

计算物理



# 引言

这只是课堂笔记，如有错误欢迎反馈。

开场白：找到喜欢的专业

## 0.1 主要内容

1. 数值计算的基本观念-重点讲述
2. 数值计算的常用算法-重点讲述
3. 数值模拟-涉及部分，会讲述随机模型模拟
4. 广泛应用-基本不讲

## 0.2 为什么学习数值计算和数值模拟？

1. 一种基本的计算技巧

-物理理论与实验的需要

-物理研究的过程 A、new ideas, B、建模推导公式, C、画图, D、总结。其中 B 到 C 的过程往往需要许多计算物理的内容。

-有些式子无法计算，必须数值模拟 例如： $\frac{\sin \omega}{\omega}$

-一个数值模拟的实例 渗流模型

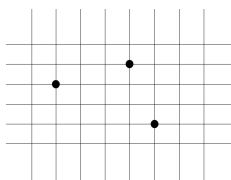


图 0.1: 渗流模型

二维系统的渗流模型简单模拟：遍历每一格点用 random 函数随机在  $[0, 1]$  中取数  $r$ ，与事先假定的概率  $P$  进行比较。如果  $r < P$  则放置点，如果  $r > P$  则不放点，两点相邻放置时表示导通。若最终如果放置的点可以连接成导通相对边界的路线，则表示可以发生渗流。在这一过程中有一临界概率  $P_c$ ，当  $P > P_c$  是总可以发生渗流，模拟可以得到

该临界概率为  $P_c = 0.593$ 。渗流模型可以应用于物理中的一些相变问题，例如从导通金属态到绝缘态的相变。

## 2. 拓展数值计算能力

-高数 + 线代 + 概率论与数理统计 + 方程 加以拓展从解析运算到数值计算，某些复杂积分需要理用计算物理的方法处理才能到计算机上运算,例如当  $n$  很大时计算  $\sqrt{n+1} - \sqrt{n}$  或者计算  $\int_0^1 p \ln p dp$ 。

## 3. 开拓新领域的需要

A、人工智能-机器学习 与固体物理中的层状结构有一些联系，涉及重整化理论

B、工业工程（系统分析） data science（大数据）

C、CT 成像 通过计算分析解决某些部分难成像的问题

D、数量... 量化... 涉及这两个词一般都与数值计算有关

## 0.3 算法（Algorithm）

字面理解即计算方法，进一步指某一问题和实体的计算方法。

错误分类

1. 编程错误（低级错误，可通过编译器的提示纠错）
2. **算法错误**（高级错误，具有隐蔽性不易察觉，如反常积分的计算），这门课将提供一些解决此类错误的思路。

## 0.4 参考书

1. W.H Press, Numerical Recipes  
-主要讲述了一些算法错误，偏向于应用，可以在学习完课程后阅读，会有一些收获
2. F.S.Action,Real Computing Made Real,1996  
-书中插图有关中国门神，可能寓意抓住计算物理中的小鬼（错误）
3. F.S.Acton,Numerical Method that Work,1970  
-年代比较久远
4. 徐翠薇，孙继武，计算方法引论

以上教科书仅供参考，老师所讲内容涵盖范围较广，重点还是上课跟上老师。

## 0.5 计算软件

关于 Python 的库，Google 公司开发了 TensorFlow 库。曾经 Acton 反对用库、包，他认为这些就是 black box，可能会存在错误。但现如今许多库和包已经经过了很多人和长时间的验证，错误可以说是很小了。而且即便有错误，也可以通过对算法和结果的分析进行纠正，因此更多还是折衷选择调用。

表 1: 计算软件对比

	Windows	Linux
语言	Fortran,C,C++,Python	Fortran,C,C++,Python
编译器	Intel, 其它	gcc,intel
作图	Origin 软件	2D 图 Grace,Python 库作图
写文章	Word Vs LaTeX	LaTeX

## 0.6 计算工具（更广泛一些）

1. 数值计算-掌握一些常用算法，防止算法错误

2. 解析计算-手工推导公式

常用工具：Scientific Workplace，所见即所得，底层命令为 LaTeX。

3. 符号计算-Mathematica®，（数学手册）

4. 并行计算

- 传统 MPI+Fortran\C，多个 CPU
- 并行 GPU（基于 CUPA 平台）

5. 计算机机群，一般为 Linux system

- cluster，多个 CPU 核
- cluster，多个 CPU+GPU



# 目录

引言	iii
0.1 主要内容	iii
0.2 为什么学习数值计算和数值模拟?	iii
0.3 算法 (Algorithm)	iv
0.4 参考书	iv
0.5 计算软件	iv
0.6 计算工具 (更广泛一些)	v
<b>1 数值计算的基本观念 (concept)</b>	<b>1</b>
1.1 浮点数、浮点运算和舍入误差	2
1.1.1 十进制、二进制	2
1.1.2 浮点数和规格化的浮点数	2
1.1.3 机器精度	4
1.1.4 浮点运算	5
1.1.5 数值计算的两种误差	5
1.2 舍入误差的传递	6
1.2.1 舍入误差的传递	6
1.2.2 防止舍入误差的方法 (措施)	7
1.3 算法的准确性和稳定性	8
1.3.1 算法的准确性	8
1.3.2 算法的稳定性	8
<b>2 函数求值</b>	<b>11</b>
2.1 $\frac{\sin x}{x}$ 型函数求值	11
2.2 级数, 连分式, 阶乘	12
2.2.1 级数	12
2.2.2 连分式 (格点格林函数)	12
2.2.3 阶乘 (整数化为浮点数)	12
<b>3 方程求根</b>	<b>13</b>
3.1 迭代法求根	13
3.1.1 迭代法求根的步骤	13

3.1.2	迭代法的收敛条件	14
3.1.3	迭代法的几何表示	16
3.2	牛顿迭代法和弦截法	16
3.2.1	牛顿迭代法	16
3.2.2	弦截法	17
3.3	二分法	18
3.4	“笨”方法	19
<b>4</b>	<b>函数插值和数值微分</b>	<b>21</b>
4.1	拉格朗日插值	21
4.1.1	线性插值	21
4.1.2	二次插值	23
4.1.3	$n$ 次插值	24
4.1.4	Runge 现象和分段线性插值	24
4.2	数值微分	25
4.2.1	两点公式	25
4.2.2	三点公式	25
4.3	厄米特插值	26
4.4	实验数据拟合与函数插值	26
<b>5</b>	<b>数值积分</b>	<b>27</b>
5.1	梯形积分法	27
5.1.1	梯形积分法	27
5.1.2	截断误差	28
5.1.3	其他简单形式	30
5.1.4	数值积分公式	30
5.2	辛普森 (Simpson) 积分法	30
5.2.1	辛普森积分方法	30
5.2.2	截断误差	31
5.2.3	柯特斯公式	32
5.3	龙贝格 (Romberg) 积分法	32
5.4	高斯积分法	35
5.4.1	两点高斯积分	35
5.4.2	$n$ 点高斯积分	36
5.5	反常积分	37
5.5.1	处理反常积分的基本方法	38
5.5.2	反常积分的几何行为	41
5.6	反常积分举例	42
5.6.1	含可去奇点的反常积分	42
5.6.2	具有水平方向渐进行为的反常积分	43
5.6.3	含有 $\ln x (x \rightarrow 0 \text{ 或 } x \rightarrow \text{很大})$ 的反常积分	44
5.6.4	含有动态奇异性的反常积分	48

当时考了  
高数



<b>6</b>	<b>快速傅里叶变换算法</b>	<b>49</b>
6.1	有限离散傅氏变换	49
6.2	快速傅氏变换	50
<b>7</b>	<b>线性代数方程组的解法</b>	<b>53</b>
7.1	直接法（高斯消去法）	53
7.1.1	高斯消去法	53
7.2	三对角线型和一般带型线性代数方程组	54
7.2.1	三对角型	54
7.2.2	一般带型	55
7.3	迭代法	55
7.3.1	简单迭代法	55
7.3.2	Seidal 迭代形式	56
7.3.3	松弛法	56
7.4	LU 分解法	56
7.4.1	LU 分解法	57
7.4.2	矩阵的三角分解	57
7.4.3	矩阵求逆	57
7.5	“病态”线性代数方程组	58
7.5.1	“病态”线性代数方程组	58
7.5.2	实际计算出现“病态”情况的原因	58
7.5.3	“病态”线性方程组求解	58
<b>8</b>	<b>矩阵的本征值和本征矢的计算</b>	<b>61</b>
8.1	雅可比方法	62
8.1.1	实对称矩阵的性质	62
8.1.2	二阶对称矩阵对角化	63
8.1.3	雅可比方法	64
8.2	QR 算法	67
8.2.1	Householder 变换	67
8.2.2	QR 算法	68
8.3	厄米矩阵的本征值与本征矢	68
<b>9</b>	<b>随机模型模拟</b>	<b>75</b>
9.1	随机模型	76
9.1.1	单位圆求 $\pi$	76
9.1.2	浦丰投针	77
9.1.3	渗流模型 (Percolaton Model)	77
9.1.4	分形结构	78
9.2	随机数	78
9.3	随机数发生器——产生随机数	78
9.3.1	抛硬币	78

9.3.2	用程序算出来的随机数（伪随机数）	79
9.3.3	随机数的统计检验	80
9.3.4	实际应用	80

## 10 机器学习导论 81

10.1	神经网络的架构	83
10.1.1	人的大脑的信息处理	83
10.1.2	神经元模型	83
10.1.3	神经网络结构（建模，Building model）	85
10.2	神经网络和学习算法	85
10.2.1	监督学习的过程	85
10.2.2	反向传播算法 (Back propagation, BP 算法)	86
10.3	简单应用，手写数字识别	88
10.3.1	数据集	88
10.3.2	构建模型	89
10.3.3	训练	90
10.3.4	测试	90
10.3.5	泛化	90
10.4	伊辛模型的相变识别	91
10.5	伊辛模型	91
10.5.1	Wolff 算法（集团算法）	91
10.5.2	神经网络识别 Ising Model 相变	91
10.5.3	相变温度的确定	92

这书也老  
都很难学

# 第1章 数值计算的基本观念 (concept)

例 1、

求和程序:  $\sum_{i=1}^{10000} 0.0001$

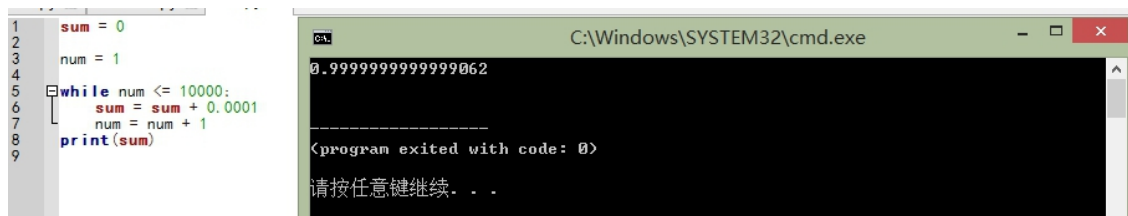


图 1.1: Python 代码及结果

可以看出计算机算出的结果与实际结果（人脑算出的结果）有一些差别。究其原因，有以下两点：

- 0.0001 在计算机中要有十进制数转化成二进制数，而最后结果又要再化为十进制数，这就会导致循环小数的出现。
- 计算机存储的字节有限

例二

解一元二次方程：

$$x^2 - (10^{17} + 1)x + 10^{17} = 0 \quad (1.1)$$

如果我们利用求根公式  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ ，在计算机中编一小程序计算得到的结果为  $x_1 = 10^{17}, x_2 = 0$ （如果得不出这个结果，可以提高幂次）。然而我们很快的利用十字相乘的方法得出解为  $x_1 = 10^{17}, x_2 = 1$ ，原因依然有两条

- 由于  $-b = 10^{17} + 1$ ，在计算机的存储过程中  $0.00 \dots 1 \times 10^{17}$  可能会被“抹掉”，即在计算机中  $-b = 10^{17}$ ，同理  $\sqrt{b^2 - 4ac} = b$ ，这样就会导致一个解为 0
- $-b$  和  $\sqrt{b^2 - 4ac}$  十分接近，在计算机中相近的数相减，得到的数误差会很大（就像是测量时的估读位，相减计算实际是估读位间的计算，而估读位本身就有误差，在做运算误差会更大？）

上述的错误算是算法错误, 可以采用别的算法得到正确结果, 如果认为求根公式是算法一, 我们可以利用如下公式当做算法 2

$$x_1 = \frac{-b - \text{sign}(b)\sqrt{b^2 - 4ac}}{2a}, x_2 = \frac{c}{ax_1} (\text{利用 } x_1 x_2 = \frac{c}{a}) \quad (1.2)$$

上述公式可以得到正确结果, 其有如下特点

- 避免了两个相近的数相减
- 巧合: 对于这个特定的方程, 仍有如下关系  $-b \approx 10^{17}$ ,  $\sqrt{b^2 - 4ac} \approx b$ , 代入上式仍可得到正确结果。

总之算法 2 对该类方程不完全适用, 但对一般方程有相对的优越性。

## 1.1 浮点数、浮点运算和舍入误差

### 1.1.1 十进制、二进制

计算机中默认的是二进制, 主要是可以通过电控制门电路产生 01。一些语言像是 C++、Python 先是通过英文语句的编译, 自动地将数转化成二进制数有如下特点

1. 整数可以严格转换
2. 分数
  - 一部分可以严格转化
  - 另一部分将转化为循环的二进制数

### 1.1.2 浮点数和规格化的浮点数

实数  $\rightarrow$  浮点数

人脑  $\rightarrow$  计算机

这种浮点数的存储方法虽然丧失了严格性和精确性, 但是存储的数范围扩大了。

#### 浮点数

简单来说实数在计算机中的存储和运算操作方式。进一步讲是属于有理数中某特定子集的数的数字表示, 在计算机中用以近似表示任意某个实数。具体的说, 这个实数由一个整数或定点数 (即尾数) 乘以某个基数 (计算机中通常是 2) 的整数次幂得到, 这种表示方法类似于基数为 10 的科学计数法。

表 1.1: 32 为浮点数存储结构

									...
0	1	2	3	4	5	6	7	8	23 位
符号位	指数								尾数

### 浮点数存储结构（32 字节）

浮点数： 实数 = 符号 尾数  $\times 2^{\text{指数}}$  ( $0 < \text{尾数} < 1$ )

科学计数法： 实数 =  $A \times 10^B$  ( $1 < A < 10$ )

存储时只存符号、指数和尾数（小于 1 的小数部分，即小数点后的部分）

例 十进制 1 化为浮点数

表 1.2: 十进制 1 化为浮点数

	0	00000001	100...00
$0.5 \times 2^1$	+	$2^1$	$0.5 = 2^{-1}$
	0	00000010	010...00
$0.25 \times 2^2$	+	$2^2$	$0.25 = 2^{-2}$

...

其中指数对应的是整数的二进制数（从  $2^1$  开始），尾数对用的是小数的二进制数（从  $2^{-1}$  开始）。

因为存尾数的字节有限，为避免浮点数存不上（1 前面有很多 0），常采用规格化的浮点数

### 规格化的浮点数

在尾数中没有初始零的浮点数称为规格化浮点数。

浮点数存在范围，单精度  $10^{-44} - 10^{38}$ ,  $-3.4 \times 10^{38} - 3.4 \times 10^{38}$ ,  $10^{-38} - 10^{38}$ ，双精度  $10^{-324} - 10^{308}$ ,  $-1.797 \times 10^{308} - 1.797 \times 10^{308}$ ，符合 IEEE 标准。

例

电子动能  $E = \frac{\hbar^2 k^2}{2m_e}$ ，其中在国际单位制下  $\hbar = 1.05 \times 10^{-34} \text{ J} \cdot \text{s}$ ,  $m_e = 9.11 \times 10^{-31} \text{ kg}$ ，可以看出计算过程中有的数值超出了浮点数的范围，这是需要人手动地去化简（可能比较繁琐，但更准确）。

### 浮点数的溢出

是以前阶码溢出表现出来的。在加减运算过程中要检查是否产生了溢出：若阶码正常，加（减）运算正常结束；若阶码溢出，则要进行相应处理。另外对尾数的溢出也需要处理。

- 阶码上溢超过了阶码可能表示的最大值的正指数值，一般将其认为是  $-\infty$  和  $+\infty$ 。
- 阶码下溢超过了阶码可能表示的最小值的负指数值，一般将其认为是 0。

- 尾数上溢两个同符号尾数相加产生了最高位向上的进位，将尾数右移，阶码增 1 来重新对齐。
- 尾数下溢在将尾数右移时，尾数的最低有效位从尾数域右端流出，要进行舍入处理。

### 1.1.3 机器精度

浮点数表面的数与真实数之间的差异多大？

若一个最小的浮点数加浮点数 1.0 后，产生一个不同于 1.0 的浮点数，则称这个最小的浮点数为机器精度  $\varepsilon$ 。即：满足  $1.0 + \varepsilon > 1.0$  的最小值  $\varepsilon_{min}$ ， $\frac{\varepsilon_{min}}{2}$  对应于尾数最小精度的变化。

表 1.3: 机器精度

十进制数	符号位	指数	尾数
1.0	0	00000001	10...00
$\varepsilon$	0	00000001	000.....001
...			
$\varepsilon_{min}$	0	00000001	00...01

...

可以看出  $\varepsilon_{min}$  是一个临界数，恰好可以在计算机中加到 1.0 上，使  $1.0 + \varepsilon_{min} > 1.0$  可以通过编程计算机器精度。

若  $\varepsilon = c \times 2^1$ ，则  $c = \frac{\varepsilon}{2}$  是最小精度位的变化量。可以得出机器精度如下：

- 单精度  $\varepsilon = 5.96 \times 10^{-8}$
- 双精度  $\varepsilon = 1.11 \times 10^{-16}$

可以近似认为  $\varepsilon \approx \frac{\varepsilon}{2}$

### 相对误差

$$\text{相对误差} = \frac{(A + \frac{\varepsilon}{2}) \times 2^B - A \times 2^B}{A \times 2^B} \times 100\% = \frac{\varepsilon}{A} \quad (1.3)$$

设浮点数为  $x^*$ ，真实值为  $x$ ，则

$$\frac{x^* - x}{x} = \varepsilon \Rightarrow x^* = x(1 + \varepsilon) \quad (1.4)$$

可以写成

$$x^* = x + \varepsilon_x, (\varepsilon_x = x\varepsilon) \quad (1.5)$$

人脑中真实数是连续的且从  $-\infty$  到  $+\infty$ ，但在计算机中浮点数是离散的 (分立的) 且有一定的范围。即并非一一对应，而是一个浮点数对应一个范围内的真实数，如下图所示

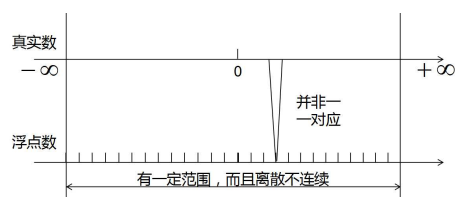


图 1.2: 真实数与浮点数

### 1.1.4 浮点运算

#### 浮点预算

浮点数（计算机里）加减乘除运算

例 十进制加法  $0.99+0.0001$ , 具体步骤

1. 每个数以规格化的浮点数形式存起来

$$-0.0001 \rightarrow 0.1101 \dots \times 2^{-13}$$

$$-0.9900 \rightarrow 0.1111 \dots \times 2^0$$

2. 加法对阶（使指数相同），往大数上对阶，一般就是在尾数前添零，导致末尾有些数字加不上导致误差

$$-0.0001 \rightarrow 0.0000 \dots 1101 \dots \times 2^0 \text{ (小数对阶)}$$

$$-0.9900 \rightarrow 0.1111 \dots \times 2^0 \text{ (大数不变)}$$

3. 尾数求和运算

$$\text{结果为: } 0.1111 \dots 1011 \times 2^0$$

4. 结果的规格化处理

5. 舍入处理

其他的运算与上述过程类似。

### 1.1.5 数值计算的两种误差

1. 舍入误差：由于计算机存储的字节有限，没有用尽二进制数的所有字节，去掉后面的数字而引起的误差。最原始的来源：

-实数化为浮点数时不能较为准确地转化

-浮点运算时引入的误差

这种误差虽然是由计算机硬件引起的，但实际中我们无法改变硬件，所以解决的办法就是利用不会增加舍入误差的算法。

2. 截断误差：数值计算时，由数值计算方法（算法）求出的近似解和准确解之间的误差称为截断误差。因为截断误差是数值计算方法固有的，因此又称方法误差。这种误差是由算法引起

的, 解决方法就是发展更好的算法。

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + f''(x_0)(x - x_0)^2 + \dots (\text{略去高阶项引入的误差}) \\ &\approx f(x_0) + f'(x_0)(x - x_0) + f''(x_0)(x - x_0)^2 \end{aligned} \quad (1.6)$$

## 1.2 舍入误差的传递

设  $a$ 、 $b$  为真实数,  $a^*$ 、 $b^*$  为计算机中相对应的浮点数。

### 1.2.1 舍入误差的传递

存储时

真实数在计算机中存储时, 就会舍入误差, 我们表示如下:

$$a^* = a + \varepsilon_a, \quad b^* = b + \varepsilon_b. \quad (1.7)$$

其中  $\varepsilon_a = \varepsilon \times a$ ,  $\varepsilon_b = \varepsilon \times b$ ,  $\varepsilon$  为机器精度。

浮点运算时

1. 加法:  $a + b$ , 计算机中是浮点数相加

$$a^* + b^* = a + b + \varepsilon_a + \varepsilon_b. (\text{大致精度}) \quad (1.8)$$

2. 减法:  $a - b$ , 类似于加法得到

$$a^* - b^* = a - b + \varepsilon_a - \varepsilon_b. \quad (1.9)$$

3. 乘法:  $a \times b$ , 在计算机中是浮点数相乘

$$a^* \times b^* = (a + \varepsilon_a) \times (b + \varepsilon_b) = a \times b + a \times \varepsilon_b + b \times \varepsilon_a + \varepsilon_a \times \varepsilon_b. \quad (1.10)$$

略去二阶小量, 我们可以得到相对误差

$$\frac{a^* \times b^* - a \times b}{a \times b} = \frac{a \times \varepsilon_b + b \times \varepsilon_a}{a \times b} = \frac{\varepsilon_b}{b} + \frac{\varepsilon_a}{a} \quad (1.11)$$

4. 除法:  $\frac{a}{b}$ , 代入浮点数

$$\begin{aligned} \frac{a^*}{b^*} &= \frac{a + \varepsilon_a}{b + \varepsilon_b} (\text{分母在 } b \text{ 的邻域作泰勒展开}) \\ &= (a + \varepsilon_a) \left( \frac{1}{b} - \frac{1}{b} \frac{\varepsilon_b}{b} + \frac{1}{b} \frac{\varepsilon_b^2}{b^2} + \dots \right) (\text{打开括号, 略去高阶项}) \\ &= \frac{a}{b} + \frac{\varepsilon_a}{b} - \frac{a \varepsilon_b}{b^2} + \dots \end{aligned} \quad (1.12)$$



相对误差为

$$\frac{a^*/b^* - a/b}{a/b} = \frac{\frac{\varepsilon_a}{b} - \frac{a\varepsilon_b}{b^2}}{a/b} = \frac{\varepsilon_a}{a} - \frac{\varepsilon_b}{b} \quad (1.13)$$

5. 函数  $f(x)$ , 计算  $f(a)$ , 则计算机中为  $f(a^*) = f(a) + f'(a)\varepsilon_a$  可以得到误差为

$$\Delta f = f(a^*) - f(a) = f'(a)\varepsilon_a \quad (1.14)$$

我们可以看出这时舍入误差的大小与具体的函数形式有关。

- $f'(a) > 1$  时, 放大误差
- $f'(a) < 1$  时, 缩小误差

例 1.1  $y = x^n$ .

