USING "h=#####, ERROR=+#####";H,DIFF
70 GOTO 20
```

(如果你是 BASIC 的初学者, 请注意以下几点: 系统是怎样请求从键盘送入  $H$  值的; 若送入非正的  $H$  值程序就会停止; 变量名怎样根据其意义自然地选取<sup>①</sup>; 40 行中用 SIN 函数重写了数

<sup>①</sup> 这一段程序所用的几个变量名中, EXACT 的英文意义是精确值, FPRIME 代表  $F'$ , DIFF 是 DIFFERENCE(差)的缩写. ——译者.

学公式(1.3b); 60行中在要把结果输出到屏幕时如何规定有效数字位数; 以及70行中的转移控制)。

用这个程序所产生的结果, 以及用前向和后向差分公式(1.4a, b)算得的类似结果, 通通列在表1.1中。注意结果的精度随着我们减小 $h$ 而改进, 但只到某一 $h$ 值为止, 此后结果反而更差。这是因为计算机中的算术运算只有有限的精度(对于BASIC中的单精度变量, 其精度为十进制数5—6位), 因此求得的近似公式分子中二项之差, 当 $h$ 很小因而 $f_1$ 和 $f_{-1}$ 相差很小时, 就会发生很大的“舍入”误差。例如, 若 $h=10^{-6}$ , 则

$$f_1 = \sin(1.000001) = 0.841472; f_{-1} = \sin(0.999999) = 0.841470,$$

因此, 若取六位有效数字, 则 $f_1 - f_{-1} = 0.000002$ 。代入(1.3b)后得 $f' \approx 1.000000$ , 这个结果很不好。但是, 如果我们取10位有效数字进行算术运算, 那么

$$f_1 = 0.8414715251; f_{-1} = 0.8414704445,$$

根据这些数据从(1.3b)式算出 $f' \approx 0.540300$ , 这个结果还说得过去。从这个意义上说, 数值微分是一个具有固有不稳定性的过程( $h \rightarrow 0$ 时没有确定的极限), 因此在实行时必须小心。

表 1.1 计算 $d \sin x / dx|_{x=1} = 0.540302$ 的误差

$h$	对称3点 (1.3b)式	前向2点 (1.4a)式	后向2点 (1.4b)式	对称5点 (1.5)式
0.50000	0.022233	0.228254	-0.183789	0.001092
0.20000	0.003595	0.087461	-0.080272	0.000028
0.10000	0.000899	0.042938	-0.041139	0.000001
0.05000	0.000225	0.021258	-0.020808	0.000000
0.02000	0.000037	0.008453	-0.008380	0.000001
0.01000	0.000010	0.004224	-0.004204	0.000002
0.00500	0.000010	0.002108	-0.002088	0.000006
0.00200	-0.000014	0.000820	-0.000848	-0.000017
0.00100	-0.000014	0.000403	-0.000431	-0.000019
0.00050	0.000105	0.000403	-0.000193	0.000115
0.00020	-0.000163	-0.000014	-0.000312	-0.000188
0.00010	-0.000312	-0.000312	-0.000312	-0.000411
0.00005	0.000284	0.001476	-0.000908	0.000681
0.00002	0.000880	0.000880	0.000880	0.000873
0.00001	0.000880	0.003860	-0.002100	0.000880

通过把 $f'$ 同离 $x=0$ 更远的格点联系起来, 可以对3点公式(1.3b)作改进。例如, 应用(1.2)式容易证明, 下述“5点”公式:

$$f' \approx \frac{1}{12h} [f_{-2} - 8f_{-1} + 8f_1 - f_2] + O(h^4) \quad (1.5)$$

消去了Taylor级数中直到四阶的各阶导数。用这种方法计算导数实质上是假定在5点区间 $[-2h, 2h]$ 上 $f$ 被一个四次多项式很好地逼近。它虽然需要更多的计算量, 但得到的近似也

精确得多, 这由表 1.1 可以看出. 实际上, 用(1.5)式以大约大 10 倍的步长就可以得到与(1.3b)式相当的精度. 这在必须把多个  $f$  值储存在计算机内时可能是重要的, 因为更高的精度就允许  $f$  排得更稀疏一些, 从而节省存储空间. 但是, 因为(1.5)式要比(1.3b)式进行更多的数学运算, 并且各项(既有正系数又有负系数)之间有很大程度的相互抵消, 精度问题在  $h$  值更大时就显露出来.

取(1.2)式的适当组合可以构造出更高阶导数的公式. 例如, 容易看出

$$f_1 - 2f_0 + f_{-1} = h^2 f'' + O(h^4), \quad (1.6)$$

因此, 准确到  $h^2$  量级的二阶导数的近似公式为

$$f'' \approx \frac{f_1 - 2f_0 + f_{-1}}{h^2} \quad (1.7)$$

准确到  $h$  的更高次的  $f$  的各阶导数的差分公式可以直接推出. 表 1.2 汇集了一些 4 点和 5 点表达式.