相对误差为  $\frac{\Delta y}{y} = d \ln y = n d \ln x = n \frac{\Delta x}{x}$

### 1.2.2 防止舍入误差的方法（措施）

1. 避免“大数吃小数”，处理方法：变换公式

2. 避免两个相近的数相减，处理方法：变换公式

设  $x, y$  相近,  $u = x - y$ ,  $x^* = x + \varepsilon_x$ ,  $y^* = y + \varepsilon_y$ , 则相对误差为

$$\frac{\Delta u}{u} = \frac{(x^* - y^*) - (x - y)}{x - y} = \frac{\varepsilon_x - \varepsilon_y}{x - y} \quad (1.15)$$

由于  $x, y$  很相近, 所有  $x - y$  是一个很小的数, 这时误差就会很大<sup>1</sup>

例 1.2 一元二次方程  $ax^2 + bx + c = 0$ 。

一般的求根公式  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  并不准确。

变换公式:  $x_1 = \frac{-b - \text{sign}(b)\sqrt{b^2 - 4ac}}{2a}$ ,  $x_2 = \frac{c}{ax_1}$

其中第一式是为了避免两个相近的数相减, 第二式是为了避免“大数吃小数”。

例 1.3  $\sqrt{1+x} - \sqrt{x}$ ,  $x$  很大。

这是之前提到的例子, 可以化为  $\frac{1}{\sqrt{1+x} + \sqrt{x}}$

例 1.4  $\frac{1 - \cos x}{\sin x}$ ,  $x$  很小且为正数。

上式当  $x$  趋于零时,  $\cos x$  趋于 1, 为避免相近数相减, 可做如下变换

$$\frac{1 - \cos x}{\sin x} = \frac{(1 - \cos x)(1 + \cos x)}{\sin x(1 - \cos x)} = \frac{\sin x}{1 + \cos x}$$

<sup>1</sup> 由于  $\varepsilon_x = \varepsilon \times x$ , 误差似乎恒等于  $\varepsilon$ , 实际上误差每次不同  $\varepsilon_x$  是最大的误差,  $\varepsilon_x \neq \varepsilon_y$  (具体我也不清楚)。

3. 做除法  $\frac{a}{b}$  时, 避免  $|a| \gg |b|$  情况, 说明:

$$\begin{aligned} \frac{a^*}{b^*} - \frac{a}{b} &= \frac{\varepsilon_a}{b} - \frac{a\varepsilon_b}{b^2} = \frac{b\varepsilon_a - a\varepsilon_b}{b^2} \quad (\varepsilon_a \approx \varepsilon_b, a\varepsilon_b \gg b\varepsilon_a) \\ &= -\frac{a\varepsilon_b}{b^2} = -\frac{a}{b} \left( \frac{\varepsilon_b}{b} \right). \end{aligned} \quad (1.16)$$

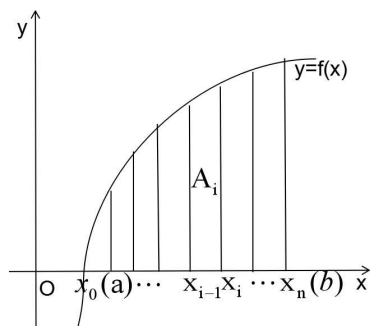
可以看出,  $a \gg b$  时, 舍入误差被放大。

4. 尽量减少运算次数

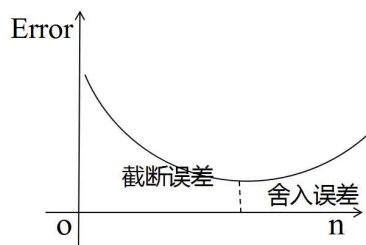
例 1.5  $S = a_0 + a_1 \frac{x}{1!} + a_2 \frac{x^2}{2!} + a_3 \frac{x^3}{3!} + a_4 \frac{x^4}{4!} + \dots$

简化计算可以得到  $S = a_0 + \frac{x}{1!} \{a_1 + \frac{x}{2} [a_2 + \frac{x}{3} (a_3 + \frac{x}{4} a_4)]\}$

例 1.6 用面积近似求某一积分  $\int_a^b f(x)dx$ , 将  $[a,b]$  等分为  $n$  份, 用梯形的面积和近似求积分。如果我们选择一个可以解析求解的积分, 可以做出误差随  $n$  变化的图像, 可以看出随  $n$  的增加误差并非一直减小, 而是减小到一个极小值后增加。



(a) 梯形积分法



(b) 梯形积分法计算的误差

图 1.3: 梯形积分法

## 1.3 算法的准确性和稳定性

### 1.3.1 算法的准确性

选择适合于问题的算法。

### 1.3.2 算法的稳定性

有时由于舍入误差放大也得不到准确值, 而导致结果错误。

例 1.7 计算积分  $I_n = \int_0^1 \frac{x^n}{x+5} dx$ ,  $x=0,1,2,3,\dots$

算法 1: 直接利用数值积分公式解析求解。

算法 2: 由  $I_n + 5I_{n-1} = \int_0^1 \frac{x^n}{x+5} dx + \int_0^1 \frac{5x^{n-1}}{x+5} dx = \frac{1}{n}$  可以得到两项递推关系:  $I_n + 5I_{n-1} = \frac{1}{n}$ 。

我们有  $I_0 = \ln \frac{6}{5} \approx 0.182322$ , 可以依次求出之后的积分值如下:

可以看出以上的结果有些事明显不对的 (积分值应该大于零)。分析如下:

理论值	迭代法计算值
[0.132321556763955]	0.132321556763955
[0.038392216050227]	-0.411607733369772]
[0.058038919848866]	2.591372253182199]
[0.04313873469005]	-11.706861255910996]
[0.03430632854975]	58.734306329564983]
[0.028468362256126]	-2.83648640811033e-02]
[0.02432490641035]	1.4676e+182043398e-03]
[0.021232615151970]	-7.338210910241993e-03]
[0.018036924240150]	3.669116566232107e-04]
[0.016926489610363]	-1.824557283115051e-05]

图 1.4: 迭代法计算值与理论值对比

计算机中:

$$\begin{aligned}
 I_0^* &= I_0 + \varepsilon \\
 I_1^* &= \frac{1}{1} - 5 \times I_0^* = 1 - 5I_0 - 5\varepsilon \\
 I_2^* &= \frac{1}{2} - 5 \times I_1^* = \frac{3}{10} + 25I_0 + (-5)^2\varepsilon \\
 &\dots
 \end{aligned} \tag{1.17}$$

可以看出  $\Delta I_n^* \sim (-5)^n \varepsilon$ , 即舍入误差逐渐被放大, 导致最后的结果错误, 这种方法数学上是正确的, 但对计算机来说是错误的算法。

算法 3: 依然利用上面的递推公式, 只是向  $n$  减小的方向做递推公式, 例如: 已知  $I_{20}$ , 利用  $I_{n-1} = \frac{1}{5n} - \frac{1}{5}I_n$  依次向下运算。可以推出  $\Delta I_n^* \sim \left(-\frac{1}{5}\right)^{20-n} \varepsilon$ , 即这时舍入误差是递减的。

例 1.8  $\varphi = \frac{\sqrt{5}-1}{2} = 0.618$ , 计算  $\varphi^n$ ,  $n$  为整数 ( $\varphi^0 = 0$ )

算法 1: 直接计算  $\varphi^n = \varphi \cdot \varphi \dots \varphi$ 。

算法 2: 利用三项递推关系  $\varphi^{n+1} = \varphi^{n-1} - \varphi^n$ , 当  $n$  很大时该算法的结果与实际不一致, 这种误差更隐蔽。

例 1.9 勒让德多项式, 满足  $(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$ , 但用这种方法算勒让德多项式在  $n$  很大时, 误差也很大。

总之, 注意防止舍入误差和截断误差。



## 第2章 函数求值

### 2.1 $\frac{\sin x}{x}$ 型函数求值

例 2.10  $y = \frac{\sin x}{x}$ .

用计算机直接计算上式  $x = 0$  的邻域 ( $0-10^{-7}$  左右) 的值, 会出现不规则的振荡。

$x = 0$  处的值可以用解析的方法求得 (利用洛必达法则)  $\frac{\sin x}{x}|_{x=0} = 1$ 。

在计算机中分两段求  $\frac{\sin x}{x}$ :

1.  $x \in [0, \varepsilon]$  ( $\varepsilon$  是趋于零的数), 一般采用泰勒展开的方法,  $\sin x$  可展为:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots \quad (2.1)$$

则  $y = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} + \dots$ , 这样便可以较为准确地求解了。

2.  $x \in [\varepsilon, \infty]$ , 直接计算。

例 2.11  $y = \frac{1 - \cos x}{x^2}$ .

类似于  $\frac{\sin x}{x}$ ,  $x=0$  处的值可以用解析的方法求得 (利用洛必达法则)  $y = \frac{\sin x}{2x}|_{x=0} = \frac{\cos x}{2}|_{x=0} = 1$ 。

在计算机中同样分两段求:

1.  $x \in [0, \varepsilon]$  ( $\varepsilon$  是趋于零的数), 一般采用泰勒展开的方法,  $\cos x$  可展为:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \dots \quad (2.2)$$

则  $y = \frac{1}{2} - \frac{x^2}{4!} + \frac{x^4}{6!} \dots$ 。

2.  $x \in [\varepsilon, \infty]$ , 直接计算。

## 2.2 级数，连分式，阶乘

### 2.2.1 级数

$$f(x) = \sum_0^{\infty} a_k (x - x_0)^k \quad (2.3)$$

级数中每一项  $(x - x_0)^k$  可以从头算，为简化计算可以由前一项生成后一项，即  $(x - x_0)^{k-1} \rightarrow (x - x_0)^k$ ，同样阶乘也可以递推运算  $((k - 1)! \rightarrow k!)$ 。

不论如何算，一般级数的运算都要做截断。

### 2.2.2 连分式（格点格林函数）

$$f(x) = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \frac{a_4}{b_4 + \dots}}}} \quad (2.4)$$

对连分式的计算也要有一截断，令  $f_n$  表示对连分式求至系数  $a_n, b_n$  为止的结果，则有  $f_n = \frac{A_n}{B_n}$ ， $A_n, B_n$  递推关系：

$$A_j = b_j A_{j-1} + a_j A_{j-2}, \quad B_j = b_j B_{j-1} + a_j B_{j-2} \quad (2.5)$$

其中  $A_{-1} \equiv 1, B_{-0} \equiv 0, A_0 = b_0, B_0 \equiv 1, j = 1, 2, \dots, n$ 。（可以用数学归纳法证明）

### 2.2.3 阶乘（整数化为浮点数）

$$n! = 1 \cdot 2 \cdot 3 \dots \cdot n. \quad (2.6)$$

设  $y = n!$

$$\ln y = \ln(n!) = \ln 1 + \ln 2 + \dots + \ln n = k \quad (2.7)$$

$$y = e^k \quad (\text{指数形式}) \quad (2.8)$$

排列组合中

$$P_n^k = \frac{n!}{(n - k)!} \quad (2.9)$$

$$\ln P_n^k = \ln(n!) - \ln[(n - k)!] = w \quad (2.10)$$

$$y = e^w \quad (2.11)$$

## 第3章 方程求根

### 3.1 迭代法求根

#### 3.1.1 迭代法求根的步骤

求解方程  $f(x) = 0$  的根。

1. 可以先将其改写为  $x = g(x)$  ( $g(x)$  有很多种形式)。
2. 给定  $x$  一个初值  $x_0$ , 则可以得出如下式子

$$\begin{aligned}x_1 &= g(x_0) \\x_2 &= g(x_1) \\&\dots \\x_{i+1} &= g(x_i) \\&\dots\end{aligned}$$

直到  $x_i, x_{i+1}$  相同, 满足  $x = g(x)$ , 即  $f(x) = 0$  的根。(这个根可以视为迭代中的不动点)

例 3.12 求方程  $x^2 - 5x + 4 = 0$  的根。

1. 改写为  $x = \frac{x^2 + 4}{5} \Rightarrow x_{i+1} = \frac{x_i^2 + 4}{5}$

2. 猜一个初值

- 初值  $x = 2$ , 数列收敛, 可以得到近似根为 1。
- 初值  $x = 5$ , 数列发散, 得不到根。

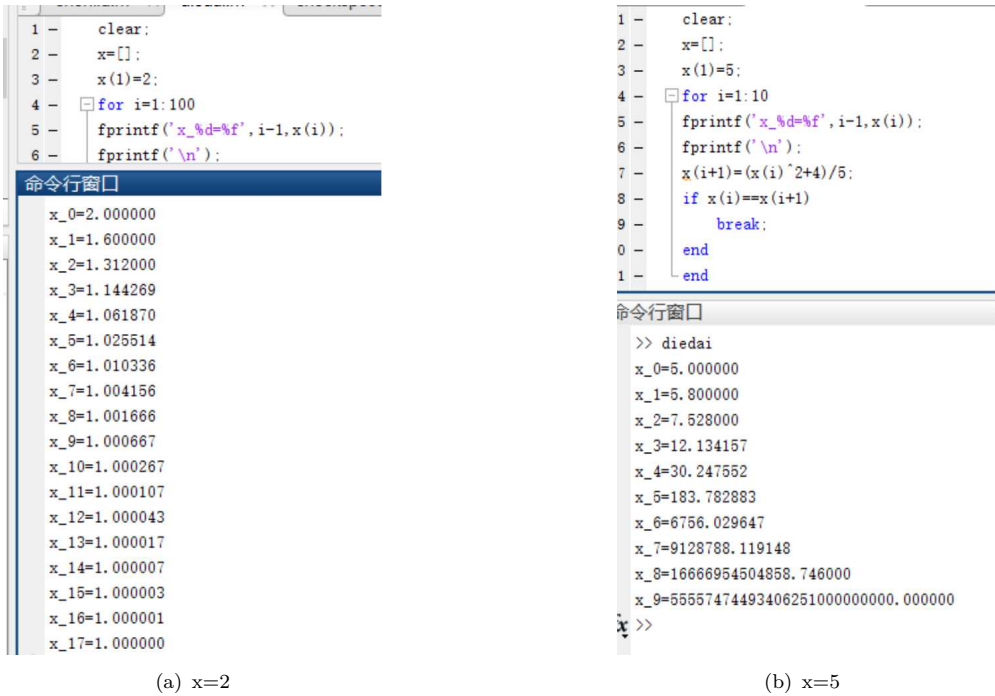


图 3.1: 不同初值运行结果

迭代法依赖：

- 1. 初始值的选取
- 2. 迭代形式， $x = g(x)$ ， $g(x)$  有很多写法

3.1.2 迭代法的收敛条件

平均值定理（中值定理？）

函数  $g(x)$  在区间  $[a,b]$  上有定义，则至少存在一点  $\xi(\xi \in [a,b])$ ，使得  $g'(x) = \frac{g(b) - g(a)}{b - a}$ ，即  $g(b) - g(a) = g'(\xi)(b - a)$ 。

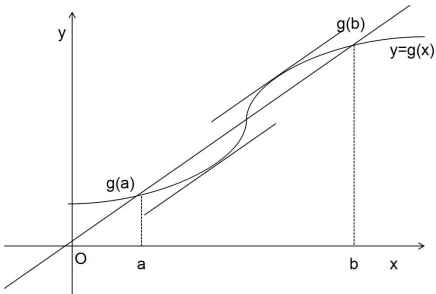


图 3.2: 平均值定理



迭代方程中  $x = g(x)$ ,  $x_i = g(x_{i-1})$ ,  $x_{i+1} = g(x_i)$ 。令  $a = x_{i-1}$ ,  $b = x_i$ , 利用平均值定理得

$$g(x_i) - g(x_{i-1}) = g'(\xi_i)(x_i - x_{i-1}), \quad \xi_i \in [x_{i-1}, x_i] \quad (3.1)$$

$$x_{i+1} - x_i = g'(\xi_i)(x_i - x_{i-1}) \quad (3.2)$$

取绝对值有

$$|x_{i+1} - x_i| = |g'(\xi_i)| |x_i - x_{i-1}| \quad (3.3)$$

全写出来有

$$|x_2 - x_1| = |g'(\xi_1)| |x_1 - x_0|$$

$$|x_3 - x_2| = |g'(\xi_2)| |x_2 - x_1|$$

...

$$|x_i - x_{i-1}| = |g'(\xi_{i-1})| |x_i - x_{i-1}|$$

假设  $g'(\xi)$  有限, 即  $|g(\xi)| \leq M$

$$\left. \begin{aligned} |x_2 - x_1| &\leq M|x_1 - x_0| \\ |x_3 - x_2| &\leq M|x_2 - x_1| \end{aligned} \right\} |x_3 - x_2| \leq M^2|x_1 - x_0|$$

...

$$|x_i - x_{i-1}| \leq M|x_i - x_{i-1}|$$

分析

1. 若  $M < 1$ ,  $x_0, x_1, x_2 \dots x_i$  是一定收敛的。

2. 若  $M \geq 1$ ,  $x_0, x_1, x_2 \dots x_i$  大多数情况是发散的

回到之前的例子:  $x^2 - 5x + 4 = 0$ , 迭代形式为  $x = g(x) = \frac{x^2 + 4}{5}$ ,  $g'(x) = \frac{2x}{5}$ , 其导数小于 1 的区间为  $[\frac{5}{2}, \frac{5}{2}]$ , 在此区间内取初值 (或者某一步的值落在这一区间) 可以得到收敛的数列进而得到所求的根, 而在此区间外取初值则会得到发散的数列得不到根。

在  $g'(x) > 1$  取初值, 但收敛的例子

$$g(x) : x = x^2 - 4x + 4 \quad (3.4)$$

$$g'(x) = 2x - 4 \quad (3.5)$$

取  $x_0 = 0$ ,  $g'(x_0) = -4$ ,  $|g'(x)| > 1$ 。此时  $x_1 = x_0^2 - 4x_0 + 4 = 4$  恰好为方程的根, 也就是得到的结果收敛, 但这是很特殊的情况。

### 3.1.3 迭代法的几何表示

迭代法:  $f(x) = 0 \Rightarrow x_{i+1} = g(x_i)$ 。自洽条件  $\bar{x} = g(\bar{x})$ ,  $\bar{x}$  为根。表现在几何图形上就是

$$\begin{cases} y = x \\ y = g(x) \end{cases} \quad \text{交点就是所求的根}$$

迭代法的具体过程也可以在几何上表现 (先取一初值  $x_0$ , 找到其对应的  $y=g(x)$  上的点, 再在  $y=x$  上找到纵坐标与其相同的点, 其横坐标即为  $x_1$ , 如此进行下去, 可能得到根, 也可能发散), 如图所示几种情况。

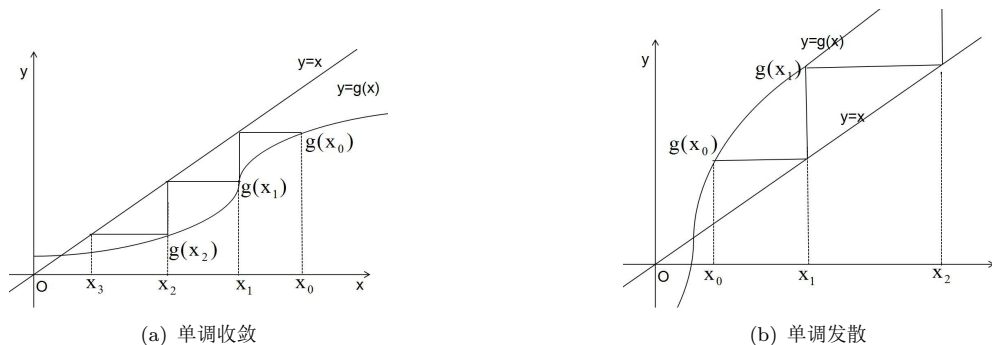


图 3.3: 单调情况

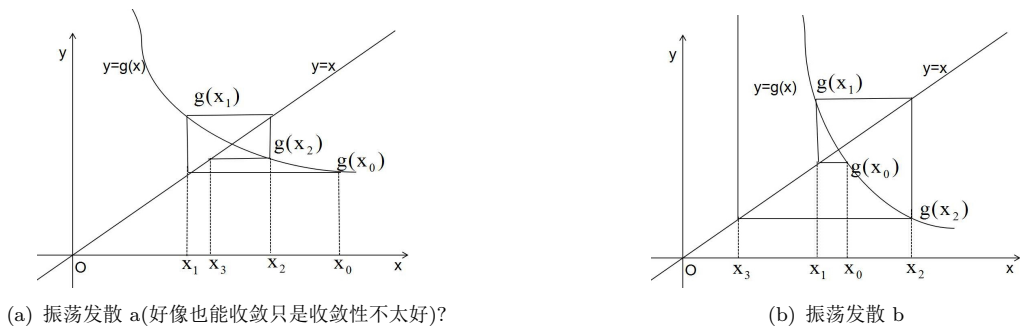


图 3.4: 振荡情况

## 3.2 牛顿迭代法和弦截法

### 3.2.1 牛顿迭代法

$x = g(x)$ ,  $g(x)$  取特殊形式。

取两个相邻的点 (应该是迭代过程中先后算出的两点)  $x_k, x_{k+1}$ ,  $x_{k+1} = x_k + \Delta x$ 。  $f(x_{k+1})$  在  $x_k$  附近展开得

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + \dots \quad (3.6)$$

令  $f(x_{k+1}) = 0$  (原因如下, 可能不是十分明了), 取前两项 (近似嘛, 一般都这样, 但是  $\Delta x$

是否很小?)