表 1.2 各阶导数的 4 点和 5 点差分公式

	4 点	5 点
$hf'$	$\pm \frac{1}{6} (-2f_{\mp 1} - 3f_0 + 6f_{\pm 1} - f_{\pm 2})$	$\frac{1}{12} (f_{-2} - 8f_{-1} + 8f_1 - f_2)$
$h^2 f''$	$f_{-1} - 2f_0 + f_1$	$\frac{1}{12} (-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2)$
$h^3 f'''$	$\pm (-f_{\mp 1} + 3f_0 - 3f_{\pm 1} + f_{\pm 2})$	$\frac{1}{2} (-f_{-2} + 2f_{-1} - 2f_1 + f_2)$
$h^4 f^{(iv)}$		$f_{-2} - 4f_{-1} + 6f_0 - 4f_1 + f_2$

习题 1.1 用任何一个能够以解析方法算出各阶导数的函数, 考察表 1.2 中各个公式在不同的  $h$  值下的精度.

## 1.2 数值求积

求积问题是求  $f$  在两个积分限  $a < b$  之间的定积分. 我们很容易把  $a, b$  两点之间的区间分为偶数个间隔, 即

$$N = \frac{(b-a)}{h}$$

为偶数. 于是只要推出从  $-h$  到  $+h$  的积分公式就够了, 因为这个公式累用多次就得到

$$\int_a^b f(x) dx = \int_a^{a+2h} f(x) dx + \int_{a+2h}^{a+4h} f(x) dx + \cdots + \int_{b-2h}^b f(x) dx \quad (1.8)$$

我们下面要讨论的一切求积公式(所谓闭合的 Newton-Cotes 型求积公式)的基本思想, 是用一个可以精确求积的函数在  $-h$  和  $+h$  之间逼近  $f$ . 例如, 分别考虑区间  $[-h, 0]$  和  $[0, h]$ , 并且

假设  $f$  在每一个区间里是线性的(见图1.1).这种内插造成的误差之量级为  $h^2 f''$ , 因此近似积分值为

$$\int_{-h}^h f(x)dx = \frac{h}{2} (f_{-1} + 2f_0 + f_1) + O(h^3), \quad (1.9)$$

这就是人们熟知的梯形法则.

Taylor级数(1.1)可以给出对  $f$  的一个更好的内插, 从而得到一个更好的近似. 应用关于  $f'$  和  $f''$  的差分公式(1.3b)和(1.7), 对  $|x| < h$  可得

$$f(x) = f_0 + \frac{f_1 - f_{-1}}{2h} x + \frac{f_1 - 2f_0 + f_{-1}}{2h^2} x^2 + O(x^3), \quad (1.10)$$

对它积分得出

$$\int_{-h}^h f(x)dx = \frac{h}{3} (f_1 + 4f_0 + f_{-1}) + O(h^5). \quad (1.11)$$

这是 Simpson 法则, 可以看出, 它的精度要比梯形法则高两阶. 注意, 实际误差要比简单地根据(1.10)式预期的误差小, 因为  $x^3$  项对积分没有贡献. 按照(1.8)式累用这个公式, 得

$$\int_a^b f(x)dx = \frac{h}{3} [f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \dots + 4f(b-h) + f(b)] \quad (1.12)$$

作为一个例子, 下面这个 BASIC 程序<sup>①</sup>应用 Simpson 法则根据输入的  $N = 1/h$  值计算

$$\int_0^1 e^x dx = e - 1 = 1.718282.$$

(这种插在正文中的程序的源代码不包含在书后所附的软盘上, 但是很容易用键盘送进读者的计算机.)

```

5 DEF FNF(X)=EXP(X)           '被积函数
10 EXACT=EXP(1)-1
15 INPUT "Enter N (even,>=2)",N%
20 IF N%<2 THEN STOP
25 '
30 H=1/N%
35 SUM=FNF(0)                 'X=0 的贡献
40 FAC=2                       'Simpson 法则中的因子
45 '
50 FOR I%=1 TO N%-1           '对格点进行循环
55 IF FAC=2 THEN FAC=4 ELSE FAC=2 '改换因子
60 X=I%*H                     '该点的 X 值
65 SUM=SUM+FNF(X)*FAC        '对积分的贡献
70 NEXT I%
75 '
80 SUM=SUM+FNF(1)             'X=1 的贡献
85 INTEGRAL=H*SUM/3
90 DIFF=EXACT-INTEGRAL
95 PRINT USING "N=#### ERROR=#.####";N%,DIFF
100 GOTO 15                   '取另一 N% 值
    
```

① 为了方便读者读懂程序, 我们把正文中程序内的注解都译成中文. 将这些源程序送入计算机时, 这些注解不必送入. ——译者注.