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0 \quad (3.7)$$

从几何上来看相当于  $x$  轴沿  $y$  方向平移了  $f(x_{k+1})$ , 可以看成  $f(x_{k+1}) = 0$ 。(可以理解为取  $x_{k+1}$  处本征坐标系)

解得

$$x_{k+1} = x_k - \frac{f(x_k) - f(x_{k+1})}{f'(x_k)} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (3.8)$$

对于第一项

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \Rightarrow f'(x_0) = -\frac{f(x_0)}{x_1 - x_0} \quad (3.9)$$

其几何意义明显, 是逐渐向根趋近的。

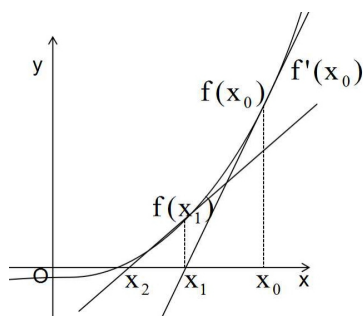
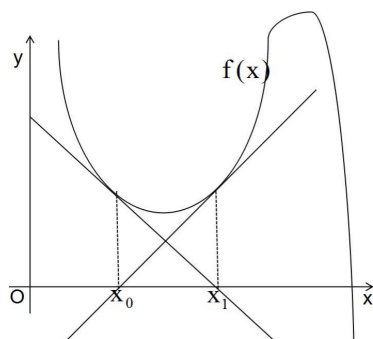


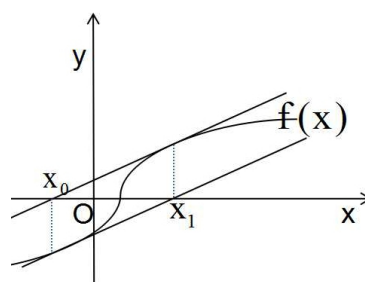
图 3.5: 牛顿迭代法

牛顿迭代法的弱点:

1. 存在不收敛情况, 如图



(a) 交叉



(b) 平行

图 3.6: 无根极端情况

2. 导数不易计算

### 3.2.2 弦截法

几种理解:

1. 作为牛顿迭代法的改进, 导数用差商代替:

$$f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \quad (\text{向后差商}) \quad (3.10)$$

代入到  $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ , 得

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) \quad (3.11)$$

这个差商的选取较为特殊。

2. 直接推导弦截法

取  $f(x)$  上的两点  $(x_0, f(x_0)), (x_1, f(x_1))$  作直线(弦), 其方程为:  $y = f(x_1) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_1)$ 。

令  $y=0$ , 解出  $y=0$  的解  $x$ , 作为  $x_2$ 。  $x_2 = x_1 - \frac{f(x_1)}{f(x_1) - f(x_0)}(x_1 - x_0)$ 。

验证: 若  $f(x_2) = 0$ ,  $x_2$  为根, 否则重复上述过程。

$x_3 = x_2 - \frac{f(x_2)}{f(x_1) - f(x_2)}(x_2 - x_1)$ , 一般地, 有

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1}) \quad (\text{与之前的推导一致}) \quad (3.12)$$

令  $x_{k-1} = x_1$  (即固定一点, 也可以不固定, 但比较麻烦), 方程化为

$$x_{k+1} = x_k - \frac{x_k - x_1}{f(x_k) - f(x_1)} f(x_k) \quad (3.13)$$

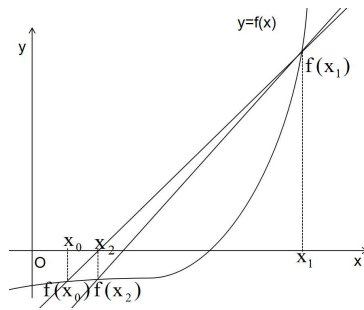


图 3.7: 弦截法

### 3.3 二分法

$f(x)=0$ , 在  $[A,B]$  区间求根。

1. 确定有根的小区间

从  $A$  开始, 令  $a=A$ , 步长为  $H$ ,  $b=a+H$ , 构造小区间  $[a,b]$

- 若  $f(a)$ ,  $f(b)$  同号,  $[a,b]$  区间内无根
- 若  $f(a)$ ,  $f(b)$  异号,  $[a,b]$  区间内有根

再作下部,  $a=b$ ,  $b=a+H$ , 重复上述步骤。

## 2. 二分法求区间 $[a,b]$ 的根

$[a,b]$  分成  $[a, \frac{a+b}{2}]$ ,  $[\frac{a+b}{2}, b]$

计算  $f(\frac{a+b}{2})$

- 若  $f(\frac{a+b}{2}) = 0$ , 则  $\frac{a+b}{2}$  为根。
- 若  $f(\frac{a+b}{2})$  与  $f(a)$  异号,  $[a, \frac{a+b}{2}]$  有根。
- 若  $f(\frac{a+b}{2})$  与  $f(b)$  异号,  $[\frac{a+b}{2}, b]$  有根。

无根区间舍去, 有根区间  $[a,b] \rightarrow [a_1,b_1] \rightarrow \dots [a_n,b_n]$ , 当  $n$  很大时, 根近似为  $x = \frac{a_n + b_n}{2}$ 。

二分法弱点

1. 步长  $H$  的选取, 太大会漏掉根, 太小计算量大;
2. 在  $[A,B]$  区间范围都要寻找有根区间。

## 3.4 “笨”方法

解方程  $f(x)=0$ , 有时可以大致知道解得区间范围 (例如费米能), 可以直接在这个大致区间里取点挨个试。

How to research? Theoretical

John Bardeen (最正宗)

主要贡献: BCS 理论, 半导体放大效应

1. 物理机制 (mechanism)
2. Building model 实现想法



# 第4章 函数插值和数值微分

拟合

## 4.1 拉格朗日插值

已知一个比较复杂的函数  $y = f(x)$ ，现构造一个简单函数  $\varphi(x)$ （一般是多项式形式），用来作为  $y = f(x)$  的近似表达式，即  $f(x) \approx \varphi(x)$ ，但是要使得在某些特殊点满足  $\varphi(x_0) = y_0, \varphi(x_1) = y_1, \dots, \varphi(x_n) = y_n$ ，这里  $y_i = f(x_i)$ ， $f(x)$  称为被插函数， $\varphi(x)$  称为插值函数。

### 4.1.1 线性插值

x	$x_0$	$x_1$
y	$y_0$	$y_1$

表 4.1: 线性插值取点

只取两个特殊点，可以得到线性函数  $\varphi(x) = ax + b$

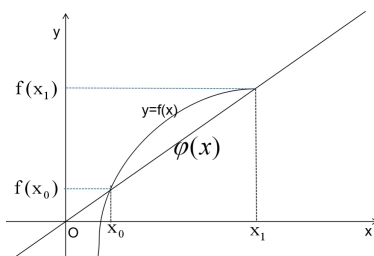


图 4.1: 线性插值几何表示

这种插值方法虽然简单，但是应用广泛，比如在神经网络的单个神经元中就有这种线性插值，

$$\begin{aligned} y &= ax + b \\ z &= \theta(y - c) \end{aligned} \tag{4.1}$$

其中  $\theta$  为阶跃函数。

可以看出阶跃函数很像零温时的费米函数（费米面）。



图 4.2: 神经网络

多个神经元，有如下关系

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (4.2)$$

上述参数要通过一些“训练”来确定，几乎所有的关系包含在系数矩阵中。

回到线性插值，我们可以通过特殊的解出  $a, b$

$$\begin{cases} \varphi_1(x_0) = y_0 \\ \varphi_1(x_1) = y_1 \end{cases} \Rightarrow \begin{cases} a = \frac{y_1 - y_0}{x_1 - x_0} \\ b = y_0 - \frac{y_1 - y_0}{x_1 - x_0} x_0 \end{cases} \quad (4.3)$$

将  $\varphi_1(x) = ax + b$  改写一下

$$\varphi_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0} \quad (4.4)$$

令  $l_0 = \frac{x - x_1}{x_0 - x_1}, l_1 = \frac{x - x_0}{x_1 - x_0}$ ,  $l_0, l_1$  可以看成是基函数（基底）。类似于量子力学中的本征函数系，有两种看法

1. 已知某一状态，向本征态上展开

$$\varphi(x) \Rightarrow \sum_n a_n \phi_n(x) \quad (4.5)$$

2. 只知道本征态，设某一状态是本征态的叠加

$$\tilde{\varphi}(x) \Leftarrow \sum_n a_n \phi_n(x) \quad (4.6)$$

线性插值基底的表示

	$x_0$	$x_1$
$l_0(x)$	1	0
$l_1(x)$	0	1

表 4.2: 线性插值基底的数值表示



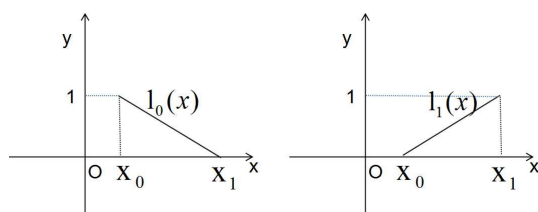


图 4.3: 线性插值基底的几何表示

### 4.1.2 二次插值

二次插值是取三个特殊点

x	$x_0$	$x_1$	$x_2$
y	$y_0$	$y_1$	$y_2$

表 4.3: 二次插值的特殊点

这样的三个特殊点，可以设抛物线： $\varphi_2(x) = a_0 + a_1x + a_2x^2$ ，代入特殊点得

$$\begin{cases} y_0 = a_0 + a_1x_0 + a_2x_0^2 \\ y_1 = a_0 + a_1x_1 + a_2x_1^2 \\ y_2 = a_0 + a_1x_2 + a_2x_2^2 \end{cases} \quad (4.7)$$

可以解出  $a_0, a_1, a_2$ ，用基函数表示为

$$\varphi_2(x) = y_0l_0(x) + y_1l_1(x) + y_2l_2(x) \quad (4.8)$$

其中基底为

$$l_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \quad (4.9a)$$

$$l_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \quad (4.9b)$$

$$l_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \quad (4.9c)$$

基函数的几何形式

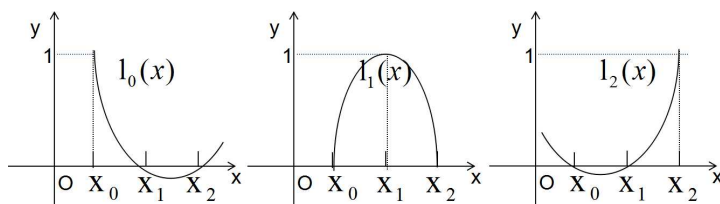


图 4.4: 基函数的几何形式

### 4.1.3 n 次插值

n 次插值是取 n+1 个特殊点，如下所示

x	$x_0$	$x_1$	...	$x_n$
y	$y_0$	$y_1$	...	$y_n$

表 4.4: n 次插值的特殊点

这里我们可以总结出对于所设多项式的要求

1. 次幂不高于 n

2.  $\varphi_n(x_i) = y_i$

设  $\varphi_n = a_0 + a_1x + \dots + a_nx^n$ ，其满足

$$\varphi_n(x) = \sum_j^n y_j l_j(x) \quad (4.10)$$

其中  $l_j(x)$  为基底，有如下形式

$$l_j(x) = \frac{(x - x_0) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)} \quad (4.11)$$

计算后有如下结果

	$x_0$	$x_1$	...	$x_n$
$l_0(x)$	1			
$l_1(x)$		1		
$\vdots$			$\ddots$	
$l_n(x)$				1

### 4.1.4 Runge 现象和分段线性插值

例 4.13  $f(x) = \frac{1}{1 + 25x^2}, [-1, 1]$ ，用多项式作为简单插值函数。

我们写一个 10 次的多项式，特殊点取区间内 10 等分点和端点，有如下结果

$$\varphi_{10} = \sum_{i=1}^{10} f(x_i) l_i(x), \quad x_i = -1 + \frac{2}{10}i \quad (i = 0, 1, 2, \dots, 10) \quad (4.12)$$

其中

$$l_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_{10})}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \quad (4.13)$$

被插函数和插值函数的图像对比如下：

可以看出误差比较大。

分析可以知道：多项式  $\varphi = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ ，n 很大时， $\varphi$  很“柔软”，容易出现拐点，这种现象称为 Runge 现象。因此不是 n 越大，近似越好。

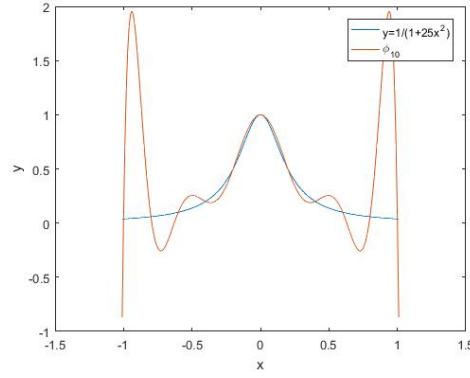


图 4.5: Runge 现象

简单有效的解决方式就是分段线性插值: 在  $[a, b]$  区间取点 (一般取等分点)  $a = x_0 < x_1 \dots < x_n = b$ , 将其分成许多小区间,  $\varphi(x)$  取每一小段上的线性插值函数。

多项式拟合的缺陷:

- 多项式不能拟合渐进行为 (奇异行为)
- 存在 Runge 现象

## 4.2 数值微分

对于一复杂函数  $f(x)$  作其插值函数  $\varphi(x)$ , 即  $f(x) \approx \varphi(x)$ , 可以认为  $f'(x) = \varphi'(x)$ 。

### 4.2.1 两点公式

两点公式即过两点插值函数

$$\varphi'_1(x_0) = \varphi'_1(x_1) = \frac{y_1 - y_0}{h}, \quad h = x_1 - x_0 \quad (4.14)$$

所以被插函数的导数为

$$f'(x_0) = f'(x_1) = \frac{y_1 - y_0}{h} \quad (4.15)$$

### 4.2.2 三点公式

三点公式即过三点插值函数, 假设这三点之间等距。

$$\varphi_2(x) = \frac{(x - x_1)(x - x_2)}{2h^2}y_0 - \frac{(x - x_0)(x - x_2)}{h^2}y_1 + \frac{(x - x_0)(x - x_1)}{2h^2}y_2 \quad (4.16)$$

可以得到被插函数的导数

$$f'(x_0) = \varphi'_2(x_0), \quad f'(x_1) = \varphi'_2(x_1), \quad f'(x_2) = \varphi'_2(x_2) \quad (4.17)$$

### 4.3 厄米特插值

厄米特插值 就是由函数值本身 + 函数导数构成的多项式。具体如下

1. 构造基态（基函数）

$$2. H(x) = y_0 h_0(x) + y_1 h_1(x) + m_0 H_0(x) + m_1 H_1(x)$$

基底如下

	$x_0$	$x_1$	$x_0$	$x_0$
$h_0(x)$	1			
$h_1(x)$		1		
$H_0(x)$			1	
$H_1(x)$				1

### 4.4 实验数据拟合与函数插值

物理中拟合的方法为最小二乘法（最常用方法），其目的就是找到数据的趋势（变化规律）。具体步骤如下：

1. 猜曲线形状：  $y = ax + b, y = \alpha e^{\beta x}, \dots$

2. 计算拟合曲线  $y$  与实验值  $y^*$  的差方（以线性函数为例）：

$$Q = \sum_{i=1}^N (y_i^* - y_i)^2 \quad (4.18)$$

其中  $Q$  为  $(a, b)$  的函数。

3. 通过极小值条件求参（优化）

$$\begin{cases} \frac{\partial Q}{\partial a} = 0 \\ \frac{\partial Q}{\partial b} = 0 \end{cases} \Rightarrow \text{解出 } a, b, \text{ 进而得出拟合曲线} \quad (4.19)$$

上述只通过一组实验求一次导数，便得出结果，实际过程中（如神经网络）通常是多次求解，不断优化。

# 第5章 数值积分

前面提到了插值函数用多项式  $\varphi(x)$  来拟合复杂函数  $f(x)$ ，同样可以拟合导数  $f'(x) = \varphi'(x)$ ，可以推断其也可以拟合积分  $\int_a^b f(x)dx = \int_a^b \varphi(x)dx$ 。

通常认为数值积分扩大了我们的计算范围。

积分可分为如下两种：

- 正常积分——简单算法
- 反常积分（广义积分）——特殊处理

用多项式拟合被积函数有如下缺陷：

1. 多项式幂次很高时， $\varphi(x)$  会变得“柔软”，拟合不准确，即出现 Runge 现象——这通常出现在正常积分中；
2. 多项式拟合不能描述奇异行为（渐进行为）——通常出现在反常积分中。

## 5.1 梯形积分法

### 5.1.1 梯形积分法

计算积分  $\int_a^b f(x)dx$

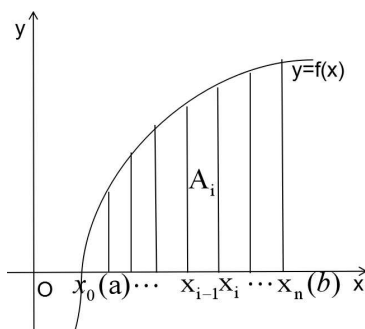


图 5.1: 梯形积分法

可以将区间  $[a,b]$  等距离划分为  $n$  份，总的积分为每个小梯形面积之和：

$$T(n) = \sum_{i=1}^n n_i, \quad n_i = \frac{\omega}{2} [f(x_{i-1}) + f(x_i)], \quad \omega = \frac{b-a}{n} \quad (5.1)$$

考虑是否  $n$  越大越好？（之前提到过）

## 另一推导（分段线性插值）

考虑小区间内的线性拟合：

$$\varphi(x) = f(x_{i-1}) \frac{x - x_i}{x_{i-1} - x_i} + f(x_i) \frac{x - x_{i-1}}{x_i - x_{i-1}} \quad (5.2)$$

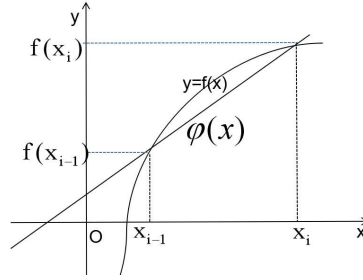


图 5.2: 线性插值

则积分为：

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx \int_{x_{i-1}}^{x_i} \varphi(x) dx = \int_{x_{i-1}}^{x_i} f(x_{i-1}) \frac{x - x_i}{x_{i-1} - x_i} + f(x_i) \frac{x - x_{i-1}}{x_i - x_{i-1}} dx = \frac{\omega}{2} [f(x_{i-1}) + f(x_i)] \quad (5.3)$$

这样我们就得到了梯形积分公式，上式中的被积函数为常数和一次项的线性组合，则公式严格成立。

### 5.1.2 截断误差

小区间内的积分精确计算得到  $I_i = \int_{x_{i-1}}^{x_i} f(x) dx = F(x_i) - F(x_{i-1})$ ，其中  $F(x)$  为  $f(x)$  的原函数，即  $f(x) = F'(x)$ 。

数值计算  $A_i = \frac{\omega}{2} [f(x_{i-1}) + f(x_i)]$

一个小梯形的截断误差：

$$E_i = A_i - I_i \quad (5.4)$$

为计算这个误差，我们将  $f(x_{i-1})$  以  $x_i$  为展开中心作泰勒展开：( $x_{i-1}x_i$  很接近)

$$\begin{aligned} f(x_{i-1}) &= f(x_i) + f'(x_i)(x_{i-1} - x_i) + \frac{1}{2!} f''(x_i)(x_{i-1} - x_i)^2 + \frac{1}{3!} f'''(x_i)(x_{i-1} - x_i)^3 + \dots \\ &= f(x_i) - f'(x_i)\omega + \frac{1}{2!} f''(x_i)\omega^2 - \frac{1}{3!} f'''(x_i)\omega^3 + \dots \end{aligned} \quad (5.5)$$

同样我们可以得到  $F(x_{i-1})$  的展开式：

$$\begin{aligned} F(x_{i-1}) &= F(x_i) + F'(x_i)(x_{i-1} - x_i) + \frac{1}{2!} F''(x_i)(x_{i-1} - x_i)^2 + \frac{1}{3!} F'''(x_i)(x_{i-1} - x_i)^3 + \dots \\ &= F(x_i) - f(x_i)\omega + \frac{1}{2!} f'(x_i)\omega^2 - \frac{1}{3!} f''(x_i)\omega^3 + \dots \end{aligned} \quad (5.6)$$

这样我们可以改写  $A_i, I_i$

$$\begin{aligned}
 A_i &= \frac{\omega}{2} [f(x_{i-1}) + f(x_i)] \\
 &= \frac{\omega}{2} [f(x_i) + f(x_i) - f'(x_i)\omega + \frac{1}{2!}f''(x_i)\omega^2 - \frac{1}{3!}f'''(x_i)\omega^3 + \dots] \\
 &= f(x_i)\omega - \frac{1}{2}f'(x_i)\omega^2 + \frac{1}{2 \cdot 2!}f''(x_i)\omega^3 + \dots
 \end{aligned} \tag{5.7}$$

$$\begin{aligned}
 I &= F(x_i) - F(x_{i-1}) \\
 &= F(x_i) - F(x_i) + f(x_i)\omega - \frac{1}{2!}f'(x_i)\omega^2 + \frac{1}{3!}f''(x_i)\omega^3 + \dots \\
 &= f(x_i)\omega - \frac{1}{2!}f'(x_i)\omega^2 + \frac{1}{3!}f''(x_i)\omega^3 + \dots
 \end{aligned} \tag{5.8}$$

所以小区间中的误差为:

$$E_i = A_i - I_i = f''(x_i)\omega^3\left(\frac{1}{4} - \frac{1}{6}\right) + O(\omega^4) \tag{5.9}$$

在  $[a, b]$  区间内, 总的误差为:

$$E_T = \sum_{i=1}^n E_i = \sum_{i=1}^n \frac{\omega^3}{12} f''(x_i) \tag{5.10}$$

假设  $f''(x_i)$  有界, 即存在  $M > 0$ , 使得  $|f''(x_i)| \leq M$ , 则误差可表示为:

$$E_T \leq n \frac{\omega^3}{12} M = (b-a) \frac{\omega^2}{12} M \tag{5.11}$$

可以看出截断误差  $\propto \omega^2$ , 可以预期当  $\omega \rightarrow \frac{\omega}{2}$  时, 即  $n \rightarrow 2n$  时, 截断误差或小于原来的  $\frac{1}{4}$ 。

实际数值计算时的误差还要考虑舍入误差, 以下展示了随取点数的增加计算误差的变化, 由之前的分析我们知道, 这实际上是截断误差和舍入误差竞争的结果。

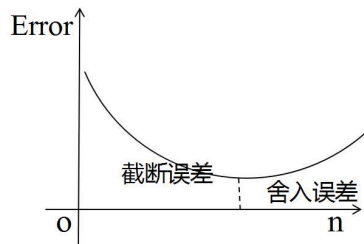


图 5.3: 梯形积分法计算的误差

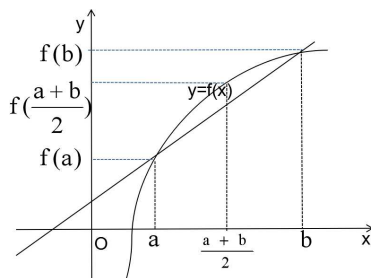


图 5.4: 梯形积分的形式

### 5.1.3 其他简单形式

$$\int_a^b f(x)dx = f(a)(b-a) \quad (5.12a)$$

$$\int_a^b f(x)dx = f\left(\frac{a+b}{2}\right)(b-a) \quad (5.12b)$$

$$\int_a^b f(x)dx = f(b)(b-a) \quad (5.12c)$$

### 5.1.4 数值积分公式

以上我们可以看出数值积分的公式都是取积分区间点  $x_k$  处的  $f(x_k)$  加权  $A_k$  后求和（几乎都是这种形式），我们将在后面看到更多这样的形式：

$$\int_a^b f(x)dx = \sum_{k=1}^n A_k f(x_k) \quad (5.13)$$

## 5.2 辛普森 (Simpson) 积分法

### 5.2.1 辛普森积分方法

前面的梯形积分法，可以看成取两点进行线性插值的结果，我们可以预期取更多的点进行插值近似。当取三个点，即进行二次插值时，我们可以获得辛普森积分公式。

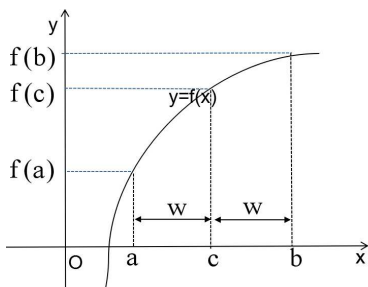


图 5.5: 辛普森积分法

对于积分  $\int_a^b f(x)dx$ ，我们过  $(a, f(a)), (b, f(b)), (c, f(c))$  作插值函数（取  $c$  为中点），由前面的