这个程序对不同的  $N$  值的运行结果，连同应用梯形法则所得到的数值，都列在表 1.3 中。高阶公式带来的改进是明显的。注意结果是稳定的，意即随着  $N$  变得很大和格子间隔  $h$  变小会得到一个确定的极限。舍入误差在这个例子中不重要，因为所有的  $f$  值都以同样的符号进入求积公式，这同数值微分的例子中的情况相反。

表 1.3 计算  $\int_0^1 e^x dx = 1.718282$  的误差

$N$	$h$	梯形法则 (1.9)式	Simpson 法则 (1.12)式	Bode 法则 (1.13b)式
4	0.250000	-0.008940	-0.000037	-0.000001
8	0.125000	-0.002237	0.000002	0.000000
16	0.062500	-0.000559	0.000000	0.000000
32	0.031250	-0.000140	0.000000	0.000000
64	0.015625	-0.000035	0.000000	0.000000
128	0.0078125	-0.000008	0.000000	0.000000

求积时一个重要的问题是，为了把积分计算到给定的精度， $h$  必须小到什么程度。虽然对我们讨论过的公式能够推出严格的误差限，但实际中最简单的做法是，以越来越小的  $h$  值重复进行计算，而观察结果的变化。

在 Taylor 展开式中保留更多的项数以对格点之间的  $f$  值进行内插，同时对各阶导数相应采用更好的近似，可以导出更高阶的数值求积公式。用三次多项式和四次多项式进行内插所得出的对 Simpson 法则的推广分别叫做 Simpson  $\frac{3}{8}$  法则和 Bode 法则，它们是：

$$\int_{x_0}^{x_3} f(x) dx = \frac{3h}{8} [f_0 + 3f_1 + 3f_2 + f_3] + O(h^5); \quad (1.13a)$$

$$\int_{x_0}^{x_4} f(x) dx = \frac{2h}{45} [7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4] + O(h^7). \quad (1.13b)$$

应用 Bode 法则的结果也列在表 1.3 中，它带来的改进是明显的，虽然是以更复杂的计算为代价。（注意：为了能够用这个方法， $N$  必须是 4 的倍数。）人们也许会以为，用次数很高的多项式作内插所得出的求积公式将会更为合用，但实际情况并非如此。这样的多项式容易发生激烈的振荡，从而导致不准确的内插。此外，在高阶公式中，不同格点上的  $f$  值前面的系数既可能带正号也可能带负号，这使舍入误差成为一个潜在的问题。因此，通常比较保险的改进精度的方法，是使用低阶方法并使  $h$  值小一些，而不是用高阶方法。如果我们放弃坐标格点等间隔的要求，可以导出精度达到很高阶的求积公式，这些将在第四章中讨论。

习题 1.2 用任何一个能够用解析方法计算其定积分的函数，考察上面讨论的各种求积方法在不同  $h$  值下的精度。

在应用上面讨论的数值求积公式时，必须注意若干事项并且具备某些常识。例如，上限很大的积分最好是通过一个变量替换来处理。比如用 Simpson 法则计算

$$\int_1^b dx \ x^{-2}g(x),$$

其中  $g(x)$  在  $x$  值很大时是常数，这个积分将导至一个有限和，它在  $b$  变大但  $h$  保持固定时收敛得很慢（因而要用很长的时间计算！）。但是，将积分变量换为  $t = x^{-1}$ ，则积分变为

$$\int_{b^{-1}}^1 g(t^{-1})dt,$$

它可以用我们讨论过的任何一个公式计算。

使简朴的求积公式失去意义的可积奇点也可以用下述简单方法处理。例如，

$$\int_0^1 dx (1-x^2)^{-1/2} g(x)$$

在  $x=1$  有一个可积奇点（如果  $g$  在该点正则），积分值是一个有限数。但是，由于  $f(x=1) = \infty$ ，上面讨论的各个求积公式给出一个无穷大结果。把积分变量换为  $t = (1-x)^{1/2}$  可以得到准确的结果，这时积分变为

$$2 \int_0^1 dt (2-t^2)^{-1/2} g(1-t^2),$$

它可以用以上的求积公式逼近而不会遇到麻烦了。

可积奇点也可以通过推导出专门对它们适用的公式来处理。设我们要计算

$$\int_0^1 f(x)dx = \int_0^h f(x)dx + \int_h^1 f(x)dx,$$

其中  $f(x)$  在  $x=0$  附近的行为近似于  $cx^{-1/2}$ ， $c$  是一个常数。从  $h$  到 1 的积分是正则的，容易处理，而从 0 到  $h$  的积分则可以近似为  $2ch^{1/2} = 2hf(h)$ 。

**习题 1.3** 写一个程序，用上面讨论过的求积公式之一计算定积分

$$\int_0^1 t^{-2/3} (1-t)^{-1/3} dt = 2\pi/\sqrt{3},$$