知识我们知道

$$P(x) = f(a) \frac{(x-c)(x-b)}{(a-c)(a-b)} + f(b) \frac{(x-a)(x-c)}{(b-a)(b-c)} + f(c) \frac{(x-a)(x-b)}{(c-a)(c-b)} \quad (5.14)$$

积分得:

$$\int_a^b f(x)dx = \int_a^b P(x)dx = \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (5.15)$$

我们进一步将  $[a, b]$   $n$  等分,  $n$  为偶数 (保证每一组有三个点), 令  $w = \frac{b-a}{n}$ , 我们可以得到每一份的面积 (积分) 为:

$$A_i = \frac{w}{3} [f(x_{i-1}) + 4f(x_i) + f(x_{i+1})], \quad w = x_i - x_{i-1} \quad (5.16)$$

最终的积分为:

$$T_n = \sum_{i=1}^{n-1} A_i \quad (i \text{ 为奇数}) \quad (5.17)$$

由于是二次插值, 所以上式对  $x^2, x, c$  的线性组合严格成立。

### 5.2.2 截断误差

采用和梯形积分中同样的分析方法。

先分析一小段内的误差 ( $A_i$  为近似值,  $I_i$  为真实值):

$$E_i = A_i - I_i, \quad [x_{i-1}, x_{i+1}] \quad (5.18)$$

$$A_i = \frac{w}{3} [f(x_{i-1}) + 4f(x_i) + f(x_{i+1})] \quad (5.19)$$

$$I_i = \int_{x_{i-1}}^{x_{i+1}} f(x)dx = F(x_{i+1}) - F(x_{i-1}), \quad (F \text{ 为原函数}) \quad (5.20)$$

在  $x_i$  处作泰勒展开 (只展开到 4 次项) 得

$$A_i = \frac{w}{3} \left[ 6f(x_i) + 2 \cdot \frac{f^{(2)}(x_i)}{2!} w^2 + 2 \cdot \frac{f^{(4)}(x_i)}{4!} w^4 \right] = 2f(x_i)w + \frac{2f^{(2)}(x_i)}{3 \cdot 2!} w^3 + \frac{2f^{(4)}(x_i)}{3 \cdot 4!} w^5 \quad (5.21)$$

$$I_i = 2F^{(1)}(x_i)w + 2 \frac{F^{(3)}(x_i)}{3!} w^3 + 2 \frac{F^{(5)}(x_i)}{5!} w^5 = 2f(x_i)w + \frac{2f^{(2)}(x_i)}{3!} w^3 + \frac{2f^{(4)}(x_i)}{5!} w^5 \quad (5.22)$$

两式相减得:

$$E_i = A_i - I_i = \frac{2f^{(4)}(x_i)}{3 \cdot 4!} w^5 - \frac{2f^{(4)}(x_i)}{5!} w^5 = \frac{f^{(4)}(x_i)}{90} w^5 \quad (5.23)$$

考虑  $f^{(4)}(x_i)$  有界, 即存在  $M > 0$ , 使得  $|f^{(4)}(x_i)| \leq M$ , 上式可化为

$$E_i \leq \frac{w^5}{90} M \quad (5.24)$$

总的误差为

$$E_T = \sum_{i=\text{奇数}}^n \leq \frac{n}{2} \frac{w^5}{90} M = (b-a) \frac{w^4}{180} M \quad (5.25)$$

可以看出  $E_T \propto w^2$ , 当  $n \rightarrow 2n$  时, 误差减小为原来的  $\frac{1}{16}$ 。同样实际计算时要考虑舍入误差。

### 5.2.3 柯特斯公式

在  $[a, b]$  区间进一步取 5 个点作为插值函数, 再积分。我们直接给出结果:

$$A = \frac{b-a}{90} [2f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 2f(x_4)] \quad (5.26)$$

其中  $x_k = a + kh, k = 0, 1, 2, 3, 4, h = \frac{b-a}{4}$

同样可以推出截断误差  $\propto w^6$ 。

前面的数值积分方法都是用多项式“拟合”被积函数, 之前的分析也说明了不是取的点数越多越好, 除了对于计算机要考虑舍入误差外, 还有 Runge 现象 (即被积函数拟合的多项式变软)。

对于 Runge 现象有两点分析

1. 直观地, 从面积角度看差距很大

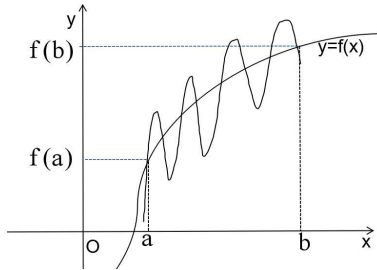


图 5.6: 数值积分中的 Runge 现象

2. 不直观,  $\int_a^b f(x)dx = \int_a^b g(x)dx = \sum_{k=1}^n A_k f(x_k)$ ,  $x_k$  为插值的节点。Runge 现象影响不是通过  $f(x_k)$  (这是固定的), 而是通过  $A_k$  (前面的权重) 实现的。

现在我们知道单纯的增加节点数是行不通的, 要寻找新的方法, 下面我们要介绍以下两种方法

1. 外推法: 梯形积分  $\rightarrow$  辛普森积分  $\rightarrow$  柯特斯积分  $\rightarrow$  更高次积分
2. 高斯积分法: 待定系数

## 5.3 龙贝格 (Romberg) 积分法

利用梯形积分公式计算在不同  $w(w, \frac{w}{2}, \frac{w}{4}, \dots)$ , 即  $n$  不同时, 对应的积分值, 通过递推关系 (外推法) 得出更精确的结果, 称为龙贝格积分法。

对于梯形积分公式 (记为 0 阶) 截断误差  $E_T \leq \frac{b-a}{12} w^2 M \propto w^2$  ( $w$  记为 0 次截断, 初始值应该是自己选的), 进一步推得  $w' = \frac{w}{2}$ ,  $E'_T = \frac{1}{4} w^2$  ( $\frac{w}{2}$  记为 1 次截断, 即  $\frac{w}{2^i}$  对易  $i$  次截断, 之后的标记用得到)

现在令  $I$  为积分精确值,  $T_{00}$  表示梯形积分取区间间隔为  $w$  时的积分, 之后的依次类推。我们可以得到:

$$\begin{aligned} T_{00} \cdots \frac{w}{2^i} (i=0) \\ T_{01} \cdots \frac{w}{2^i} (i=1) \\ T_{02} \cdots \frac{w}{2^i} (i=2) \\ \dots \end{aligned} \quad (5.27)$$

误差分别为:  $E_{00} = T_{00} - I, E_{01} = T_{01} - I, \dots$

Romberg 认为:  $E_{01} \approx \frac{E_{00}}{4}$ , 这样我们得到

$$T_{01} - I = \frac{T_{00} - I}{4} \Rightarrow I = \frac{4T_{01} - T_{00}}{3} = T_{11} \quad (5.28)$$

可以证明上式对应于步长为  $\frac{w}{2}$  的辛普森积分公式, 记为  $T_{11}$ 。证明如下:  
取三个节点  $x_0, x_1, x_2$ 。

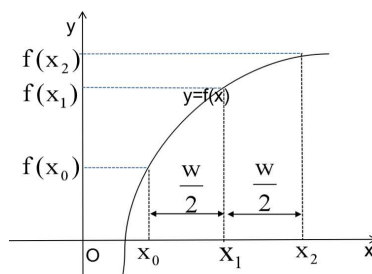


图 5.7: 龙贝格积分法的推导

只用一个梯形计算:

$$T_{00} = \frac{w}{2} [f(x_0) + f(x_2)] \quad (5.29)$$

区间划分增加一个节点 (取中点), 即用两个梯形计算:

$$T_{01} = \frac{w}{2} [f(x_0) + f(x_1)] + \frac{w}{2} [f(x_1) + f(x_2)] = \frac{w}{4} [f(x_0) + 2f(x_1) + f(x_2)] \quad (5.30)$$

代入得

$$\begin{aligned} \frac{4T_{01} - T_{00}}{3} &= \frac{1}{3} \{w[f(x_0) + 2f(x_1) + f(x_2)] - \frac{w}{2} [f(x_0) + f(x_2)]\} \\ &= \frac{w}{6} [f(x_0) + 4f(x_1) + f(x_2)] = T_{11} \end{aligned} \quad (5.31)$$

可以看出最后的结果恰好是积分区间为  $\frac{w}{2}$  的辛普森积分公式。进一步我们可以知道  $\frac{4T_{0i} - T_{0i-1}}{3} = T_{1i}$ 。

对于辛普森积分, 截断误差  $E_S \leq \frac{b-a}{180} w^4 M \propto w^4$ , 同样按之前的思路我们有

$$\begin{aligned} T_{11} \cdots \frac{w}{2^i} (i=1) \\ T_{12} \cdots \frac{w}{2^i} (i=2) \\ \dots \end{aligned} \quad (5.32)$$

误差分别为:  $E_{11} = T_{11} - I, E_{12} = T_{12} - I, \dots$

同样认为  $E_{12} \approx \frac{1}{16} E_{11}$ , 可以推出:

$$T_{12} - I = \frac{T_{11} - I}{16} \Rightarrow I = \frac{16T_{12} - T_{11}}{15} = T_{22} \quad (5.33)$$

进一步的推导我们有

$$T_{ki} = \frac{4^k T_{k-1i} - T_{k-1i-1}}{4^k - 1} = \frac{4^k c_{k-1} - b_{k-1}}{4^k - 1} = d_k \quad (5.34)$$

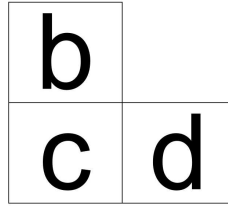


图 5.8: bcd

当  $|T_{kk} - T_{k-1k-1}| < \varepsilon$  (认为选定的极限) 时, 认为  $T_{kk}$  为近似值。

龙贝格积分方法仍是多项式拟合, 只是避免了 Runge 现象。

外推次数 复化次数	梯形积 分序列 0	辛普森积 分序列 1	柯特斯积 分序列 2	龙贝格积 分序列 3	...
$w$	$T_{00}$				
$\frac{w}{2}$	$T_{01}$	$T_{11}$			
$\frac{w}{4}$	$T_{02}$	$T_{12}$	$T_{22}$		
$\frac{w}{8}$	$T_{03}$	$T_{13}$	$T_{23}$	$T_{33}$	
$\frac{w}{16}$	$T_{04}$	$T_{14}$	$T_{24}$	$T_{34}$	$T_{44}$
截断误差 $\propto$	$w^2$	$w^4$	$w^6$	$w^8$	$w^{10}$

表 5.1: 龙贝格积分法各项

## 5.4 高斯积分法

由前面的分析我们知道梯形积分，辛普森积分，龙贝格积分都有如下形式：

$$\int_a^b f(x)dx = \sum_{k=1}^n A_k f(x_k) \quad (5.35)$$

我们知道以上都是取等间距划分的区间，即  $x_k$  是固定的，同时权重  $A_k$  也是固定的。

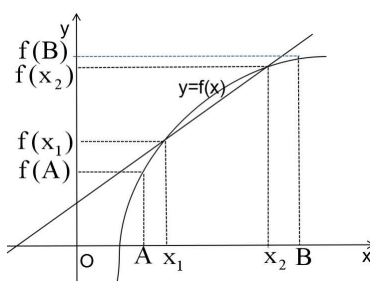


图 5.9: 高斯积分法

而高斯积分法中的权重则是不确定的，例如两点高斯积分公式  $A = c_1 f(x_1) + c_2 f(x_2)$ ，可以看出有如下特点：

1. 高斯积分承认并推广了数值积分形式：

$$\int_a^b f(x)dx = \sum_{k=1}^n A_k f(x_k) \quad (5.36)$$

2. 推导方法做改进：发展了待定系数法确定权重和节点。

高斯积分法适用范围广。

### 5.4.1 两点高斯积分

先将积分区间  $[a, b]$  变化为  $[-1, 1]$ :

$$\int_a^b \tilde{f}(x)dx = \int_{-1}^1 f(x)dx \quad (5.37)$$

考虑区间  $[-1, 1]$  上  $\int_{-1}^1 f(x)dx$  的数值积分公式。

前面我们已经提到了两点高斯积分公式

$$A = c_1 f(x_1) + c_2 f(x_2) \quad (5.38)$$

取  $f(x)$  特殊的函数形式（一般还是多项式，因为多项式的普适性）确定  $x_1, x_2, c_1, c_2$ 。

共有 4 未知数，我们取 3 次多项式：（多项式拟合被积函数）

$$f(x) = ax^3 + bx^2 + cx + d, \quad (a, b, c, d \text{ 是任意的}) \quad (5.39)$$

使得

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 f(x) dx \quad (5.40)$$

精确成立。代入得

$$\begin{aligned} & c_1(ax_1^3 + bx_1^2 + cx_1 + d) + c_2(ax_2^3 + bx_2^2 + cx_2 + d) \\ &= a \int_{-1}^1 x^3 dx + b \int_{-1}^1 x^2 dx + c \int_{-1}^1 x dx + 2d \\ & a(c_1 x_1^3 + c_2 x_2^3) + b(c_1 x_1^2 + c_2 x_2^2) + c(c_1 x_1 + c_2 x_2) + d(c_1 + c_2) \\ &= a \cdot 0 + b \cdot \frac{2}{3} + c \cdot 0 + d \cdot 2 \end{aligned} \quad (5.41)$$

因为  $a, b, c, d$  是任意的, 因此系数对应相等。

$$\begin{cases} c_1 x_1^3 + c_2 x_2^3 = 0 \\ c_1 x_1^2 + c_2 x_2^2 = \frac{2}{3} \\ c_1 x_1 + c_2 x_2 = 0 \\ c_1 + c_2 = 2 \end{cases} \text{ 解得 } \begin{cases} x_1 = \frac{1}{\sqrt{3}} \\ x_2 = -\frac{1}{\sqrt{3}} \\ c_1 = 1 \\ c_2 = 1 \end{cases} \quad \begin{cases} x_1 = -\frac{1}{\sqrt{3}} \\ x_2 = \frac{1}{\sqrt{3}} \\ c_1 = 1 \\ c_2 = 1 \end{cases} \quad (5.42)$$

由第一式和第二式得  $x_1^2 = x_2^2$ , 这样方便求解。

这样我们的到两点积分公式

$$A = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \quad (5.43)$$

可以看出这个积分公式是对三次多项式 ( $x^3, x^2, x, c$  的线性组合) 严格成立的。这种积分方法中节点数少, 但精度提高了。

### 5.4.2 n 点高斯积分

同样将积分区间  $[a, b]$  变化为  $[-1, 1]$ , 对于  $n$  点积分公式我们有:

$$\int_{-1}^1 f(x) dx = \sum_{k=1}^n A_k f(x_k) \quad (5.44)$$

同样我们取特殊函数 ( $m$  次多项式  $f(x) = a_m x_m^m + \dots + a_1 x_1 + a_0, (m = 2n - 1)$ ), 令上式对特殊函数准确成立。同样用待定系数法, 我们得到:

$$\int_{-1}^1 (a_m x_m^m + \dots + a_1 x_1 + a_0) dx = A_1 f(x_1) + A_2 f(x_2) + \dots + A_n f(x_n) \quad (5.45)$$

上式对任意的  $a_m, \dots, a_1, a_0$  都成立，令系数对应相等，我们得到：

$$\left\{ \begin{array}{l} A_1 + A_2 + \dots + A_n = \mu_0 \\ A_1 x_1 + A_2 x_2 + \dots + A_n x_n = \mu_1 \\ \vdots \\ A_1 x_1^m + A_2 x_2^m + \dots + A_n x_n^m = \mu_m \end{array} \right\} \text{2n 个方程, } \mu_k = \int_{-1}^1 x^k dx$$

(5.46)

解出  $A_1, A_2, \dots, A_n, x_1, x_2, \dots, x_n$ ，参数便确定下来。

n	$x_k$	$A_k$
2	$\pm 0.5773503$	1.0
4	$\pm 0.861136$	0.347
	$\pm 0.33998$	0.652
6	0.013	0.17
	$\pm 0.66$	0.36
	$\pm 0.238$	0.4677
$\vdots$	$\vdots$	$\vdots$

平时用的一般就  
龙贝格和高斯积分

表 5.2: 高斯积分法各项系数

How to do physics research?(实验)

1. 观测到新的现象
2. 实验途径
  - 研究对象（大多数情况）：体系不同，研究对象不同（10 100nm）
  - 观测手段（少数情况）：新的观测方法，新仪器设备
    - 低温制冷设备，温度降到 4.2K，超导
    - 扫描隧道电子显微镜，纳米现象

理论与试验：

1. 共同点：New
2. 相结合

5.5 反常积分

之前的内容中我们都是用多项式“拟合”被积函数，但是有如下缺点：

1. Runge 现象

## 2. 多项式不能拟合渐进行为（奇异行为）

但是在实际科研生活中我们总是要计算一些有奇异性质的积分，例如对于费米体系求其总能量：

$$\int_0^{\infty} f(\varepsilon) \varepsilon d\varepsilon = \int_0^{\Lambda} f(\varepsilon) \varepsilon d\varepsilon \quad (5.47)$$

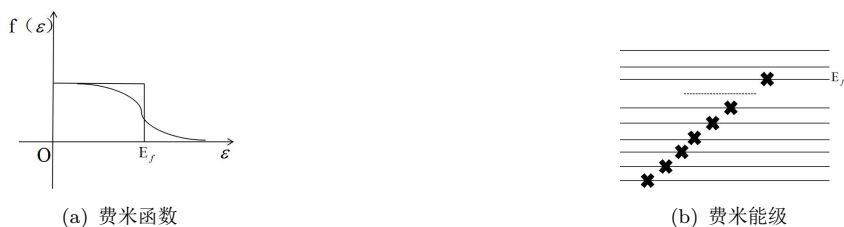


图 5.10: 费米系统

这种情况要对整个能量区间  $[0, +\infty]$  求积分，但一般情况下，我们是取一个很大的积分上限  $\Lambda$ ，实际数值计算中只需要取到  $E_f + 20k_B T$ ，手工计算甚至可以只取到  $E_f + 3k_B T$  ( $4k_B T$ )。这样的取法主要是因为费米系统中，温度不高时，只有费米面附近的粒子才有可能激发，但是多个费米子激发的概率很小所以可以只取到费米能以上的小范围内。

## 反常积分分类

1. 积分上下限、被积函数有限，存在极限值，

$$\int_0^{\frac{\pi}{2}} \frac{\sin x}{x} dx$$

2. 积分上下限为  $\pm\infty$ ，

$$\int_0^{\infty} \frac{e^{-x}}{1+x} dx$$

3. 积分区间两端含可去奇点，

$$\int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{x}} dx$$

4. 积分区间含可去奇点。

### 5.5.1 处理反常积分的基本方法

1. (变量) 变换
2. 对消
3. 分部积分

下面用各种例子说明上述方法。



## (1) 变量变换

(1)

$$\int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{x}} dx$$

奇异性主要来源于  $\frac{1}{\sqrt{x}}$ , 可以用变量代换的方式来消除这种奇异性, 令  $x = u^2$ , 则  $dx = 2u du$ , 代入得:

$$\int_0^{\sqrt{\frac{\pi}{2}}} \frac{1}{u} 2u du = \sqrt{2\pi}$$

上述结果可推广为:

$$\int_0^a \frac{f(x)}{x} dx \xrightarrow{x=u^2} \int_0^{\sqrt{a}} \frac{f(u^2)}{u} 2u du$$

其中  $f(x)$  是  $[0, a]$  ( $a > 0$ ) 上的解析函数。

(2)

$$\int_0^{\frac{\pi}{2}} \frac{\cos x}{\sqrt{x}} dx \xrightarrow{x=u^2} \int_0^{\sqrt{\frac{\pi}{2}}} 2 \cos u^2 du$$

(3)

$$\int_0^1 \frac{\sin x}{\sqrt{1-x}} dx \xrightarrow{1-x=u} \int_0^1 \frac{\sin(1-u)}{\sqrt{u}} du \xrightarrow{u=w^2} 2 \int_0^1 \sin(1-w^2) dw$$

上面都是一端有奇异性的情况, 对于两端都是可去奇点, 可以将区间分割。

(4)

$$\begin{aligned} & \int_0^1 \frac{\cos x}{\sqrt{x(1-x)}} dx \\ &= \int_0^{\frac{1}{2}} \frac{\cos x}{\sqrt{x(1-x)}} dx + \int_{\frac{1}{2}}^1 \frac{\cos x}{\sqrt{x(1-x)}} dx \end{aligned}$$

第一项令  $x = u^2$ , 第二项令  $1-x = u$ , 然后令  $u = w^2$  得到:

$$= \int_0^{\frac{1}{\sqrt{2}}} \frac{2 \cos u^2}{\sqrt{1-u^2}} du + \int_0^{\frac{1}{\sqrt{2}}} \frac{\cos(1-w^2)}{\sqrt{1-w^2}} dw$$

有时要将积分函数变化一下。

(5)

$$\begin{aligned} & \int_0^2 \frac{e^{-x}}{\sqrt{4-x^2}} dx = \int_0^2 \frac{e^{-x}}{\sqrt{(2-x)(2+x)}} dx \\ & \xrightarrow[\substack{2-x=u^2 \\ x=2-u^2}]{x=2-u^2} 2e^{-2} \int_0^{\sqrt{2}} \frac{e^{u^2}}{\sqrt{4-u^2}} du \end{aligned}$$

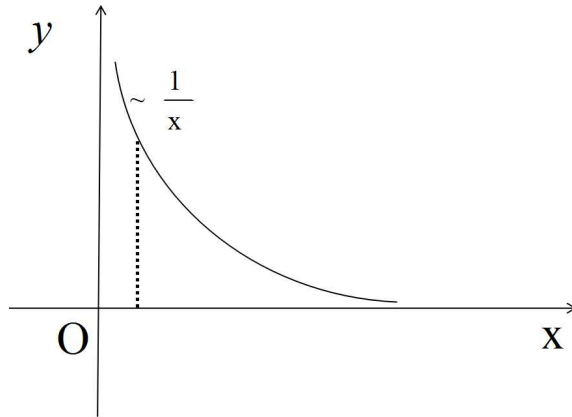
## (2) 利用对消, 消去奇点

(1)

$$\int_{\varepsilon}^1 \frac{e^{-x}}{x} dx$$

 $\varepsilon$  为很小的正数。

奇异性主要来源于  $\frac{1}{x}$ , 当  $x$  趋近  $\varepsilon$  时,  $\frac{e^{-x}}{x} \approx \frac{e^{-\varepsilon}}{x} \approx \frac{1}{x}$ , 我们知道  $\int \frac{1}{x} dx = \ln x + C$ 。