并且考察它对不同  $h$  值的精度。（提示：把积分域分为两部分，在每个积分中作不同的变量替换以处理奇点）。

### 1.3 求 根

最后一种经常要用的初等运算是求一个函数  $f(x)$  的根，这个函数  $f(x)$  在任何  $x$  值上都可以求值。若已知一个根（比方说  $x = x_0$ ）的大致位置，一个肯定能成功的方法是，先随便猜一个  $x$  的试验值，要保证这个  $x$  值比根小，然后以小的正步长增加  $x$  之值，每当  $f$  值改变符号就将步长减半倒退回来。用这种方法产生的  $x$  值显然将收敛到  $x_0$ ，因此这个搜索过程在步长小于所要求的容许误差时就可以终止。下面的 BASIC 程序求函数  $f(x) = x^2 - 5$  的正根 ( $x_0 = \sqrt{5} = 2.236068$ ) 至  $10^{-6}$  的容许误差，以  $x = 1$  为初始猜测值，初始步长为 0.5：

```

5 DEF FNF(X)=X*X-5           '待求根的函数
10 TOLX=1.E-06              '搜索的容许误差
15 X=1:FOLD=FNF(X):DX=.5    '初始猜测值,函数值,步长
20 ITER%=0                  '迭代次数初始化
25 '
30 WHILE ABS(DX)>TOLX
35   ITER%=ITER%+1          '迭代次数增值
40   X=X+DX                 'X 向前一步
45   PRINT ITER%,X,SQR(5)-X '输出当前值
50   IF FOLD=FNF(X)>0 THEN GOTO 60 '若符号不变,前进一步
55   X=X-DX:DX=DX/2        '向后,步长减半
60 WEND
65 '
70 STOP
    
```

得到的  $x$  值序列示于表 1.4 中，它显然收敛到正确的答案，虽然要经过大约 33 次迭代。在使用这个方法时必须小心，因为如果初始步长取得太大，在  $f$  有几个根的情况下有可能跨过想要的根。

表 1.4 求  $f(x) = x^2 - 5$  的正根过程中的误差

迭代次数	搜索法	Newton 法 (1.14)式	弦割法 (1.15)式
0	1.236076	1.236076	1.236076
1	0.736068	-0.763932	-1.430599
2	0.236068	-0.097265	0.378925
3	-0.263932	-0.002027	0.098137
4	-0.013932	-0.000001	-0.009308
5	0.111068	0.000000	0.000008
6	-0.013932	0.000000	0.000000
	.....		
33	0.000001	0.000000	0.000000

**习题 1.4** 对不同的容许误差、初始猜测值和初始步长运行上面的程序。注意有时可能收敛到负根。若你从初始猜测值  $x = -3$  和初始步长 6 出发，会发生什么情况？

如果我们能够算出任意  $x$  处  $f$  的导数之值, 就有一个更有效的算法: Newton-Raphson 方法. 这个方法产生出一个序列  $x^i$ , 假定  $f$  在  $x_0$  附近具有局部线性性质, 这个序列就收敛到真正的根  $x_0$  (见图 1.2). 序列  $x^i$  的递推关系是

$$x^{i+1} = x^i - \frac{f(x^i)}{f'(x^i)} \quad (1.14)$$

用这个方法求  $\sqrt{5}$  的结果也示于表 1.4 中, 它的快速收敛特性 (仅作 5 次迭代) 是很明显的. 这是计算机算平方根的常用算法: (1.14) 式在  $x_0$  周围的线性化表明, 每次迭代将使有效数字位数加倍, 表 1.4 中明显地示出这一点.

弦割法在 Newton-Raphson 法的效率与必须计算导数的麻烦这二者之间提供了一个很好的折衷. 如果用 (1.4b) 式给出的差分公式

$$f'(x^i) = \frac{f(x^i) - f(x^{i-1})}{x^i - x^{i-1}}$$

近似代替 (1.14) 式中的导数, 我们将得到下述递推公式, 它包含三项  $x^{i+1}$ ,  $x^i$  和  $x^{i-1}$ ,  $x^{i+1}$  通过  $x^i$  和  $x^{i-1}$  给出 (见图 1.2):

$$x^{i+1} = x^i - f(x^i) \frac{(x^i - x^{i-1})}{f(x^i) - f(x^{i-1})} \quad (1.15)$$

$x_0$  的任何两个近似值可以用作  $x^0$  和  $x^1$  以启动这个算法, 当相继两次迭代所得的  $x$  值的变化小于所要求的容许误差时使算法终止. 用弦割法解我们的例题的结果 (从  $x^0 = 0.5$  和  $x^1 = 1.0$  开始) 也示于表 1.4 中. 只要初始的猜测值和真正的根邻近, 准确答案收敛的速度几乎同 Newton-Raphson 算法一样快.

**习题 1.5** 写出用 Newton-Raphson 方法和弦割法求解  $x^2 - 5$  的正根的程序. 改变对根的初始猜测值, 考察弦割法的收敛行为.

当函数在其根附近的行为不是良性时 (例如在  $x_0$  附近有一拐点), 或者当函数有几个根时, “自动的” Newton-Raphson 法和弦割法可能根本不收敛, 或者收敛到错误的答案, 如果对根的初始猜测值离真值很远的话. 因此, 一个保险的同时也是保守的办法是, 先用搜索法近似确定  $x_0$  的位置, 然后再用两种自动方法中的一种.

**习题 1.6** 函数  $f(x) = \tanh x$  在  $x=0$  处有一个根. 写一个程序以表明, 对于初始猜测值  $x \geq 1$  Newton-Raphson 法不收敛. 考察  $\tanh x$  的图形, 你能理解毛病出在哪儿吗? 从关于这个问题的 (1.14) 式的显形式出发, 导出初始猜测值的临界值, 在此值之上将不收敛. 试用弦割法解这个问题. 如果试用这两种方法求  $\tan x$  的  $x=0$  那个根, 那么对于不同的初始猜测值, 又会发生什么情况?

## 1.4 分子振动的半经典量子化

作为一个包含有几种基本数学运算的例子, 我们来考虑如何描述一个双原子分子, 这种分子 (比如  $O_2$ ) 由两个原子核组成, 两个核被环绕它们运动的电子束缚在一起. 由于原子核比电

子重得多, 我们可以假定电子运动得足够快, 使得能够进行即时的调整, 以适应原子核的不断变化的位置(Born - Oppenheimer近似). 于是, 上述问题就化为这样一个问题, 在其中两个原子核的运动受一个只依赖于两个原子核之间距离  $r$  的位势  $V$  支配. 产生  $V$  的物理原理将在课题VIII中详细讨论, 但是根据一般的理由可以说, 这个位势在远距离处是吸引势(van der Waals相互作用), 而在近距离处是排斥势(原子核的Coulomb相互作用和电子的Pauli斥力). 能够体现这些特征的一个常用的  $V$  的形式是Lennard - Jones位势或称6 - 12位势:

$$V(r) = 4V_0 \left[ \left( \frac{a}{r} \right)^{12} - \left( \frac{a}{r} \right)^6 \right], \quad (1.16)$$

其形状如图1.3上部所示, 极小点在  $r_{\min} = 2^{1/6}a$  处, 深为  $V_0$ . 我们在下面的大部分讨论中将假定  $V$  取这一形式. 对双原子分子的透彻讨论见文献 [He 50], 而Born - Oppenheimer近似则在 [Me 68] 中有讨论.

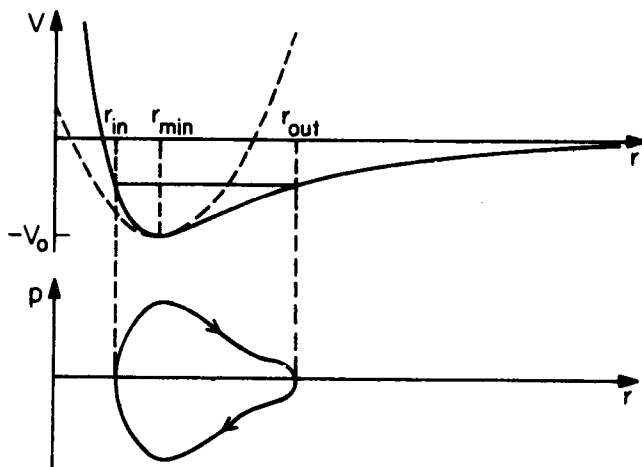


图 1.3 (上部)Lennard - Jones 位势和处于负能量的内外转折点. 虚线表示对位势的抛物线近似.  
(下部)相空间中对应的轨道.

原子核的巨大质量还允许对问题进一步简化, 即消除原子核的缓慢转动同它们之间距离的更为快速的变化之间的耦合. 前者由一个刚性哑铃的量子力学转动很好地描写, 而能量为  $E_n$  的相对运动的振动态则由一维Schrödinger方程

$$\left[ -\frac{\hbar^2}{2m} \frac{d^2}{dr^2} + V(r) \right] \psi_n = E_n \psi_n \quad (1.17)$$

的束缚态解  $\psi_n(r)$  描述, 其中  $m$  是两个原子核的约化质量.