图 5.11:  $e^{-x}/x$  示意图

$$\begin{aligned} & \int_{\varepsilon}^1 \frac{e^{-x}}{x} dx \\ &= \int_{\varepsilon}^1 \left( \frac{e^{-x}}{x} - \frac{1}{x} \right) dx + \int_{\varepsilon}^1 \frac{1}{x} dx \\ &= \int_{\varepsilon}^1 \frac{e^{-x} - 1}{x} dx + \ln 1 - \ln \varepsilon \\ &= \int_{\varepsilon}^{\varepsilon'} \frac{e^{-x} - 1}{x} dx + \int_{\varepsilon'}^1 \frac{e^{-x} - 1}{x} dx - \ln \varepsilon \quad (\text{在小区间 } [\varepsilon, \varepsilon'] \text{ 上将 } e^{-x} \text{ 展开}) \\ &= \int_{\varepsilon}^{\varepsilon'} -\left(1 - \frac{x}{2!} + \frac{x^2}{3!} + \cdots\right) dx + \int_{\varepsilon'}^1 \frac{e^{-x}}{x} dx - \ln \varepsilon \end{aligned}$$

这样在区间  $[\varepsilon, \varepsilon']$  上积分可以用多项式求得, 而其他区间中被积函数没有奇异性, 可以直接用之前的数值积分方法直接求积分。

(2)

$$\int_0^{\frac{\pi}{2}} \frac{\sin x}{x^2} dx$$

其中  $x$  趋于  $\varepsilon$  时,  $\frac{\sin x}{x}$  有极限值, 奇异性主要来源于  $\frac{1}{x}$ , 同样用上述方法。

$$\begin{aligned} \int_0^{\frac{\pi}{2}} \frac{\sin x}{x^2} dx &= \int_0^{\frac{\pi}{2}} \frac{1}{x} dx - \int_0^{\frac{\pi}{2}} \left(1 - \frac{\sin x}{x}\right) \frac{1}{x} dx \\ &= \ln\left(\frac{\pi}{2\varepsilon}\right) - \int_{\varepsilon}^{\varepsilon'} x \left(\frac{1}{3!} - \frac{1}{5!} + \cdots\right) dx - \int_{\varepsilon'}^{\frac{\pi}{2}} \frac{1}{x} \left(1 - \frac{\sin x}{x}\right) dx \end{aligned}$$

(3) 分部积分 (笔记是积  $\ln x$ )

(1)

$$\int_0^1 x \ln x dx$$

这个积分直接直接用数值计算的方法不准确, 但是可以用分部积分的方法解析求解。

$$\int_0^1 x \ln x dx = \frac{1}{2} x^2 \ln x \Big|_0^1 - \frac{1}{2} \int_0^1 x dx \ln x = 0 - \frac{1}{2} \int_0^1 x \frac{1}{x} + x \ln x dx = -\frac{1}{3}$$

一般情况下简单去掉奇异性的方法不可取, 例如对于  $\int_a^b f(x) dx$ ,  $x=c$  是奇异点, ( $a < c < b$ ), 不能简单地令

$$\int_a^b f(x) dx = \int_a^{c-\varepsilon} f(x) dx + \int_{c+\varepsilon}^b f(x) dx$$

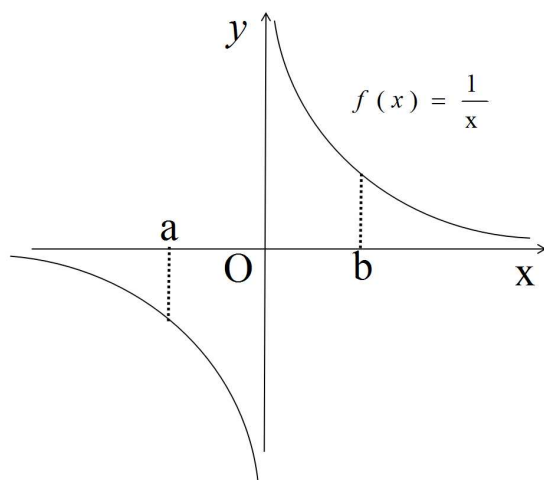


图 5.12: 简单去掉奇异性

例如对于  $f(x) = \frac{1}{x}$ , 如果  $a + b = 0$ , 反常积分就得零, 但是用上述方法就算不准了。

## 5.5.2 反常积分的几何行为

$$(1) \int \frac{1}{x} dx = \ln x + C$$

$$(2) \int \frac{dx}{x\sqrt{x^2 - a}} = \frac{1}{a} \cos^{-1} \left| \frac{a}{x} \right| + C$$

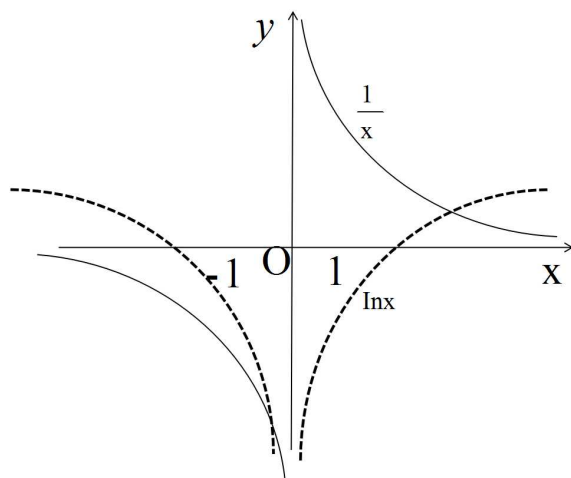


图 5.13:  $\int \frac{1}{x} dx = \ln x + C$

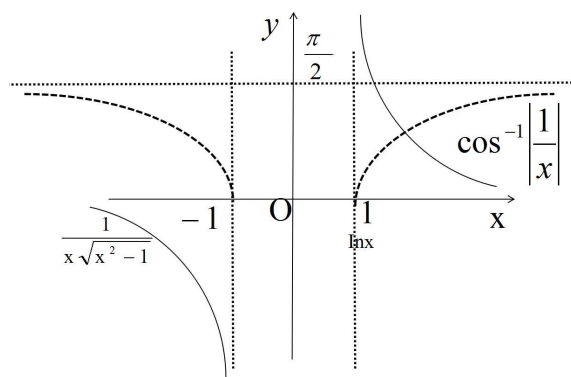


图 5.14:  $\int \frac{dx}{x\sqrt{x^2-1}} = \cos^{-1} \left| \frac{1}{x} \right| + C$

## 5.6 反常积分举例

### 5.6.1 含可去奇点的反常积分

(1)

$$\int_0^{\frac{\pi}{2}} \frac{\frac{\pi}{2} - x}{\sqrt{\sin x}} dx$$

奇异性来源于  $\frac{1}{\sqrt{\sin x}}$ , 可以凑出  $\frac{\sin x}{x}$  的形式。

$$\begin{aligned}
 \int_0^{\frac{\pi}{2}} \frac{\frac{\pi}{2} - x}{\sqrt{\sin x}} dx &= \int_0^{\frac{\pi}{2}} \frac{\frac{\pi}{2} - x}{\sqrt{\frac{\sin x}{x}} \sqrt{x}} dx \\
 \xrightarrow{x=u^2} \int_0^{\sqrt{\frac{\pi}{2}}} \frac{\frac{\pi}{2} - u^2}{\sqrt{\frac{\sin u^2}{u^2}}} du \\
 &= \int_0^\varepsilon \frac{\frac{\pi}{2} - u^2}{\sqrt{\frac{\sin u^2}{u^2}}} du + \int_\varepsilon^{\sqrt{\frac{\pi}{2}}} \frac{\frac{\pi}{2} - u^2}{\sqrt{\frac{\sin u^2}{u^2}}} du \\
 &= \int_0^\varepsilon \frac{\frac{\pi}{2} - u^2}{\sqrt{1 - \frac{(u^2)^2}{3!} + \frac{(u^2)^4}{5!} + \dots}} du + \int_\varepsilon^{\sqrt{\frac{\pi}{2}}} \frac{\frac{\pi}{2} - u^2}{\sqrt{\frac{\sin u^2}{u^2}}} du
 \end{aligned}$$

这样第一项用多项式拟合, 第二项可以直接算。

(2)

$$\int_0^1 \frac{dx}{\sqrt{\arctan x}}$$

我们知道  $x \rightarrow 0, \arctan x \rightarrow 0$ , 奇异性来源于  $\frac{1}{\arctan x}$ , 而且有  $\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} + \dots$ , 还是可以凑出  $\frac{\arctan x}{x}$  的形式。 3? ↑不知有无打错

$$\begin{aligned}
 \int_0^1 \frac{dx}{\sqrt{\arctan x}} &= \int_0^1 \frac{1}{\sqrt{\frac{\arctan x}{x}}} \frac{1}{x} dx \\
 \xrightarrow{x=u^2} \int_0^1 \frac{2}{\sqrt{\frac{\arctan u^2}{u^2}}} du \\
 &= \int_0^\varepsilon \frac{2}{\sqrt{1 - \frac{(u^2)^2}{3} + \frac{(u^2)^4}{5} + \dots}} du + \int_\varepsilon^1 \frac{2}{\sqrt{\frac{\arctan u^2}{u^2}}} du
 \end{aligned}$$

### 5.6.2 具有水平方向渐进行为的反常积分

(1)

$$\int_0^\infty \frac{e^{-x}}{x+1} dx$$

可以分成两段

$$\int_0^1 \frac{e^{-x}}{x+1} dx + \int_1^\infty \frac{e^{-x}}{x+1} dx$$

，第一项可以直接算，第二项可以化为如下形式：

$$\begin{aligned} \int_1^\infty \frac{e^{-x}}{x+1} dx &\xrightarrow[x=-\frac{1}{y^2} dy]{x=\frac{1}{y}} \int_0^1 \frac{e^{-\frac{1}{y}}}{1+y} \frac{dy}{y} \\ &= \int_0^\varepsilon \frac{e^{-\frac{1}{y}}}{1+y} \frac{dy}{y} + \int_\varepsilon^1 \frac{e^{-\frac{1}{y}}}{1+y} \frac{dy}{y} \\ &\quad \text{第一项趋于零, 有 } e^{-\frac{1}{y}} = 0 \cdot y + 0 \cdot y^2 + \cdots \\ &= \int_\varepsilon^1 \frac{e^{-\frac{1}{y}}}{1+y} \frac{dy}{y} \text{ (可以直接算)} \end{aligned}$$

(2)

$$\int_0^\infty \frac{\arctan x}{x\sqrt{x}} dx$$

可以看出  $x$  趋于 0 时，奇异性来源于  $\frac{1}{\sqrt{x}}$ ， $x$  趋于无穷时，被积函数具有水平渐进性。可以将积分区域分成两段：

$$\int_0^\infty \frac{\arctan x}{x\sqrt{x}} dx = \int_0^1 \frac{\arctan x}{x\sqrt{x}} dx + \int_1^\infty \frac{\arctan x}{x\sqrt{x}} dx$$

分别计算

$$\int_0^1 \frac{\arctan x}{x\sqrt{x}} dx \xrightarrow{x=u^2} 2 \int_0^1 \frac{\arctan u^2}{u^2} du = 2 \int_0^\varepsilon \left(1 - \frac{(u^2)^2}{3} + \frac{(u^2)^4}{5} + \cdots\right) du + 2 \int_\varepsilon^1 \frac{\arctan u^2}{u^2} du$$

$$\begin{aligned} \int_1^\infty \frac{\arctan x}{x\sqrt{x}} dx &\xrightarrow[x=\frac{1}{y}]{x=\frac{1}{y}} \int_0^1 \frac{\arctan \frac{1}{y}}{\sqrt{y}} dy \\ &= \int_0^1 \frac{\pi}{2\sqrt{y}} dy + \int_0^1 \frac{\arctan y}{\sqrt{y}} dy \quad (\arctan \frac{1}{y} = \frac{\pi}{2} - \arctan y) \\ &\xrightarrow{y=u^2} \int_0^1 \pi du - \int_0^1 2 \arctan u^2 du = \pi - \int_0^1 2 \arctan u^2 du \end{aligned}$$

### 5.6.3 含有 $\ln x$ ( $x \rightarrow 0$ 或 $x \rightarrow$ 很大) 的反常积分

我们知道  $x \rightarrow 0, \ln x \rightarrow -\infty$ 。之前的例子：

$$\int_0^1 \ln x dx = x \ln x|_0^1 - \int_0^1 x d \ln x = -1$$

## 期中考试内容

1. 香农熵  $H(P) = -P \ln P - (1 - P) \ln(1 - P)$  在  $[0, \varepsilon]$  和  $[1 - \varepsilon, 1]$  该如何计算。

泰勒展开以  $[0, \varepsilon]$  区间上  $P \ln P$  的展开为例，主要是对  $\ln P$  展开：

$$\ln P = \ln(\varepsilon + P - \varepsilon) = \ln \varepsilon + \ln\left(1 + \frac{P - \varepsilon}{\varepsilon}\right) \approx \ln \varepsilon + \frac{P - \varepsilon}{\varepsilon} + \frac{(P - \varepsilon)^2}{2\varepsilon^2} \quad (5.48)$$

注意验证  $P = 0$  时， $P \ln P = 0$ 。

2. 已知  $ay^2 - 2y + 2 = 0$  两种方法绘制  $y$  随  $a$  的变化曲线。

(1) 解析法直接求： $a = 0, y = 1, a \neq 0$ ，有两支。(2) 求  $a(y)$ ，画出图像后作出反函数图像。

How to give a talk?

1. 准备，一般就是做好 PPT
2. 内容，包括每部分的时间分配
3. 表达：
  - 抑扬顿挫（多读诗词？）
  - 长时间演讲能力

## 题外话：香农熵（信息学）

如果事件本身发生的概率为  $P$ ，引入自信息： $I = -\log P$ ，可以看出几率大时信息量小，几率小时信息量大。

再设随机变量  $X$  的几率分布为  $P(x)$ ，则  $I(x)$  的平均为（看成离散）：

$$H(x) = \sum_x I(x)P(x) \quad (5.49)$$

可以看成信息量的均值。可以类别热力学中的熵：

$$H = \sum_S P_S \ln P_S \quad (5.50)$$

$S$  为微观状态数。

回到香农熵，如果只有两种状态（抛硬币），则其变为

$$H = -P \ln P - (1 - P) \ln(1 - P) \quad (5.51)$$

以上只是想说实际中可能会出现  $x \ln x, x \rightarrow 0$  的形式，下面就来讨论在积分中的这种情况。

我们知道  $x \rightarrow 0, \ln x \rightarrow -\infty$ ，而由洛必达法则可知  $x \ln x|_0 = 0$ ，我们可以理解为  $x \rightarrow 0$  比  $\ln x \rightarrow -\infty$  要快，同样可以推广到  $x^n \ln x, n = 1, 2, 3, \dots$ ，可以画出前几个图像如下所示：

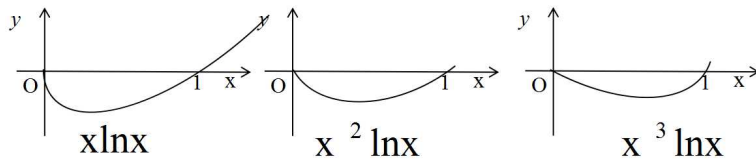


图 5.15:  $x^n \ln x, n = 1, 2, 3$  示意图

可以看出  $n$  越大，函数在  $[0, 1]$  区间围成的面积越小。下面我们就用这种性质计算带有  $\ln x, x \rightarrow 0$  的积分。

(1)

$$\int_0^{1.2} \frac{\ln x}{e^{-x} + x} dx$$

做变换  $x = w^4 (w^2, w^3 \text{ 也可以, 但是最好不要低于两次})$

$$\text{原式} = \int_0^{1.2^{1/4}} \frac{16w^3 \ln w}{(e^{-x} + x)|_{x=w^4}} dw = \int_0^\varepsilon \frac{16w^3 \ln w}{(e^{-x} + x)|_{x=w^4}} dw + \int_\varepsilon^{1.2^{1/4}} \frac{16w^3 \ln w}{(e^{-x} + x)|_{x=w^4}} dw \quad (5.52)$$

第二项可以直接算，第一项有如下几种情况：

1. 简单情况下，可以认为第一项为 0。



2.  $w^3$  不变, 在  $w = \varepsilon$  或者  $w = \frac{\varepsilon}{A}$  展开  $\ln w$  得

$$I = \int_0^\varepsilon \frac{w^3 \left[ \ln \varepsilon + \frac{1}{\varepsilon}(w - \varepsilon) + \cdots \right]}{(e^{-x} + x)|_{x=w^4}} dw \quad (5.53)$$

3. 认为分母在这一小区间近似不变, 把分母提出来:

$$I = \frac{1}{(e^{-x} + x)|_{x=w^4}} \int_0^\varepsilon w^3 \ln w dw \quad (5.54)$$

分母中  $w = 0$  或  $\varepsilon$ , 后面的积分可以用分部积分的方法算出。

$$\begin{aligned} \int_0^\varepsilon w^3 \ln w dw &= w^4 \ln w \Big|_0^\varepsilon - \int_0^\varepsilon w dw^3 \ln w = \varepsilon^4 \ln \varepsilon - \int_0^\varepsilon 3w^3 \ln w dw - \int_0^\varepsilon \frac{w^4}{w} dw \\ 4 \int_0^\varepsilon w^3 \ln w dw &= \varepsilon^4 \ln \varepsilon - \frac{\varepsilon}{4} \\ \int_0^\varepsilon w^3 \ln w dw &= \frac{1}{4} \varepsilon^4 \ln \varepsilon - \frac{\varepsilon}{16} \end{aligned} \quad (5.55)$$

最终得到:

$$I = \frac{1}{(e^{-x} + x)|_{x=w^4}} \left( \frac{1}{4} \varepsilon^4 \ln \varepsilon - \frac{\varepsilon}{16} \right) \quad (5.56)$$

(2)

$$\int_0^G \frac{\ln x}{1+x} dx$$

$G$  很大。该积分在 0 和  $G$  附近情况不同, 分两段处理:

$$\text{原积分} = \int_0^1 \frac{\ln x}{1+x} dx + \int_1^G \frac{\ln x}{1+x} dx \quad (5.57)$$

第一项同样做变换  $x = w^4$ , 第二项做变换  $x = \frac{1}{y}$ :

$$\text{原积分} = \int_0^1 \frac{16w^3 \ln w}{1+w^4} dw + \int_{1/G}^1 \frac{-\ln y}{(1+y)y} dy \quad (5.58)$$

第一项的算法前面已经讨论过了, 下面讨论第二项, 第二项的奇异性来源于  $\frac{\ln y}{y}, y \rightarrow \frac{1}{G}$ , 我们用之前提到的对消的方式, 消去奇异性:

$$\begin{aligned} \int_{1/G}^1 \frac{-\ln y}{(1+y)y} dy &= \int_{1/G}^1 \frac{-\ln y}{y} dy + \int_{1/G}^1 \frac{\ln y}{y} \frac{\ln y}{(1+y)y} dy \\ &= \left[ -\frac{1}{2} (\ln y)^2 \right] \Big|_{1/G}^1 + \int_{1/G}^1 \frac{\ln y}{1+y} dy \\ &= \frac{1}{2} (\ln y)^2 + \int_1^{(1/G)^{\frac{1}{4}}} \frac{16w^3 \ln w}{1+w^4} dw \end{aligned} \quad (5.59)$$

上式最后一步中令  $y = w^4$ ，得到的结果同样是之前讨论过的。

#### 5.6.4 含有动态奇异性的反常积分

(1)

$$\int_0^1 \frac{e^{-x}}{\sqrt{|x^2 - b^2|}} dx, (0 < b < 1)$$

这一积分的奇异性在  $\frac{e^{-x}}{\sqrt{|b^2 - x^2|}}, x \rightarrow b$ ，为方便讨论我们将积分区间分成两部分：

$$\text{原积分} = \int_0^b \frac{e^{-x}}{\sqrt{b^2 - x^2}} dx + \int_b^1 \frac{e^{-x}}{\sqrt{x^2 - b^2}} dx \quad (5.60)$$

对第一项我们可以采取变量变换（换元）的方法，令  $x = b \sin \theta$ ，第二项可以采取对消的方式：

$$\text{原积分} = \int_0^{\frac{\pi}{2}} e^{-b \sin \theta} d\theta + e^{-b} \int_b^1 \frac{dx}{\sqrt{x^2 - b^2}} - \int_b^1 \frac{e^{-b} - e^{-x}}{\sqrt{x^2 - b^2}} dx \quad (5.61)$$

第一项可以直接用数值积分的方法计算，第二项可以直接解析求解，第三项仍可以用换元的方法，令  $x - b = v^2 > 0, dx = 2v dv$

$$\text{原积分} = \int_0^{\frac{\pi}{2}} e^{-b \sin \theta} d\theta + e^{-b} [\ln(1 + \sqrt{1 - b^2}) - \ln b] + 2e^{-b} \int_0^{\sqrt{1-b^2}} \frac{1 - e^{-v^2}}{\sqrt{2b + v^2}} dv \quad (5.62)$$

可以看出最后一项的奇异性也已经消除了。

上式利用了积分公式  $\int \frac{1}{\sqrt{x^2 \pm b^2}} dx = \ln(x + \sqrt{x^2 \pm a^2}) + C$ 。

How to write article in English?

1. 模仿（网上查，不是 copy）
2. 地道英文
3. 想法

# 第6章 快速傅里叶变换算法

傅氏变换分为两类（两种方法）：

- 解析方法，傅氏积分，傅氏函数
- 代数方法，有限离散傅氏变换

下面主要介绍有限离散傅氏变换。

## 6.1 有限离散傅氏变换

已知函数  $f(x)$  在  $0 \leq x \leq 2\pi$  区间  $N$  个等分点（ $N$  个点也行这里方便讨论） $x = \frac{2\pi l}{N}, l = 0, 1, 2, \dots, N-1$  的函数值为  $f_l$ ，即  $f_l = f(\frac{2\pi l}{N})$ ，则这  $N$  个点的傅氏变换为：

$$c_k = \frac{1}{N} \sum_{l=0}^{N-1} f_l e^{-ik \frac{2\pi l}{N}}, \quad k = 0, 1, \dots, N-1 \quad (6.1)$$

$$f_l = \sum_{k=0}^{N-1} c_k e^{ik \frac{2\pi l}{N}}, \quad l = 0, 1, \dots, N-1 \quad (6.2)$$

可以看出傅氏变换反映一种集体效应。

回想固体物理中倒空间与正空间之间的变化（李正中——固体理论），还有格波的集体效应  
.....