在本例题中我们的目的是, 给定一个具体位势后求出各个能量  $E_n$ . 这些能量值可以通过解微分本征值方程(1.17)精确求得; 其数值方法将在第3章讨论. 不过, 原子核的巨大质量意味着

它们的运动接近经典运动, 因此, 各个振动能  $E_n$  的近似值可以通过考虑原子核在势场  $V$  中的经典运动, 然后再应用“量子化法则”决定能量来得到. 这些量子化法则原来由 N. Bohr, Sommerfeld 和 Wilson 等人提出, 它们曾是“旧”量子论的基石, 量子力学的近代表述是由旧量子论脱胎而来的. 但是, 它们也可通过考虑波动方程(1.17)的 WKB 近似而得到. (详见 [Me 68]).

势场  $V$  中原子核间距离的经典约束运动, 可以在能量  $-V_0 < E < 0$  上发生. 原子核间的距离周期性地振动于图 1.3 中的内转折点  $r_{in}$  和外转折点  $r_{out}$  之间(但不一定是简谐振动). 在振动过程中, 能量在相对运动的动能和位能两种形式之间进行交换, 而总能量

$$E = \frac{p^2}{2m} + V(r) \quad (1.18)$$

是一常数( $p$  是原子核的相对运动的动量). 因此, 我们可以把任何给定能量的振动想象为在相空间(坐标为  $r$  和  $p$ )中规定了一条闭合轨道, 如图 1.3 下部所示, 沿着这条轨道满足(1.18)式. 对  $p$  求解方程(1.18), 可得到这条轨道的显式方程:

$$p(r) = \pm [2m(E - V(r))]^{1/2}. \quad (1.19)$$

上述的经典运动在  $-V_0$  和  $0$  之间的任何能量上都会发生. 为了对这种运动进行量子化, 从而得到对出现在(1.17)式中的诸本征值  $E_n$  的近似值, 我们考虑给定能量上的无量纲作用量

$$S(E) = \oint k(r) dr, \quad (1.20)$$

其中  $k(r) = \pi^{-1} p(r)$  是局部 de Broglie 波数, 积分是在振动的一个完整循环上进行. 这个作用量正是相空间轨道所包围的面积(以  $\hbar$  为单位). 量子化法则声称, 在允许的能量  $E_n$  上, 作用量是  $2\pi$  的半整数倍. 于是, 应用(1.19)式, 并想到每个循环中振动要经过每个  $r$  值两次(一次的  $p$  值为正, 一次的  $p$  值为负), 我们有

$$S(E_n) = 2 \left[ \frac{2m}{\hbar^2} \right]^{1/2} \int_{r_{in}}^{r_{out}} [E_n - V(r)]^{1/2} dr = \left( n + \frac{1}{2} \right) 2\pi, \quad (1.21)$$

其中  $n$  是一个非负整数. 在积分限即转折点  $r_{in}$  和  $r_{out}$  上, 被积函数为零.

为了对 Lennard-Jones 势(1.16)具体确定量子化条件, 我们定义几个无量纲量

$$\varepsilon = \frac{E}{V_0}, \quad x = \frac{r}{a}, \quad \gamma = \left[ \frac{2ma^2 V_0}{\hbar^2} \right]^{1/2},$$

于是(1.21)式变为

$$s(\varepsilon_n) \equiv \frac{1}{2} S(\varepsilon_n V_0) = \gamma \int_{x_{in}}^{x_{out}} [\varepsilon_n - v(x)]^{1/2} dx = \left( n + \frac{1}{2} \right) \pi, \quad (1.22)$$

其中

$$v(x) = 4 \left[ \frac{1}{x^{12}} - \frac{1}{x^6} \right]$$

是标度化后的位势。

$\gamma$  这个量是本问题的量子本性的一个无量纲的测度。在经典极限下( $\hbar$  小或  $m$  大)  $\gamma$  很大。若知道分子的转动惯量(可从分子转动的能量得知)和离解能(把分子分解为组成它的两个原子所需的能量),就能从实验观测测定参量  $a$  和  $V_0$ ,从而定出  $\gamma$ 。对于  $H_2$  分子,  $\gamma = 21.7$ ; 而对于 HD 分子,  $\gamma = 24.8$ (一个质子换成氘核时,只是  $m$  改变,  $V_0$  不变); 对于由两个  $^{16}O$  核构成的重得多的  $O_2$  分子,  $\gamma = 150$ 。这些值都相当大,它们表明,半经典近似是这种振动运动的一个恰当的描述。

例题 1 的 BASIC 程序的源程序代码列在附录 B 中,也包含在本书所附磁盘上的文件 EXAM1.BAS 中,它对输入的  $\gamma$  值求使方程(1.22)成立的各个  $\epsilon_n$  值。在求出全部能量之后,再画出对应的相空间轨道。(试图在你的计算机系统上运行这个程序之前,你应当复习一下“怎样使用本书”这一段中以及附录 B 的开头处关于程序的一些说明)。

下面这些练习的目的在于增进你对本例题所阐明的物理原理和数值方法的理解。

**习题 1.7** 用计算机作工具解算物理问题时最重要的问题之一,是应当知道什么时候才能确信程序给出的解答是正确的。在这方面,一个重要的检验方法是在可用解析方法求解的情况下,把程序运行结果同已知结果进行详尽的定量比较。修改源程序,改用一个抛物线位势(见程序第 160 行,注意 170—180 行上的警告),对于这种位势 Bohr-Sommerfeld 量子化会给出 Schrödinger 方程的精确本征值:一个等间隔的能量系列,其中最小的一个超出位势极小值的高度是能级间隔的一半。对于几个不同的  $\gamma$  值,比较这种情形下的数值结果和用解析方法解方程(1.22)所得到的结果。相空间轨道是你预期的吗?