推导：

由于我们定义的函数是在  $[0, 2\pi]$  区间，可以延拓成  $2\pi$  为周期的函数，我们选取周期基函数  $\{e^{ijx}\}, j = 0, 1, \dots, N-1, e^{ijx} = \cos(jx) + i \sin(jx), i = \sqrt{-1}$ 。

做插值函数来拟合：

$$P(x) = \sum_{j=0}^{N-1} c_j e^{ijx} \quad (6.3)$$

把我们取的特殊点  $\frac{2\pi l}{N}$  代入得：

$$p(\frac{2\pi l}{N}) = f(\frac{2\pi l}{N}) = \sum_{j=0}^{N-1} c_j e^{ij \frac{2\pi l}{N}} \quad (6.4)$$

即:  $f_l = \sum_{j=0}^{N-1} c_j e^{ij \frac{2\pi l}{N}}$ , 可以类比量子力学中动量空间与位置空间的变化。

为了计算  $c_j$ , 可以将上式乘以  $e^{-ik \frac{2\pi l}{N}}$ , 再对  $k$  求和 (量子力学中为了提出某一项系数, 左乘一个左矢):

$$\begin{aligned} \sum_{l=0}^{N-1} f_l e^{-ik \frac{2\pi l}{N}} &= \sum_{l=0}^{N-1} \sum_{j=0}^{N-1} c_j e^{i(j-k) \frac{2\pi l}{N}} \\ &= \sum_{j=0}^{N-1} c_j \sum_{l=0}^{N-1} e^{i(j-k) \frac{2\pi l}{N}} \\ &= \sum_{j=0}^{N-1} c_j N \delta_{jk} = N c_k \end{aligned} \quad (6.5)$$

最终我们得到:

$$c_k = \frac{1}{N} \sum_{l=0}^{N-1} f_l e^{-ik \frac{2\pi l}{N}} \quad (6.6)$$

上式求和的计算:

- $j = k$  时,

$$\sum_{l=0}^{N-1} e^{i(j-k) \frac{2\pi l}{N}} = \sum_{l=0}^{N-1} 1 = N \quad (6.7)$$

- $j \neq k$ , 时是一个等比数列求和:

$$\sum_{l=0}^{N-1} e^{i(j-k) \frac{2\pi l}{N}} = \frac{1 - e^{i(j-k) 2\pi}}{1 - e^{i(j-k) \frac{2\pi}{N}}} = 0 \quad (6.8)$$

最终我们得到:  $\sum_{l=0}^{N-1} e^{i(j-k) \frac{2\pi l}{N}} = N \delta_{jk}$

几种看法:

1. 傅氏变换体现集体效应
2. 泛函,  $c_k$  的形式与数值是和  $f(x)$  有关的。

## 6.2 快速傅氏变换

现在我们知道了如何进行傅氏变换, 分析一下计算量: 每一个  $c_k$  要计算  $N$  个指数  $e^{-ik \frac{2\pi l}{N}}$ ,  $l = 0, 1, \dots, N-1$ , 再求和, 一共有  $N$  个  $k$ , 所以要做  $N^2$  次运算, 计算量很大, 现在我们介绍一种快速傅里叶变换的算法。

我们看出每次计算的指数  $e^{-ik\frac{2\pi l}{N}}$  不是独立（即会有重复的结果），真正独立的是

$$\{1, e^{-ik\frac{2\pi l}{N}}, \dots, e^{-ik\frac{2\pi(N-1)}{N}}\}$$

，我们可以找到一种方法来简化这种重复的运算，具体算法：取  $N = 2^m, m = 1, 2, \dots$ ，令  $w = e^{-i\frac{2\pi}{N}}$ ，可以看出  $w^N = 1$ ，重新定义  $f_l = \frac{1}{N}f\left(\frac{2\pi l}{N}\right)$ ， $c_k = \sum_{l=0}^{N-1} f_l w^{kl}$ ， $k = 0, 1, \dots, N-1$ ，可以把  $k$  为奇偶数分开讨论：

1.  $k$  为偶数， $k = 2j_1, j_1 = 0, 1, \dots, N/2 - 1$

$$\begin{aligned} c_{2j_1} &= \sum_{l=0}^{N-1} f_l w^{2j_1 l} = \sum_{l=0}^{N/2-1} f_l w^{2j_1 l} + \sum_{l=N/2}^{N-1} f_l w^{2j_1 l} \\ &\quad \text{第二项令 } l' = l - \frac{N}{2}, \text{ 再令 } l = l' \\ &= \sum_{l=0}^{N/2-1} f_l w^{2j_1 l} + \sum_{l'=0}^{N/2-1} f_{l+N/2} w^{2j_1(l+N/2)} \\ &= \sum_{l=0}^{N/2-1} (f_l + f_{l+N/2}) w^{2j_1 l} \end{aligned} \tag{6.9}$$

可以看成对每一个  $c_{2j_1}$  要计算  $\frac{N}{2}$  个指数， $c_{2j_1}(c_0, \dots, c_{N-2})$  一共有  $\frac{N}{2}$  个，运算量为  $\frac{N}{2} \times \frac{N}{2} = \frac{N^2}{4}$ 。

2.  $k$  为奇数， $k = 2j_1 + 1, j_1 = 0, 1, \dots, N/2 - 1$

$$\begin{aligned} c_{2j_1+1} &= \sum_{l=0}^{N-1} f_l w^{(2j_1+1)l} = \sum_{l=0}^{N/2-1} f_l w^{(2j_1+1)l} + \sum_{l=N/2}^{N-1} f_l w^{(2j_1+1)l} \\ &\quad \text{第二项令 } l' = l - \frac{N}{2}, \text{ 再令 } l = l' \\ &= \sum_{l=0}^{N/2-1} f_l w^{(2j_1+1)l} + \sum_{l'=0}^{N/2-1} f_{l+N/2} w^{(2j_1+1)(l+N/2)} \\ &\quad \text{(注意到 } w^{(2j_1+1)N/2} = e^{-i\frac{2\pi}{N}(2j_1+1)\frac{N}{2}} = -1) \\ &= \sum_{l=0}^{N/2-1} (f_l - f_{l+N/2}) w^{2j_1 l} \end{aligned} \tag{6.10}$$

同样可以看出奇数项  $c_{2j_1+1}$  每项计算  $\frac{N}{2}$  个指数，共有  $\frac{N}{2}$  个奇数项，计算量和偶数项一样都为  $\frac{N^2}{4}$ ，这样总的计算量为  $\frac{N^2}{2}$ ，相对原来减少了一半，进一步简化还可以把  $j_1$  分奇数偶数 ( $j_1 = 2j_2, j_1 = 2j_2 + 1$ ) 讨论。



# 第7章 线性代数方程组的解法

这一章主要讨论线性方程组的解法，一个  $n$  阶的线性方程组如下所示：

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases} \quad (7.1)$$

当方程组线性相关时不好解。将介绍以下几种数值方法：

1. 消去法（直接法）
2. 迭代法
3. LU 分解法
4. 病态线性方程组的解法

## 7.1 直接法（高斯消去法）

### 7.1.1 高斯消去法

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases} \Rightarrow \begin{cases} b_{11}x_1 + b_{12}x_2 + \cdots + a_{1n}x_n = g_1 \\ b_{22}x_2 + \cdots + b_{2n}x_n = g_2 \\ \vdots \\ b_{nn}x_n = g_n \end{cases} \quad (7.2)$$

注意：

- $a_{kk} \neq 0$ ，而且  $a_{kk}$  要尽量大；
- 主元素消去，取  $a_{kk}$  最大，即通过变换把每一列最大的系数移到  $a_{kk}$  的位置。

## 7.2 三对角线型和一般带型线性代数方程组

### 7.2.1 三对角型

$$AX = D$$

$$A = \begin{bmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ & a_{32} & a_{33} & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & a_{n-1n-1} & a_{n-1n} \\ & & & & a_{nn-1} & a_{nn} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{bmatrix} \quad D = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \vdots \\ d_n \end{bmatrix} \quad (7.3)$$

$n \times n$  矩阵在电脑里存储，需要内存， $n$  越大，需要的内存容量也要大，解决方案：

1. 特殊型——压缩存储技术

2. 扩大内存容量

- Cluster
- 超算，高性能计算机机群

目前的计算水平很高，通过云服务器，我们可以进行很大规模的运算。老师推荐阿里云。



### 7.2.2 一般带型

$AX = D$ , 一般要压缩降维。

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1l+1} & & \\ \vdots & \ddots & \vdots & \ddots & \\ a_{l+11} & \cdots & a_{l+1l+1} & \cdots & a_{l+12l+1} \\ & \ddots & & \ddots & \\ & & a_{n-ln-2l} & \cdots & a_{n-ln-l} & \cdots & a_{n-ln} \\ & & & \vdots & \ddots & \vdots & \\ & & & & a_{nn-l} & \cdots & a_{nn} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & \cdots & a_{1l+1} & & \\ \vdots & \ddots & \vdots & \ddots & \\ a_{l+11} & \cdots & a_{l+1l+1} & \cdots & a_{l+12l+1} \\ \vdots & & \vdots & & \\ a_{n-ln-2l} & \cdots & a_{n-ln-l} & \cdots & a_{n-ln} \\ \vdots & / & \vdots & / & \\ a_{nn-l} & \cdots & a_{nn} & & 0 \end{bmatrix} \quad (7.4)$$

## 7.3 迭代法

### 7.3.1 简单迭代法

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases} \Rightarrow \begin{cases} x_1 = b_{11}x_1 + b_{12}x_2 + \cdots + a_{1n}x_n + g_1 \\ x_2 = b_{12}x_1 + b_{22}x_2 + \cdots + b_{2n}x_n + g_2 \\ \vdots \\ x_n = b_{1n}x_1 + b_{2n}x_2 + \cdots + b_{nn}x_n + g_n \end{cases} \quad (7.5)$$

迭代关系  $X = BX + g$ , 取初始  $X^{(0)}$

$$\begin{aligned} x^{(1)} &= BX^{(0)} + g \\ &\vdots \\ x^{(k+1)} &= BX^{(k)} + g \end{aligned} \quad (7.6)$$

当  $|x_i^{(k+1)} - x_i^{(k)}| < \varepsilon$ , 对每一个  $i$  都成立时, 解为:

$$x^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} \quad (7.7)$$

也存在收敛条件  $\max_{1 \leq i \leq n} \sum_j |b_{ij}| < 1$ 。

### 7.3.2 Seidal 迭代形式

$$x_i^{(k+1)} = \sum_{j=1}^{i-1} b_{ij} x_j^{(k+1)} + \sum_{j=i}^N b_{ij} x_j^{(k)} + g_i \quad (7.8)$$

这种方式就是及时校正。

### 7.3.3 松弛法

引入系数  $w$ :

$$x_i^{(k+1)} = x_i^{(k)} + w(\bar{x}_i^{(k+1)} - x_i^{(k)}) \quad (7.9)$$

破坏了迭代关系, 但是加快收敛, 最终还是要用原始的迭代关系  $x_i^{(k+1)} = \sum_{j=1}^{i-1} b_{ij} x_j^{(k+1)} + \sum_{j=i}^N b_{ij} x_j^{(k)} + g_i$  检验解的准确性。

How to think critically?

1. 多个不同角度考虑问题: New ideas, make decision。
2. 基石——知道很多, 系统地知道很多, 知识的宽度和深度。
3. 下笨功夫。

## 7.4 LU 分解法

特点:

只对系数矩阵做变换。

对于线性方程组  $AX = b$ 。

### 7.4.1 LU 分解法

主要做法：系数矩阵  $A$  分成单位下三角矩阵（对角元为 1）和上三角矩阵，条件是  $a_{ij} \neq 0$ 。

$$A = LU \quad (7.10)$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ L_{21} & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ L_{n1} & \cdots & \cdots & 1 \end{bmatrix} \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \quad (7.11)$$

这样线性方程组可写为如下形式：

$$Ax = b \Rightarrow LUX = b \quad (7.12)$$

解方程的步骤变为：

1. 通过  $LY = b$  作出  $Y$ ；
2.  $UX = Y$ （代入  $Y$  的解）作出  $X$ 。

### 7.4.2 矩阵的三角分解

$L$ 、 $U$  矩阵代入  $A = LU$ ，令矩阵元对应相等。（可以通过矩阵的初等变换进行  $LU$  分解，初等变换把  $A$  变为上三角矩阵， $L$  就可以写为这些初等变换矩阵逆的乘积）

### 7.4.3 矩阵求逆

设  $A$  为  $n \times n$  方阵，其逆矩阵为  $X$ ， $AX = I$ ， $X$  仍为  $n \times n$  的方阵， $I$  为单位矩阵。设：

$$X = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(i)} & \cdots & x_1^{(n)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_i^{(1)} & \cdots & x_i^{(i)} & \cdots & x_i^{(n)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_n^{(1)} & \cdots & x_n^{(i)} & \cdots & x_n^{(n)} \end{bmatrix} \quad (7.13)$$

令  $e_i$  为单位矩阵的第  $i$  列 (即列向量),  $x^{(i)}$  为矩阵  $X$  的第  $i$  列, 我们可以得到  $AX^{(i)} = e_i$ , 即:

$$\begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{ni} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(i)} \\ \vdots \\ x_i^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{第 } i \text{ 行} \quad (7.14)$$

解这一系列系数矩阵相同的方程组, 用 LU 分解法容易处理。

## 7.5 “病态”线性代数方程组

### 7.5.1 “病态”线性代数方程组

定性: 系数矩阵  $A$  或  $b$  的矩阵元微小的变化, 引起解有巨大变化。

同样考虑  $AX = b$ , 对于特定情况:

$$A = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \tilde{b} = \begin{bmatrix} b_1 + \varepsilon \\ b_2 + \varepsilon \\ b_3 + \varepsilon \end{bmatrix} \quad X - \tilde{X} = \begin{bmatrix} -492\varepsilon \\ 1860\varepsilon \\ -1500\varepsilon \end{bmatrix} \quad (7.15)$$

### 7.5.2 实际计算出现“病态”情况的原因

1. 主元素出现小的主元 (对角元);
2. 系数矩阵行 (列) 近似线性相关;
3. 出现很大的解。

### 7.5.3 “病态”线性方程组求解

步骤:

1. 用全主元素高斯消去法，得到一组解：

$$X^{(1)} = \begin{bmatrix} x_1^{(1)} \\ \vdots \\ x_n^{(1)} \end{bmatrix} \quad (7.16)$$

2. 计算剩余向量：

$$R = B - AX^{(1)} = AE \quad (7.17)$$

差异方程：  $A(X^{(1)} + E) = B$

3. 再用全元素高斯消去法解：

$$AE = R, \text{ 得出 } E = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} \quad (7.18)$$

4. 计算  $X^{(2)} = X^{(1)} + E$ ;

5. 令  $X^{(1)} = X^{(2)}$  转到步骤 2 重复解这个方程，直到  $\left| x_i^{(2)} - x_i^{(1)} \right| < \varepsilon$  for every i。



## 第 8 章 矩阵的本征值和本征矢的计算

回想量子力学中的不含时薛定谔方程：

$$\hat{H} |\psi\rangle = E |\psi\rangle \quad (8.1)$$

设其本征基矢为  $\{|\varphi_i\rangle\}$ ，可能是无穷维（就是有很多），这样波函数可以写为其线性组合： $|\psi\rangle = \sum_i C_i |\varphi_i\rangle$ ，薛定谔方程可写为：

$$\hat{H} \sum_i |\varphi_i\rangle = E \sum_i |\varphi_i\rangle \quad (8.2)$$

左乘  $\langle\varphi_j|$ （计算系数常用方式）得到：

$$\sum_i \langle\varphi_j| \hat{H} |\varphi_i\rangle = E C_j \rightarrow \sum_i H_{ji} C_i = E C_j \quad (8.3)$$

假设共有  $n$  个本征矢（截断近似） $z$  左乘  $n$  次可以得到：

$$\begin{bmatrix} H_{11} & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{n1} & \cdots & H_{nn} \end{bmatrix} \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} = E \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} \quad (8.4)$$

解久期方程：

$$\begin{vmatrix} H_{11} - E & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{n1} & \cdots & H_{nn} - E \end{vmatrix} = 0 \quad (8.5)$$

可以得到本征值，再把本征值带回到本征方程中可以解出相应的本征矢，但这种只  $n = 2$  适用， $n > 2$  不适用。

结果可以写为:

$$\begin{bmatrix} H_{11} & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{n1} & \cdots & H_{nn} \end{bmatrix} \begin{bmatrix} \cdots & C_1^{(i)} & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & C_n^{(i)} & \cdots \end{bmatrix} = \begin{bmatrix} \cdots & C_1^{(i)} & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & C_n^{(i)} & \cdots \end{bmatrix} \begin{bmatrix} E_1 & 0 & 0 \\ 0 & \vdots & 0 \\ 0 & 0 & E_n \end{bmatrix} \quad (8.6)$$

可以写为:

$$HC = CE \Rightarrow C^{-1}HC = E \quad (8.7)$$

这给出了求本征值本征矢的一般步骤, 通常可以选择单位正交矩阵来进行变换即  $U^{-1}HU$ , 后面会看到。

## 8.1 雅可比方法

我们先讨论实对称矩阵, 然后再推广 (复的如厄米矩阵)。

### 8.1.1 实对称矩阵的性质

1. 本征值为实;
2. 实对称矩阵 A 用正交矩阵对角化:

$$U^T A U = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \quad (8.8)$$

$U^{-1} = U^T$  为单位正交矩阵。

3. 实对称矩阵 A, 正交矩阵 R, 则  $B = R^T A R$  仍为实对称。

$$B^T = (R^T A R)^T = R^T A^T R = R^T A R = B \quad (8.9)$$



## 8.1.2 二阶对称矩阵对角化

设实对称矩阵  $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ ,  $a_{12} = a_{21}$ , 二阶正交矩阵为  $U = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}$ , 可

以算出:

$$U^{-1} = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} = U^T \quad (8.10)$$

$\varphi$  为待定系数。

作正交相似变换我们有:

$$\begin{aligned} U^{-1}AU &= \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} \\ &= \begin{bmatrix} ca_{11} + sa_{21} & ca_{12} + sa_{22} \\ -sa_{11} + ca_{21} & -sa_{12} + ca_{22} \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \\ &= \begin{bmatrix} c^2a_{11} + 2csa_{12} + s^2a_{22} & cs(-a_{11} + a_{22}) + (c^2 - s^2)a_{12} \\ cs(-a_{11} + a_{22}) + (c^2 - s^2)a_{12} & s^2a_{11} - 2csa_{12} + c^2a_{22} \end{bmatrix} \end{aligned} \quad (8.11)$$

其中  $\cos \varphi = c, \sin \varphi = s$ 。强行令非对角元为零, 可以得到:

$$cs(-a_{11} + a_{22}) + (c^2 - s^2)a_{12} = 0 \Rightarrow \frac{1}{2} \sin 2\varphi(-a_{11} + a_{22}) + \cos 2\varphi a_{12} = 0 \quad (8.12)$$

解得:

$$\tan 2\varphi = \frac{2a_{12}}{a_{11} - a_{22}} \Rightarrow \varphi = \frac{1}{2} \arctan \frac{2a_{12}}{a_{11} - a_{22}} \quad (8.13)$$

这里当  $a_{11} = a_{22}$  时, 出现一种伪奇异性, 可以选择:

$$\cot 2\varphi = \frac{a_{11} - a_{22}}{2a_{12}} \Rightarrow \varphi = \frac{1}{2} \operatorname{arccot} \frac{a_{11} - a_{22}}{2a_{12}} \quad (8.14)$$

把  $\varphi$  代入我们可以得到:

$$U^{-1}AU = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (8.15)$$

所以:

1. 本征值  $\lambda_1$ :

$$\lambda_1 = (c^2 a_{11} + 2csa_{12} + s^2 a_{22}) \Big|_{\varphi = \frac{1}{2} \arctan \frac{2a_{12}}{a_{11} - a_{22}}} \quad (8.16)$$

相应本征矢为  $\begin{bmatrix} \cos \varphi \\ \sin \varphi \end{bmatrix}$ 。

2. 本征值  $\lambda_2$ :

$$\lambda_2 = (s^2 a_{11} - 2csa_{12} + c^2 a_{22}) \Big|_{\varphi = \frac{1}{2} \arctan \frac{2a_{12}}{a_{11} - a_{22}}} \quad (8.17)$$

相应本征矢为  $\begin{bmatrix} -\sin \varphi \\ \cos \varphi \end{bmatrix}$ 。

### 8.1.3 雅可比方法

现在将二阶情况推广到任意阶情况。同样我们选择正交相似变换，只是需要一系列的相似正交变换最终实现相似对角化。

考虑  $n$  维实对称矩阵  $A$ ,

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, A^T = A$$

取  $n$  维正交矩阵  $R(p, q, \theta)$

$$R(p, q, \theta) = \begin{matrix} & \begin{matrix} \text{p 列} & & \text{q 列} \end{matrix} & \\ \begin{matrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cos \theta & 0 & \cdots & 0 & -\sin \theta & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sin \theta & 0 & \cdots & 0 & \cos \theta & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{matrix} & \begin{matrix} \text{p 行} \\ \\ \\ \\ \\ \\ \text{q 行} \\ \\ \\ \end{matrix} \end{matrix} \quad (8.18)$$

上式正交矩阵进行正交变换的同时，消去了  $a_{pq}, a_{qp}$ ，令  $B = R^T(p, q, \theta)AR(p, q, \theta)$ ，有以下结论：

1. B 仍为实对称矩阵；

2. B 与 A 比较只有第 p 行第 q 列，第 p 列第 q 行的元素发生变化，其它元素不变化，有如下关系式：

交叉点处：

$$\begin{cases} b_{pp} = a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta + a_{pq} \sin 2\theta \\ b_{qq} = a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta - a_{pq} \sin 2\theta \\ b_{pq} = \frac{1}{2}(a_{qq} - a_{pp}) \sin 2\theta + a_{pq} \cos 2\theta \\ b_{qp} = b_{pq} \end{cases} \quad (8.19)$$

非交叉点处:

$$\begin{cases} b_{pj} = a_{pj} \cos \theta + a_{qj} \sin \theta \\ b_{qj} = -a_{pj} \sin \theta + a_{qj} \cos \theta \\ b_{ip} = a_{ip} \cos \theta + a_{iq} \sin \theta \\ b_{iq} = -a_{ip} \sin \theta + a_{iq} \cos \theta \\ b_{ij} = a_{ij} \end{cases} \quad (8.20)$$

其中  $i, j = 1, 2, \dots, n$ , 但  $i, j \neq p, q$ .

令  $b_{pq} = b_{qp} = 0$ :

$$b_{pq} = \frac{1}{2}(a_{qq} - a_{pp}) \sin 2\theta + a_{pq} \cos 2\theta = 0 \Rightarrow \theta = \frac{1}{2} \operatorname{arccot} \frac{a_{pp} - a_{qq}}{2a_{pq}} \quad (8.21)$$

非对角元跑到哪了?