**习题 1.8** 对一个正在使用的程序的另一种重要的检验方法,是把它的结果同根据物理直观所预期的结果进行比较。将程序恢复到原来的样子,仍用 Lennard-Jones 势,对  $\gamma = 50$  运行这个程序。我们注意到,同上一习题所讨论的纯抛物线位势的情况一样,第一激发态超出位阱底的高度大致为基态超出位阱底的高度的 3 倍,最低的几个能态之间的间隔也大致是常数。这是因为, Lennard-Jones 势在其极小值附近大致成抛物线形状(见图 1.3)。通过计算  $V$  在极小值上的二阶导数,求出“弹簧常数”,并且证明,小振幅振动的频率之预期值为

$$\frac{\hbar\omega}{V_0} = \frac{6 \times 2^{5/6}}{\gamma} \approx \frac{10.691}{\gamma} \quad (1.23)$$

验证这个结果同数值结果一致,并且对不同的  $\gamma$  值考察这种一致性。通过比较 Lennard-Jones 势与其抛物线近似二者的形状,你能理解为什么较高的能量值之间的间隔要比较低的能量值之间的间隔密吗?

**习题 1.9** 改变数值算法或其参量时结果的不变性,可以使我们进一步相信计算的正确性。改变搜索转折点和搜索能量的容许误差(120 行)或 Simpson 法则积分点的数目(130 行),观察对结果的影响。注意:由于计算预期的束缚态数目的方式(1190—1200 行),在改变能量的容许误差时这个量可以有变化。

**习题 1.10** 把用搜索法求内外转折点改为用 Newton-Raphson 法或弦割法(当  $N\% \neq 0$ , 关于  $N\% - 1$  的内外转折点是有很好的启动值)。又把求  $S$  的 Simpson 求积法则换成更高阶的公式((1.13a)式或(1.13b)式)。观察对结果的改进。

**习题 1.11** 画出 Lennard-Jones 势的  $\epsilon_n$  同  $\gamma$  的函数关系( $\gamma$  之值从 20 运行到 200),并且解释所得的结果。(如

同对许多别的简短计算一样,你会发现,单是运行程序并在这个过程中用手画出结果,要比试图使绘图操作自动化的效率更高些.)

**习题 1.12** 对于 $H_2$ 分子,观测表明,位势的深度为 $V_0 = 4.747\text{eV}$ ,位势最小值的位置为 $r_{\min} = 0.74166\text{\AA}$ .这两个量同(1.23)式一起,给出一个振动频率.

$$\hbar\omega = 0.492V_0 = 2.339\text{eV},$$

它比实验观察到的基态同第一振动态之间的能量差 $0.515\text{eV}$ 要大三倍多.因此,Lennard-Jones 形状的位势并不是 $H_2$ 分子位势的一个很好的描述.它的另一缺陷是,它预言有6个束缚态,但已知有15个束缚态存在.(见表1.5,表中各项是从文献[Wa 67]中所引数据推出的).我们需要位势的一种更好的解析形式,它含有更

表 1.5 实验得出的 $H_2$ 分子的振动能

$n$	$E_n$ (eV)	$n$	$E_n$ (eV)
0	-4.477	8	-1.151
1	-3.962	9	-0.867
2	-3.475	10	-0.615
3	-3.017	11	-0.400
4	-2.587	12	-0.225
5	-2.185	13	-0.094
6	-1.811	14	-0.017
7	-1.466		

多的参量,才能同时复现极小值的深度和位置、极小值两旁的小振幅振动的频率、以及束缚态的总数.一种这样的形式是 Morse 位势

$$V(r) = V_0 [(1 - e^{-\beta(r-r_{\min})})^2 - 1], \quad (1.24)$$

这种位势也可以用解析方法求解<sup>①</sup>. Morse 位势在预期的位置上出现极小值,参量 $\beta$ 可以调节,以使极小点上的曲率同观测到的第一振动态的激发能相拟合.求适合于 $H_2$ 分子的 $\beta$ 值,修改上面的程序以改用 Morse 位势,并且计算振动态的谱.证明这时得到的能级数目要合理得多.把这些能量值同实验值并同由 Lennard-Jones 势得出的能量值进行比较,解释后一比较中的差异.