由前面的式中可以推出:

$$\begin{cases} b_{ip}^2 + b_{iq}^2 = a_{ip}^2 + a_{iq}^2 \\ b_{pj}^2 + b_{qj}^2 = a_{pj}^2 + a_{qj}^2 \\ b_{pp}^2 + b_{qq}^2 = a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 \end{cases} \quad (8.22)$$

第三个式子:

$$\begin{aligned} b_{pp}^2 + b_{qq}^2 &= (a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta + a_{pq} \sin 2\theta)^2 + (a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta - a_{pq} \sin 2\theta)^2 \\ &= (a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta)^2 + 2(a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta)a_{pq} \sin 2\theta + (a_{pq} \sin 2\theta)^2 \\ &\quad + (a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta)^2 - 2(a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta)a_{pq} \sin 2\theta + (a_{pq} \sin 2\theta)^2 \\ &= (a_{pp} \cos^2 \theta + a_{pp} \sin^2 \theta)^2 + (a_{qq} \cos^2 \theta + a_{qq} \sin^2 \theta)^2 + 2(2a_{pp}a_{qq} - a_{pp}^2 - a_{qq}^2) \sin^2 \theta \cos^2 \theta \\ &\quad + 2(a_{pp} \cos^2 \theta + a_{qq} \sin^2 \theta)a_{pq} \sin 2\theta - 2(a_{pp} \sin^2 \theta + a_{qq} \cos^2 \theta)a_{pq} \sin 2\theta + 2(a_{pq} \sin 2\theta)^2 \\ &= a_{pp}^2 + a_{qq}^2 + \frac{1}{2}(2a_{pp}a_{qq} - a_{pp}^2 - a_{qq}^2) \sin^2 2\theta + 2(a_{pp} - a_{qq})a_{pq} \cos 2\theta \sin 2\theta + 2a_{pq}^2 \sin^2 2\theta \\ &= a_{pp}^2 + a_{qq}^2 - \frac{1}{2}(a_{pp} - a_{qq})^2 \sin^2 2\theta + 2(a_{pp} - a_{qq})a_{pq} \cos 2\theta \sin 2\theta + 2a_{pq}^2 \sin^2 2\theta \\ &\quad (\text{对角元为零可知: } \frac{1}{2}(a_{qq} - a_{pp}) \sin 2\theta + a_{pq} \cos 2\theta = 0) \\ &= a_{pp}^2 + a_{qq}^2 - 2a_{pq}^2 \cos^2 2\theta + 4a_{pq}^2 \cos^2 2\theta + 2a_{pq}^2 \sin^2 2\theta \\ &= a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 \end{aligned} \quad (8.23)$$

矩阵 B 的非对角元的平方和为:

$$\sum_{i \neq j} b_{ij}^2 = \sum_{i \neq j} a_{ij}^2 - 2a_{pq}^2 \quad (8.24)$$

矩阵 B 的对角元的平方和为：

$$\sum_{i=1}^n b_{ii}^2 = \sum_{i=1}^n a_{ii}^2 + 2a_{pq}^2 \quad (8.25)$$

可以认为非对角元转移到了对角元上。

雅可比方法需要进行一系列的迭代，设第 m 次迭代：

$$S_m = R_i \cdots R_m \quad (8.26)$$

$$A_m = S_m^T A S_m \quad (8.27)$$

因为在运算过程中原来消除的非对角元可能又会出现，所以 m 一般很大，我们写成：

$$\lim_{m \rightarrow \infty} A_m = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} \quad (8.28)$$

$$U = \lim_{m \rightarrow \infty} S_m \quad (8.29)$$

其中  $\lambda_i$  为本征值， $U$  的第  $i$  列为本征矢。

## 8.2 QR 算法

雅可比迭代过程，需要有限的变换和 QR 算法

### 8.2.1 Householder 变换

householder 变换 (Householder transformation)，译为“豪斯霍尔德变换”，或译“豪斯霍德转换”，又称初等反射 (Elementary reflection)，这里老师讲的是矩阵可以变换为三对角矩阵。

对一个矩阵的各个列向量逐一进行相应的豪斯霍尔德变换，可以将这个矩阵变换为上海森伯格矩阵 (次对角元以下的所有元素都为 0)、上三角矩阵等形式。后者就是 QR 分解的豪斯霍尔德算法

### 8.2.2 QR 算法

存在  $A = QR$ ,  $Q$  为正交矩阵,  $R$  为上三角矩阵。通过一系列初等变换:

$$\begin{aligned} A_1 &= A = Q_1 R_1 \\ A_2 &= Q_1^{-1} A_1 Q_1 = R_1 Q_1 \\ A_2 &= Q_2 R_2 \\ A_3 &= Q_2^{-1} A_2 Q_2 = R_2 Q_2 \\ &\vdots \end{aligned}$$

可以证明最终的结果为:

$$\lim_{m \rightarrow \infty} A_m = \begin{bmatrix} \lambda_1 & \times & \times \\ 0 & \ddots & \times \\ 0 & \cdots & \lambda_m \end{bmatrix}$$

$$\text{or } \lim_{m \rightarrow \infty} A_m = \begin{bmatrix} \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & \times & \times & 0 \\ 0 & & \times & \\ & & & \ddots \end{bmatrix}$$

或者是上三角矩阵, 其对角线上的元素就是本征值; 或者是上分块对角矩阵, 成块的地方要单独计算本征值, 其他对角线上的元素就是本征值。

How to Understand IT Technology (信息技术)?

1. 吴军——《浪潮之巅》2019 年 7 月第四版;
2. 算法的力量——Google 公司的创立

## 8.3 厄米矩阵的本征值与本征矢

厄米矩阵的特点:

1. 本征值为实的;

2. 本征矢是完备的（正交归一的）。

数值方法：

1. 实对称矩阵推广到复对称矩阵；

2. 厄米矩阵化为实对称矩阵。

现在讨论第二方法：

设厄米矩阵为  $n \times n$  的方阵，满足本征方程：

$$C\psi = \lambda\psi \quad (8.30)$$

现在厄米矩阵化为实对称矩阵，设：

$$\begin{aligned} C &= A + iB \quad A、B \text{ 为实矩阵} \\ \psi &= u + iv \quad u、v \text{ 为实向量} \end{aligned} \quad (8.31)$$

把上式代入本征方程得：

$$\begin{aligned} (A + iB)(u + iv) &= \lambda(u + iv) \\ (Au - Bv) + i(Av + Bu) &= \lambda u + i\lambda v \end{aligned} \quad (8.32)$$

实部、虚部对应相等得到：

$$\begin{cases} Au - Bv = \lambda u \\ Av + Bu = \lambda v \end{cases} \Rightarrow \begin{pmatrix} A & -B \\ B & A \end{pmatrix}_{2n \times 2n} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \end{pmatrix} \quad (8.33)$$

下面讨论矩阵  $A$ 、 $B$  的性质，利用矩阵  $C$  的厄米性：

$$\begin{aligned} C^\dagger &= C \\ (A + iB)^\dagger &= A^T - iB^T = A + iB \end{aligned} \quad (8.34)$$

可见： $A = A^T$ （实对称）， $B = -B^T$ （实反对称），下面讨论  $A$  和  $B$  构成的矩阵的性质：

$$\begin{pmatrix} A & -B \\ B & A \end{pmatrix}^T = \begin{pmatrix} A^T & B^T \\ (-B)^T & A^T \end{pmatrix} = \begin{pmatrix} A & -B \\ B & A \end{pmatrix} \quad (8.35)$$

可见这个重组的  $2n \times 2n$  的矩阵是实对称矩阵。下面举两个例子说明上式：

$$B = \begin{pmatrix} 0 & b \\ -b & 0 \end{pmatrix}, \quad \begin{pmatrix} A & -B \\ B & A \end{pmatrix} = \begin{pmatrix} & 0 & -b \\ A & & b \\ & b & 0 \\ 0 & b & \\ & & A \\ -b & 0 & \end{pmatrix} \quad (8.36)$$

$$B = \begin{pmatrix} 0 & -b & -c \\ b & 0 & -a \\ c & - & 0 \end{pmatrix}, \quad \begin{pmatrix} A & -B \\ B & A \end{pmatrix} = \begin{pmatrix} & 0 & b & c \\ A & -b & 0 & a \\ & -c & -a & 0 \\ 0 & b & c \\ -b & 0 & a & A \\ -c & -a & 0 \end{pmatrix} \quad (8.37)$$

现在我们考虑  $C_{n \times n}$  有  $n$  个本征值，而  $\begin{pmatrix} A & -B \\ B & A \end{pmatrix}$  有  $2n$  个本征值两者有何关系？

假定  $\begin{pmatrix} A & -B \\ B & A \end{pmatrix}$  的某一本征值为  $\lambda$  相应的本征矢为  $\begin{pmatrix} u \\ v \end{pmatrix}$ ，则  $\begin{pmatrix} -v \\ u \end{pmatrix}$  也是  $\lambda$  的本征

矢。只需验证如下等式成立：

$$\begin{pmatrix} A & -B \\ B & A \end{pmatrix} \begin{pmatrix} -v \\ u \end{pmatrix} = \lambda \begin{pmatrix} -v \\ u \end{pmatrix} \Rightarrow \begin{cases} -Av - Bu = \lambda(-v) \\ -Bv + Au = \lambda u \end{cases} \quad (8.38)$$

与前面得到的  $A$  和  $B$  的方程组对比，可以发现上式是成立的，因此  $\begin{pmatrix} -v \\ u \end{pmatrix}$  也是本征矢。



这说明  $\begin{pmatrix} A & -B \\ B & A \end{pmatrix}$  的每一个本征值都是二重简并的, 因此如果  $\lambda_1, \lambda_2, \dots, \lambda_n$  是厄米矩阵  $C$  的本征值,  $\begin{pmatrix} A & -B \\ B & A \end{pmatrix}$  的本征值有  $2n$  个:  $\lambda_1, \lambda_1, \lambda_2, \lambda_2, \dots, \lambda_n, \lambda_n$ , 因此每个  $\lambda_i$  对应的本征矢  $\psi$  为  $u + iv, i(u + iv)$  只相差一个相位, 这在量子力学中代表相同的态。

由于课时的限制这里只简单介绍了一些微分方程的内容。



# 微分方程

## 常微分方程

- 初值问题——龙格-库塔法（Runge-Kutta methods）是用于非线性常微分方程的解的重要的一类隐式或显式迭代法。这些技术由数学家卡尔·龙格和马丁·威尔海姆·库塔于 1900 年左右发明。
- 边值问题：
  1. 打靶法：用某种离散化数值步骤求出常微分方程边值问题在离散点上的近似解的方法。适当选择和调整初值条件，求解一系列初值问题使之逼近给定的边界条件。如果把将描述的曲线视作弹道，难么求解过程即不断地调试条件使之达到预定的靶子，所以乘坐打靶法或试射法。
  2. 有限差分（中心差分公式）：有限差分是形式为  $f(x+b)-f(x+a)$  的数学表达式。如果有限差分除以  $b-a$ ，则得到差商。有限差分导数的逼近在微分方程数值解的有限差分方法，特别是边界值问题，起着关键的作用。中心差分公式：
$$\delta_h f(x) = f\left(x + \frac{1}{2}h\right) - f\left(x - \frac{1}{2}h\right)$$

## 偏微分方程

- 有限差分法
- 有限元法

参考图书：徐自新《微分方程近似解》1990 年



# 第9章 随机模型模拟

数值模拟引论。

数值模拟

- 确定性方法：分子动力学方法。
- 随机方法：
  1. 随机模型模拟
  2. Monte Carlo 方法

## 分子动力学方法

处理大量（N 个）粒子组成的体系时，我们可以得到一些物理量的平均值（热统里的一些量）：

$$\bar{A}(t_0) = \langle A(t) \rangle = \frac{1}{t} \int_{t_0}^{t_0+t} A(\vec{r}^N, \vec{p}^N) d\tau \quad (9.1)$$

A 是由 N 个粒子的位形（坐标和动量）决定的物理量，用经典力学的方法（牛顿方程等等）可以求得，但是在早期这是无法计算的。因此人们转变了思路把对时间的累积变为对微观状态的处理，而这一切都是在等概率假设的基础上的，这样 A 的平均值就是某一微观状态下的值乘以概率求和：

$$\langle A(t) \rangle = \sum_s A(s) e^{-\frac{E_s}{k_B T}} \quad (9.2)$$

但是现在可以用计算机来处理这种多粒子系统随时间的演化，而且证明了以上两种结果在一定条件下是一致的。

参考书：J.M.Haile, Molecular Dynamics Simulation: Elementary Methods. 1992

随机模拟可以用 Monte Carlo 方法模拟，按马尔可夫链作演化，同样对于 N 个粒子的问题，某一位形出现的几率为：

$$P(\{\vec{r}\}, \{\vec{p}\}) \propto e^{-\frac{E(\{\vec{r}\}, \{\vec{p}\})}{k_B T}} \quad (9.3)$$

其中  $\{\vec{r}\}, \{\vec{p}\}$  表示 N 个粒子的位形，可以看出能量越低出现的概率越大，最终得出较为准确地结果是有中心极限定理的影响。

积分变为按几率求和:

$$\int A(\vec{r}, \vec{p}) P(\vec{r}, \vec{p}) d\vec{r} d\vec{p} = \frac{1}{N} \sum_{\vec{r}, \vec{p}} \quad (9.4)$$

按马尔可夫链演化简单来讲就是后一个位形只由前一个位形决定, 经过很多这样的过程就达到平衡。

参考书:《统计力学》马上庚

这里简单地说明一下中心极限定理: 相互独立的随机变量, 随着测量次数的增多, 其分布满足高斯分布。正常来说某一物理量的测量应该是在平均值附近, 两边的偏离是由误差造成的, 由于误差很多种而且是相互独立的, 因此随着测量次数的增多误差满足中心极限定理, 进而测量的物理量也满足, 即服从高斯分布。

参考书:

Krauth W. Introduction To Monte Carlo Algorithms, 1996.

Werner Kraut. Statistical mechanics: algorithms and computations, 2006

## 随机模型模拟

重要的是建立新的模型。

随机模型的模拟: 根据建立的随机模型 (最好自己建立), 利用随机数直接构造算法做模拟。

### 9.1 随机模型

#### 9.1.1 单位圆求 $\pi$

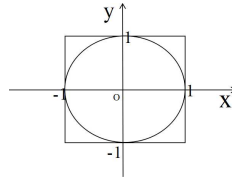


图 9.1: 单位圆计算  $\pi$

随机方法中需要用到随机数 (Random numbers), 标椎的随机数在  $[0, 1]$  内均匀独立分布。现在在  $[-1, 1]$  区间内随机地产生  $(x, y)$ , 可以对标椎随机数做一些变换:

$$0 \leq r \leq 1 \Rightarrow 0 \leq 2r \leq 2 \Rightarrow -1 \leq 2r - 1 \leq 1 \quad (9.5)$$

$$x = 2r_1 - 1, \quad y = 2r_2 - 1 \quad (9.6)$$

若落入圆内点的个数为  $N_{\text{圆}}$ , 总的个数为  $N_{\text{圆}}$ , 我们可以知道:

$$S_{\text{圆}} \propto N_{\text{圆}} \quad (9.7)$$

可以认为:

$$S_{\text{圆}} = kN_{\text{圆}} \quad (9.8)$$

其中  $k$  为平均每个点所代表的体积, 同理有  $S_{\text{总}} = kN_{\text{总}}$ , 这样就可以得到:

$$\begin{aligned} \frac{S_{\text{圆}}}{S_{\text{正}}} &= \frac{N_{\text{圆}}}{N_{\text{总}}} \\ \pi &= 4 \frac{N_{\text{圆}}}{N_{\text{总}}} \end{aligned} \quad (9.9)$$

### 9.1.2 浦丰投针

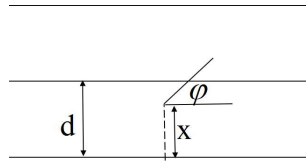


图 9.2: 浦丰投针示意图

设有一系列平行线, 其间距为  $d$ , 有一根长度为  $l(l < d)$  的针, 现向这组平行线投针, 我们想知道针与某一平行线相交的几率, 在如图所示的  $x, \varphi$  坐标系中, 我们有  $0 \leq x \leq d, 0 \leq \varphi \leq \pi$ , 可以得到相加条件为:

$$x + l \sin \varphi \geq d \Rightarrow x \geq d - l \sin \varphi \quad (9.10)$$

画出图像如图所示

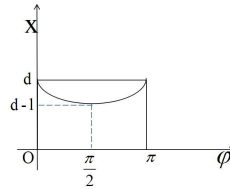


图 9.3: 浦丰投针计算  $\pi$

可见能相交的区域是曲线与矩形所围的上半部分, 可以得到概率为:

$$P = \frac{\pi d - \int_0^\pi (d - l \sin \varphi) d\varphi}{\pi d} = \frac{2l}{\pi d} \quad (9.11)$$

最终得到  $\pi$  的计算公式:

$$\pi = \frac{2l}{dP} \quad (9.12)$$

这一模模型可以通过真实地投针模拟, 也可利用计算机模拟。

### 9.1.3 渗流模型 (Percolaton Model)

渗流模型中需要将空间分成许多格子, 例如分成  $N \times N = 50 \times 50$  的格子, 每一个格点有粒子 (电子, 水分子等) 占据的概率为  $P$ ,  $[0, 1]$  区间取一随机数  $r$ , 与  $P$  比较, 若  $r > P$ , 则有

粒子占据,  $r < P$ , 就没有粒子占据。相邻点都占据就是导通的, 如果边界间可以通过某一条通路连接, 则称是导通的。这种模型可以模拟金属-绝缘体相变。通过模拟可以知道  $P$  有一临界值  $P_c = 0.593$ ,  $P$  小于这一值就一直不导通。

### 9.1.4 分形结构

#### 有限聚集扩散模型

(PRL.47(1981)1400) 可以模拟一些聚集体（灰尘）的形成。

同样需要将空间分成许多格子, 在格子中有一粒子（相当于聚集中心）。边界的粒子向四个最近邻方向的一定概率都是  $\frac{1}{4}$ （跨出边界的就舍去了, 而且边界上的粒子也是随机取的）, 当边界粒子运动到与初始粒子近邻时就不再运动了（相当于沾上了）, 重复这一过程, 直到与粒子团近邻, 这样就逐渐形成了分形结构。

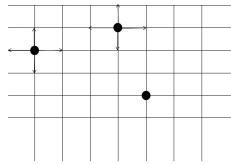


图 9.4: 有限聚集扩散模型

随机模型的重点就是如何建模和随机数的生成, 下面来介绍随机数。

## 9.2 随机数

### 随机数

是指  $[0, 1]$  之间任一个数以等概率出现并形成相互独立的序列。（等概率, 相互独立, 序列（就是很多）, 都是统计意义上的）基本特点:

1. 均匀性
2. 随机性

实际上有些随机数是伪随机数, 即不满足上述要求。下面来介绍如何产生随机数。

## 9.3 随机数发生器——产生随机数

### 9.3.1 抛硬币

抛硬币产生的结果虽然是随机的, 但是不满足随机数的要求, 因为只有两种情况:

$$P\{\xi_i = 1\} = P\{\xi_i = 0\} = \frac{1}{2} \quad (9.13)$$

How to understand "Probability and Statistics"?

1. 科普: 穆尔——“统计学的世界”;



2. 方法和“诡辩”;

3. 三门问题 (最大几率上  $\frac{2}{3}$ , 先三选一, 去掉一个后再二选一  $P = \frac{1}{3} + \frac{2}{3} \times \frac{1}{2} = \frac{2}{3}$ )。

### 9.3.2 用程序算出来的随机数 (伪随机数)

1. 平方取中法。

例 9.14 选一初始数  $\xi_0 = 3281$

$$\begin{aligned}
 \xi_0 &= 3281 \\
 z_1 &= 3281 \times 10^{-4} = 0.3281 \\
 \xi_0^2 &= 10764961, \text{ 取中间 4 位令:} \\
 \xi_1 &= 7649 \\
 z_1 &= 0.7649 \\
 \xi_2 &= 58507201 \\
 z_2 &= 0.5072 \\
 &\vdots
 \end{aligned} \tag{9.14}$$

其中的  $\{z_n\}$  可以构成随机数序列。

平方取中法缺陷:

- 有周期性 (重复);
- 不实用 (运算速度慢)。

优点: 利用了递推关系。

2. 同余法。

- 乘法同余法:

递推关系:

$$X_{n+1} = \text{Mod}(\lambda X_n, M), \quad (\text{Mod}(a_1, a_2) = a_1 - \left[ \text{int} \left( \frac{a_1}{a_2} \right) \times a_2 \right]) \tag{9.15}$$

其中 Mod 表示一种运算规则, 如括号中所示, int 表示取整。最终得到的随机数为  $Z_n = \frac{X_n}{M}$  ( $\lambda, M$  都是取较大的正整数), 可以构成随机数序列  $\{Z_n\}$ , 但是这一方法, 在参数确定后, 随机数也是确定的, 也有一定的周期, 只是其周期小于  $M$ , 对于 32 位的计算最大  $M = 2^{31} - 1$ 。

- 线性同余法:

在乘法同余法的基础上做了些改进, 递推公式为:

$$\begin{aligned}
 X_{n+1} &= \text{Mod}(\lambda X_n + C, M) \\
 Z_n &= \frac{X_n}{M}, \text{ 周期} < M
 \end{aligned} \tag{9.16}$$

C 语言中，默认  $M = 32767$

### 3. 随机数生成程序 (Press.et.al):

- 随机数生成程序 1 (Rand I)

有如下几点说明:

- 基于乘法同余法:  $\lambda = 7^5$ ,  $M = 2147483647$ ;
- 防止溢出 (Schrange 算法);
- 克服相关性 (混沌算法)。

周期为  $2.1 \times 10^9$  (32 字节), 这种方法得到的随机数序列已有较好地统计性质、均匀性、独立性, 可以用于实际计算。

- 随机数生成程序 2 (Rand II)

把两个不同周期的随机数序列组合, 生成序列, 周期为  $2.3 \times 10^{180}$  (32 位计算机)

### 9.3.3 随机数的统计检验

判断好坏的标准: 1、均匀性; 2、独立性; 3、周期大小。

#### 1. 均匀性检验:

取大量的随机数, 待统计稳定后, 绘制柱状图 (频数分布直方图):

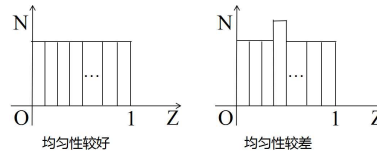


图 9.5: 均匀性检验

#### 2. 独立性检验:

$$\begin{aligned}
 C_r &= \frac{1}{N} \sum_{i=k+1}^{N+k} (x_i x_{i+r} - \bar{x}_i^2), \quad (r \text{ 为两随机数间的间隔}) \\
 S^2 &= \frac{1}{N} \sum_{i=k+1}^{N+k} (x_i^2 - \bar{x}_i^2) \\
 \bar{x} &= \frac{1}{N} \sum_{i=k+1}^{N+k} x_i
 \end{aligned} \tag{9.17}$$

### 9.3.4 实际应用

1. 随机数序列质量好, 1、均匀性好; 2、独立性好; 3、周期长;
2. 实际用到的随机数的数量不大于随机数周期的 5%

# 第 10 章 机器学习导论

## 引言

### 发展历史

#### 1、人工智能（思考、行为）

让机器做本需要人的智能才能够做到的事情的一门科学。

1956 年，AI 诞生，由此产生了不同的流派。

1. 符号主义：基于符号表示进行推理，建立系统，实现人工智能。分为推理期和知识期，但最终都不合适（不成功）。
2. 连接主义：仿生学派，模仿生物（其他方面），建立模型，如：神经网络模型。
3. 机器学习：1980 年独立出来，建立模型，让大量数据来训练它（即学习），然后具有某种能力。

发展阶段：推理期 → 知识期 → 学习期，前两个最终都失败了行不通。

#### 2、机器学习

##### 1. 机器学习发展为人工智能的核心内容：

- 统计机器学习——复杂
- 神经网络机器学习——不可解释性？
- 神经决策树

##### 2. 数据挖掘（Data mining）的核心之一：

- 机器学习
- 统计方法
- 其它方法，如 PageRank 算法（Google）

##### 3. 大数据（Big Data），数据分析（System analyse），运营调研（Operation research）

## 神经网络和深度学习

由 Hinton(Boss, 父亲的姥爷是布尔),Bengio,Lelun 提出, 获 2018 图灵奖。

深度学习与前经典机器学习的区别:

1. 以前人工提取特征 (feature), 现在发展为 “自动提取自动化”;
2. 分散表示 (零散信息, 向上汇聚);
3. 发展了表示学习——feature;
4. Hinton 的想法简单且实用。

基本原理:

$$\begin{cases} y = ax + b \\ f(x) = \Theta(x) \end{cases} \quad (10.1)$$

即线性 + 非线性。

应用范围: 科学技术、工程、商业、医学等。

## 我们的目的

1. 扩展一下视野;
2. 关注神经网络, 切入机器学习或神经挖掘;
3. 作为一种理论方法与工具, 从新的角度研究有关物理问题;
4. Find a favorite subject;
5. Find a job;
6. ...with a very little probability of success.

- PageRank 算法 (涉及矩阵、图论), S.Brinand (布林), L.Page (佩奇), 创立 Google.
- James Simons (Chern-Simons 理论), Renaissance Technologies. 他提出一个数学模型 (math model):

$$\text{大量数据} \rightarrow \text{math model} \rightarrow \text{output} \quad (10.2)$$

## 参考书

1. 周志华, 机器学习
2. S.Haykin, 神经网络与机器学习
3. Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. The MIT Press, Cambridge.
4. 综述文章: Lecun, Y., Bengio, Y., Hinton, G., et al. (2015) Deep Learning. Nature, 521, 436.

## 10.1 神经网络的架构

### 10.1.1 人的大脑的信息处理

由神经元（神经细胞）构成，神经元连接起来形成神经网络。

外界信息  $\rightarrow$  神经网络  $\rightarrow$  响应 (10.3)

类似于物理测量中：干扰  $\rightarrow$  响应。

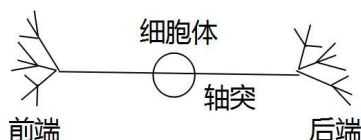


图 10.1: 一个神经元

How to communicate with others?

1. 卡耐基的书，1936 年原始版本；另一个角度看问题；
2. 说话时让人感到舒适；
3. 达到程度。

### 10.1.2 神经元模型

简化的神经元模型如下：

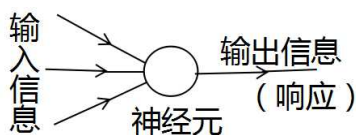


图 10.2: 一个神经元模型

神经元的非线性模型 (Mcluttoch-Pitts 模型，麦卡洛克-皮特斯模型，1943 年)

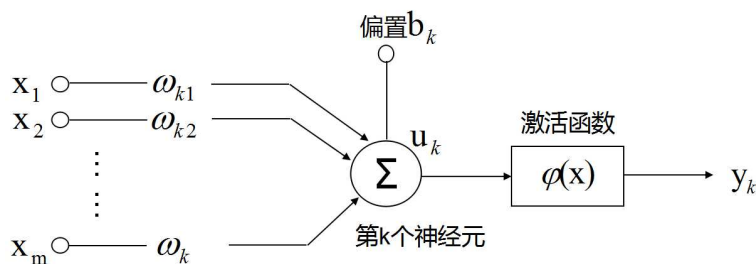


图 10.3: McCulloch-Pitts 模型

其中  $x_1, x_2, \dots, x_m$  为输入数据（信号、信息），每个量的贡献不同，其中  $\omega_{k1}, \omega_{k2}, \dots, \omega_{km}$  各输入参量的权重，是表征存储信息的参量， $u_k, y_k$  表示如下

$$u_k = \sum_{j=1}^m \omega_{kj} x_j + b_k, \quad y_k = \varphi(u_k - \theta) \quad (10.4)$$

其中  $b_k$  为偏置量， $\theta$  为阈值。激活函数为：

$$\varphi(x) = \theta(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases} \quad (10.5)$$

可以简单表述为：

$$\begin{cases} y = ax + b & \text{线性项} \\ y = \theta(y) & \text{非线性项} \end{cases} \quad (10.6)$$

上述模型只有两种响应状态：有反应、没反应。进一步可以推广到多个响应状态。实际中阶跃函数不好表示，通常采用容易算（好做的）近似函数模拟阶跃函数：

1. Sigmoid 函数：

$$\varphi(x) = \frac{1}{e^{-ax} + 1}, \quad a > 0 \quad (10.7)$$

上述函数很像 Fermi 分布函数：

$$f(E) = \frac{1}{e^{(E-\mu)/k_B T} + 1} \quad (10.8)$$

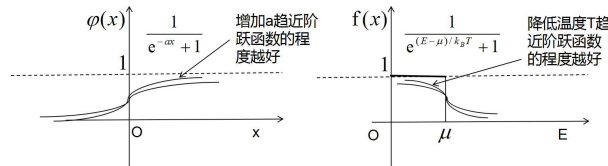


图 10.4:  $\varphi(x)$  与  $f(E)$

从图中可以看出两这很像，变化一下自变量在作镜像对称即可得到彼此。

2. ReLU 函数：

$$g(z) = \max\{0, z\} \quad (10.9)$$

其图像如下图所示：

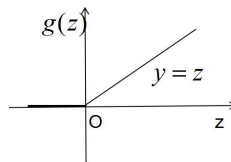


图 10.5:  $g(z)$  图像

神经元模型用图像表示（用简化模型表示 McCulloch-Pitts 模型，但本质不变）

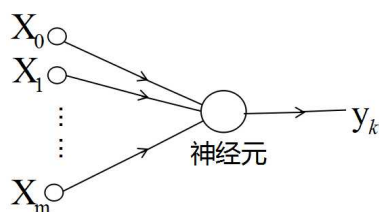


图 10.6: 简化的神经元结构图

其中  $x_0 = 1, \omega_{k0} = b_k$ ，带权重求和以及激活函数都在神经元中。

### 10.1.3 神经网络结构（建模，Building model）

由神经元相互连接起来构成神经网络。

用什么结构？由神经科学的启发，采用多层结构（层状结构）。

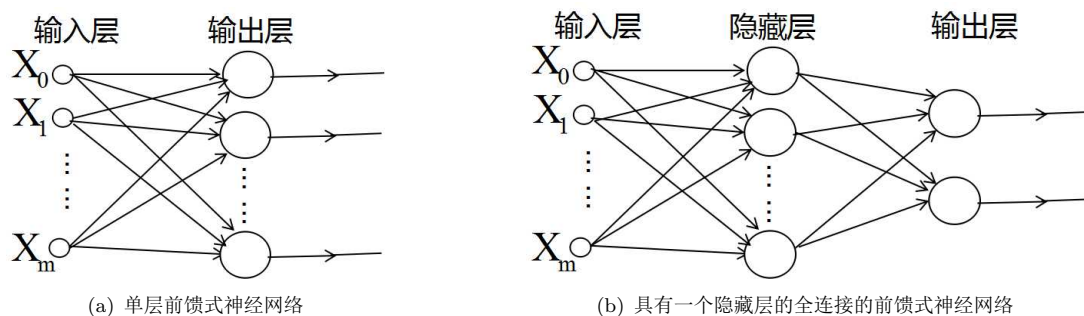


图 10.7: 神经网络

输入层只是将信号传入，不作处理。神经网络中可以加入很多层。

上述的神经网络的功能根据其结构的变化而变化，类似于物理中的结构决定性质。

这种模拟试图摆脱仿生学的影响（不受神经科学的影响），推广到更多范围。

## 10.2 神经网络和学习算法

分为监督学习和非监督学习。主要讨论监督学习。

### 10.2.1 监督学习的过程

1. 训练: 利用大量样本，通过计算，调节权重，确定权重。
2. 测试: 有了权重，用已知样本检查输出结果与原来已知结果是否一致。
3. 泛化（推广）: 模型推广到新的情况会怎样？

### 10.2.2 反向传播算法 (Back propagation, BP 算法)

这种方法基于梯度下降法（最陡下降法，最小二乘法，最优化方法，有这些方法的影子），其特点是可以自动确定权重  $\omega$ 。BP 算法涉及的参数如下图所示：

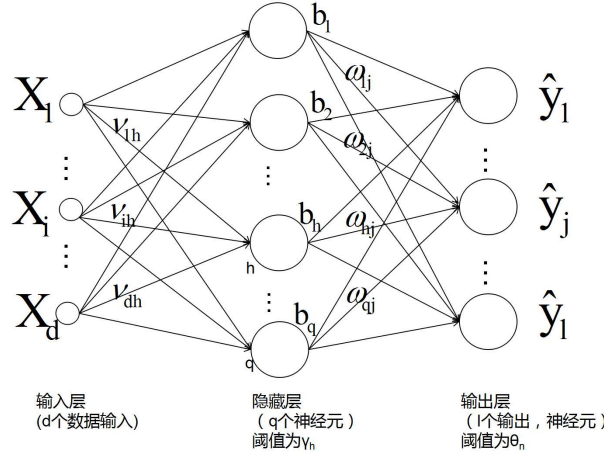


图 10.8: 标准 BP 算法中涉及的参数

下面简述其计算过程：

权重随机取初始值，输入数据  $x_1, x_2, \dots, x_d$ ，隐藏层第  $h$  个神经元输入为：

$$\alpha_h = \sum_{i=1}^d \nu_{ih} x_i \quad (10.10)$$

输出为：

$$b_h = f(\alpha_h - \gamma_h) \quad (10.11)$$

输出层第  $j$  个神经元输入为：

$$\beta_j = \sum_{h=1}^q \omega_{hj} b_h \quad (10.12)$$

最终输出为：

$$\hat{y}_j = f(\beta_j - \theta_j) \quad (10.13)$$

对于一个训练样本  $(\vec{x}, \vec{y})$ ,

$\vec{x} = x_1, x_2, \dots, x_d$  表示输入的特征

$\vec{y} = y_1, y_2, \dots, y_l$  表示标签（实际结果）

通过神经网络计算出的输出结果用  $\hat{y}$  表示： $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_l$ 。

为了进行优化需要计算  $\vec{y}$  与  $\hat{y}$  间的差异，第  $k$  组样本  $(\vec{x}_k, \vec{y}_k)$ ，计算出  $\hat{y}_k = (\hat{y}_1^k, \dots, \hat{y}_l^k)$ ，差异记为：

$$E_k = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j^k - y_j^k)^2 \quad \text{类似于最小二乘法} \quad (10.14)$$

需要调节参数  $\omega_{hj}, \nu_{ih}, \gamma_h, \theta_j$ ，来使差别变小，引入 BP 算法。



## BP 算法

调节  $\omega_{hj}$  为例。

我们已经表示出差异  $E_k(\omega_{hj}, \nu_{ih}, \gamma_h, \theta_j)$ ，为了减少差异，现在把  $\omega_{hj}$  用  $\omega_{hj} + \Delta\omega_{hj}$  代替（自动更新，使差异变小），即  $\omega_{hj} \leftarrow \omega_{hj} + \Delta\omega_{hj}$ 。

现在我们要来确定  $\Delta\omega$ ，因为差异是  $\omega_{hj}$  的函数，存在偏导数  $\frac{\partial E_k}{\partial \omega_{hj}}$ ，有下面两种情况：

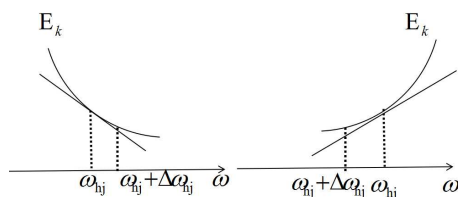


图 10.9:  $E_k$  随  $\omega_{hj}$  变化的可能情况

由图像分析可知（两种情况是一样的，都是要调制  $\omega_{hj}$  向  $E_k$  变小的方向移动） $\Delta\omega_{hj}$  总可以取如下形式：

$$\Delta\omega_{hj} = -\eta \frac{\partial E_k}{\partial \omega_{hj}} \quad (10.15)$$

其中  $\eta \in (0, 1)$ ，称为调节参数、学习效率，这个式子主要记得它包含导数的负数。

具体过程是要先算出  $E_k$ ，再算出  $\omega_{hj} + \Delta\omega_{hj}$  代替  $\omega_{hj}$ ，重新计算  $E_k$  直到差异最小，这是一个自动优化的过程。

现在我们手动计算一下导数  $\frac{\partial E_k}{\partial \omega_{hj}}$ ：

$$\frac{\partial E_k}{\partial \omega_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \frac{\partial \hat{y}_j^k}{\partial \beta_j} \frac{\partial \beta_j}{\partial \omega_{hj}} \quad (10.16)$$

其中  $\frac{\partial \beta_j}{\partial \omega_{hj}} = \frac{\partial \sum_{h=1}^q \omega_{hj} b_h}{\partial \omega_{hj}} = b_h$ ，我们选择的激活函数为 Sigmoid 函数，可以得到：

$$f'(x) = \left( \frac{1}{e^{-ax} + 1} \right)' = \frac{ae^{-ax}}{(e^{-ax} + 1)^2} = af(x)(1 - f(x)) \quad (10.17)$$

（上课时老师似乎取了  $a=1$ ）为计算方便：

$$g_j = \frac{\partial E_k}{\partial \hat{y}_j^k} \frac{\partial \hat{y}_j^k}{\partial \beta_j} = -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta) = a \hat{y}_j^k (1 - \hat{y}_j^k) (y_j^k - \hat{y}_j^k) \quad (10.18)$$

最终代回到  $\Delta\omega_{hj}$  中得到：

$$\Delta\omega_{hj} = \eta g_j b_h \quad (10.19)$$

同样地，有：

$$\Delta\theta_j = -\eta g_j \quad (10.20)$$

$$\Delta\nu_{ih} = \eta e_h x_i \quad (10.21)$$

$$\Delta\gamma_h = -\eta e_h \quad (10.22)$$

其中

$$\begin{aligned}
 e_h &= -\frac{\partial E_k}{\partial b_h} \frac{\partial b_h}{\partial \alpha_h} = -\frac{\partial b_h}{\partial \alpha_h} \sum_{j=1}^l \frac{\partial E_k}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial b_h} \\
 &= -\frac{\partial b_h}{\partial \alpha_h} \sum_{j=1}^l \frac{\partial E_k}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \beta_j} \frac{\partial \beta_j}{\partial b_h} = ab_h(1-b_h) \sum_{j=1}^l \omega_{hj} g_j
 \end{aligned} \tag{10.23}$$

BP 算法需要反复迭代自动调节权重使差异最小，上述为标准的 BP 算法。

### 累积的 BP 算法

$$\text{差异取为 } E = \frac{1}{m} \sum_{k=1}^m E_k.$$

### 误差反向传播算法

信息从左向右传递，误差从右向左传播（反向传播）。

## 10.3 简单应用，手写数字识别

tensorflow 库，手写数字识别，mnist 识别（最经典算例之一，网上有详细的介绍）。

### 10.3.1 数据集

数据集由收集的大量数据构成，但收集数据的过程一般比较困难，例如手写书就有许多中写法。

实际中将手写书  $0, 1, \dots, 9$ ，每个构成黑白图片，一个图案是  $28 \times 28 = 784$  像素（一个像素就是一块小区域），这样就可以把手写书的图案用矩阵表示，空白为 0，黑色为 1。

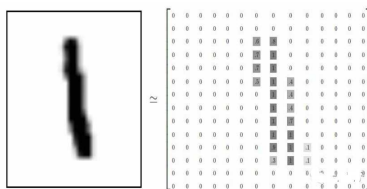


图 10.10: 对图像的变化

标签的变化（各手写数字对应的数字）写为 10 个 10 行列向量（相当于输出层有 10 个神经

元)。

$$0: \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad 1: \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad 2: \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \cdots \quad (10.24)$$

实际中收集了 55000 个样本（统计中：一个样本包含许多个体，这里应该是指 55000 个手写书样本），分成训练集（50000 个）和测试集（5000 个）

### 实际训练与测试

1. 像素矩阵和标签对应放在一起
2. 矩阵变为一维 784 个元素的行向量  $\vec{X}$ ，标签变为一维 10 个元素的行向量  $\vec{y}^*$ 。

$$\vec{X} = (x_1, x_2, \dots, x_{784}), \quad \vec{y}^* = (y_1^*, y_2^*, \dots, y_{10}^*) \quad (10.25)$$

### 10.3.2 构建模型

采用单层前馈式全连接神经网络。

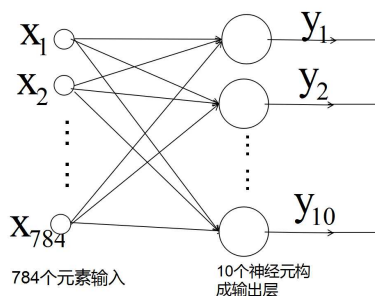


图 10.11: 数字识别神经网络示意图

有一个中间变量  $u_j$ ，表达式为：

$$u_j = \sum_{i=1}^{784} \omega_{ji} x_i + b_i, \quad (j = 1, 2, \dots, 10) \quad (10.26)$$

需要做进一步代换：

$$u_j \rightarrow e^{u_j} \rightarrow \frac{e^{u_j}}{\sum_{j=1}^{10} e^{u_j}} = y_j \quad (10.27)$$

这种代换类似于得到热统中的配分函数，获得输出几率的含义（几率最大的输出为 1，其它为 0?）。

对于某一张图片的标签，也就是正确分类是这样的形式：(1,0,0,...)，对于这张图片，我们的模型的输出可能是 (0.5,0.2,0.3,...) 这样的形式，为了恒量这种差异，这里引入交叉熵。

### 交叉熵

交叉熵（cross entropy）的概念取自信息论，刻画的是两个概率分布之间的距离，一般都会用在分类问题中。

前面提到过香农熵：

$$H(p) = - \int dx P(x) \log P(x) = \langle I(x) \rangle, \quad T(x) = - \log P(x) \quad (10.28)$$

两个几率分布间的差异（具体怎么来的，网上有，简单来说，是用一个概率代替另一个概率作差，相对熵，Kullback-Leibler (KL) divergence, KL 散度）：

$$\begin{aligned} D_{kl}(p||Q) &= \int dx p(x) [\log p(x) - \log Q(x)] = \int dx p(x) \log p(x) - \int dx p(x) \log Q(x) \\ &= -H(p) + \int dx p(x) \log \frac{1}{Q(x)} \end{aligned} \quad (10.29)$$

前一部分是信息熵（香农熵），后一部分即为交叉熵：

$$H(p, q) = \int dx p(x) \log \frac{1}{q(x)} \quad (10.30)$$

这里  $p$  表示真实的分布，如 (1,0,0,...)， $q$  为计算出来的分布，如 (0.5,0.3,0.2,...)，对应我们的参数有：

$$H = - \sum_{i=1}^{10} y_i^* \log y_i \quad (10.31)$$

其中  $y_i^*$  为标签 (label)， $y_i$  为计算出来的。

利用 BP 算法，调节权重，使交叉熵最小化。

### 10.3.3 训练

权重  $\omega$  的初始值随机给定，循环训练 1000 次，每次取 100 个样本，累积交叉熵，利用 BP 算法优化，训练好后，把权重有效存起来。

### 10.3.4 测试

将 5000 个测试集合输入，得到的结果是否与标签一致？

$$\text{准确率} = \frac{\text{正确数量}}{\text{总数}} \times 100\% \quad (10.32)$$

### 10.3.5 泛化

用更多的手写数测试。

### 真正的实际计算

程序语言和计算工具（Python（Anaconda 版本，有一些自带的库）或 C 语言 +TensorFlow（张量流动）库）

### 参考书籍：

郑泽宇，TensorFlow：实战 Google 深度学习框架（第二版），2018 年

## 10.4 伊辛模型的相变识别

### 参考文献：

Carrasquilla J , Melko R G . Machine learning phases of matter[J]. Nature Physics, 2016.

## 10.5 伊辛模型

伊辛模型是解释低温下物质铁磁-顺磁相变的一个简单模型，该模型下的哈密顿量为：

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z, \quad \sigma_i^z = \pm 1 \quad (10.33)$$

伊辛模型只考虑自旋间的经典相互作用，J 为自旋间的相互作用能。实际中发现存在一个临界温度  $T_c$ ，

$$\begin{aligned} T < T_c, & \quad \text{有序相, 铁磁相} \\ T > T_c, & \quad \text{无序相, 顺磁相} \end{aligned} \quad (10.34)$$

理论计算得简单晶格中  $T_c = 2.269K$

### 10.5.1 Wolff 算法（集团算法）

对于二维的伊辛模型，可以用二维的平面来模拟，比如可以将平面分为  $10 \times 10$  的小平面，为了计算出哈密顿量，我们需要算出一组可能的位形  $\{\sigma_i\}$ ，平衡状态下的位形分布满足玻尔兹曼分布：

$$P(\{\sigma_i\}) = \frac{1}{Z} e^{-E(\{\sigma_i\})/k_B T} \quad (10.35)$$

实际上在模拟过程中可以采取马尔科夫链 Monte Carle 方法，模拟逐渐的相变过程，最终可以获得平衡态下的位形分布。

### 10.5.2 神经网络识别 Ising Model 相变

该神经网络可以简化如下：

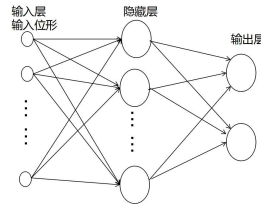


图 10.12: 识别 Ising Model 相变神经网络示意图

输出层有两个神经元，对应输出为：

$$\text{有序} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{无序} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (10.36)$$

这类似于一种分类问题，可以有如下关系，

$$\begin{array}{c} T < T_c & T > T_c \\ \hline \begin{pmatrix} 1 \\ 0 \end{pmatrix} & T_c & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{array}$$

图 10.13: 分辨相变的随温度的变化

可以看出在临界温度上下，可以较好分辨，靠近临界温度时，但是不能较好确定相变温度，因为这是有限体积的相变问题，随着体积的增大，其相变点也会随之发生变化，但是可以得到一个大致范围。

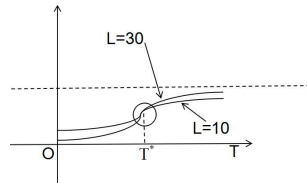


图 10.14: 不同尺寸的比较

### 10.5.3 相变温度的确定

前面提到了确定相变点的困难，我们可以采取维象的的标度律：用直线去拟合一些不同体积下得到的散点，最终合理外推出相变温度。

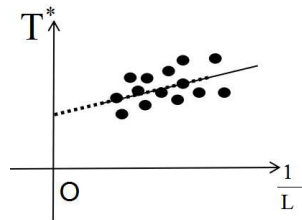


图 10.15: 标度律拟合