## 课题 I : 中心位势的散射

在本课题中,我们将考察一个中心位势(特别是上面在第1.4节中考虑过的 Lennard-Jones 位势)对一个质量为 $m$ 的粒子的经典散射.在一次散射事件中,具有初始动能 $E$ 和碰撞参量 $b$ 的粒子从很远的地方射向该位势.当它经过力心附近时受到偏转,最终以同一能量出射,但是运动方向与原来的方向成一角度 $\Theta$ .由于位势只与粒子到力心的距离有关,角动量是守恒的,运动轨道在一平面内.粒子的极坐标 $(r, \theta)$ 是描述粒子运动的方便手段,如图

<sup>①</sup> 参看 S. Flügge, Practical Quantum Mechanics, Problem 70. (有中译本:《实用量子力学》, [德]S. 福里格著,宋孝同等译,高等教育出版社,1981.) ——译者注.



I.1 所示。(详细讨论可参看任何一种经典力学教科书, 比如 [Go 80].)

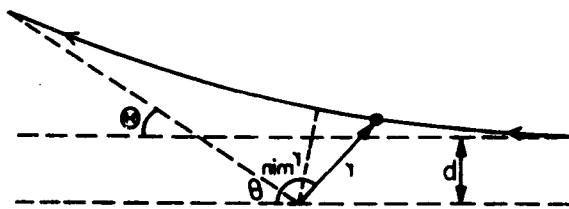


图 I.1 粒子被中心位势散射时涉及的量

具有基本意义的是偏转函数  $\Theta(b)$ , 它给出最终的散射角  $\Theta$  同碰撞参量的函数关系; 这个函数还同入射能量有关. 散射角为  $\Theta$  时的微分截面  $d\sigma/d\Omega$  是一个实验可观察量, 它同偏转函数的关系是

$$\frac{d\sigma}{d\Omega} = \frac{b}{\sin\Theta} \left| \frac{db}{d\Theta} \right|. \quad (\text{I.1})$$

于是, 如果可以算出  $d\Theta/db = (db/d\Theta)^{-1}$ , 那么微分截面就知道了.

只是对很少几种位势, 偏转函数的表示式才可以用解析方法求出, 因此通常必须采用数值方法. 求解这个问题的一种办法, 是对运动方程 (即联系加速度和力的牛顿定律) 积分, 以求出对应于不同碰撞参量的各个轨道, 然后列出最终的运动方向 (散射角). 这包括积分四个耦合的一阶微分方程, 它们分别是关于散射平面内的两个坐标及其速度的. 这一方法将在后面第 2.5 节讨论. 但是, 由于角动量守恒,  $\Theta$  的演化直接和径向运动相联系, 本问题可以化为一个一维问题, 它可以通过求积来求解. 这后一种方法比较简单而且比较精确, 下面我们就来介绍这种方法.

为了导出  $\Theta$  的一个适当的表示式, 我们从角动量守恒出发. 角动量守恒意味着

$$L = mvb = mr^2 \frac{d\theta}{dt} \quad (\text{I.2})$$

是一个运动常数, 其中  $d\theta/dt$  是角速度,  $v$  是渐近速度, 它同轰击能量的关系是  $E = \frac{1}{2} mv^2$ .

径向运动发生在由  $V$  和离心位势相加而成的有效位势中, 因此能量守恒意味着

$$\frac{1}{2} m \left[ \frac{dr}{dt} \right]^2 + \frac{L^2}{2mr^2} + V = E. \quad (\text{I.3})$$

如果我们在 (I.2) 式中用  $r$  作自变量而不用时间, 可以写出

$$\frac{d\theta}{dr} = \frac{d\theta}{dt} \left[ \frac{dr}{dt} \right]^{-1} = \frac{bv}{r^2} \left[ \frac{dr}{dt} \right]^{-1}, \quad (\text{I.4})$$

从 (I.3) 式解出  $dr/dt$ , 于是得到

$$\frac{d\theta}{dr} = \pm \frac{b}{r^2} \left[ 1 - \frac{b^2}{r^2} - \frac{V}{E} \right]^{-1/2}. \quad (\text{I.5})$$