世界著名计算机教材精选

PEARSON
Addison Wesley

数值分析与科学计算

Jeffery J. Leader 著

张 威 刘志军 李艳红 等译

NUMERICAL ANALYSIS AND SCIENTIFIC COMPUTATION

清华大学出版社

PEARSON Education

世界著名计算机教材精选

数值分析与科学计算

Jeffery J. Leader 著

张 威 刘志军 李艳红 等译

清华大学出版社

北京

图书在版编目（CIP）数据

数值分析与科学计算/(美)里德(Leader, J. J.)著;张威等译.—北京:清华大学出版社,2008.5
(世界著名计算机教材精选)
书名原文:Numerical Analysis and Scientific Computation
ISBN 978-7-302-16914-7

Ⅰ. 数…　Ⅱ. ①里… ②张…　Ⅲ. ①计算机辅助计算:数值计算—教材 ②计算机辅助计算—软件包,MATLAB—教材　Ⅳ. 0241　TP391.75

中国版本图书馆 CIP 数据核字(2008)第 008937 号

责任编辑:龙啟铭
责任校对:张　剑
责任印制:何　芊

出版发行:清华大学出版社　　**地　　址**:北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**:100084
社　总　机:010-62770175　　**邮　　购**:010-62786544
投稿与读者服务:010-62776969,c-service@tup.tsinghua.edu.cn
质　量　反　馈:010-62772015,zhiliang@tup.tsinghua.edu.cn
印 刷 者:清华大学印刷厂
装 订 者:三河市溧源装订厂
经　　销:全国新华书店
开　　本:185×260　**印　张**:32.25　　**字　数**:741 千字
版　　次:2008 年 5 月第 1 版　　**印　次**:2008 年 5 月第 1 次印刷
印　　数:1～3000
定　　价:59.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103　　产品编号:024348-01

译者序

虽然数值分析属于数学课程，但它与我们以往所熟悉的数学课程是完全不同的，从基本的思想方法和思维方式到课程的学习方式都有很大的差异。我们所熟悉的数学课程的共同特点是抽象和严格的演绎、严密和清晰的逻辑思维，但对于数值分析课程来说，除了上述特点之外，“归纳”将成为不容忽视的思维方法，这里讨论的核心问题是“误差”这个在数学上似乎根本不存在的东西。在以往的数学课程中，教师可能会强调阅读教科书和做习题，但在数值分析课程中，除了读书和做习题之外，用计算机计算也是同样重要的。

数值分析是培养学生算法意识和能力的基本课程，应从培养学生科学计算能力出发，但这一点绝不能靠单纯讲授来解决，必须在计算机环境下通过实验完成，从这一点上来说，数值分析方面的书籍应该更多地面向实际应用，或许是数值分析所涉及的范围太广，大多数数值分析教材特别是国内的教材都侧重于基本算法的推导和理论分析的细节，其目的在于给读者打下一个较为坚实的理论基础，在此基础上设计和编写自己的算法和程序，但实际上在过去的十几年中，数值分析的实践和教学已经发生了巨大的变化，人们不必再去书写那些繁杂的代码和程序，可以更多的依赖像 MATLAB 这样的软件包，同时人们所解决的问题的规模也越来越大，因而对计算机硬件的要求也越来越高。这种新的形势要求我们不仅要了解算法的基本思想及推导过程，还要进一步了解怎样才能更好地使用这些算法，其中包括如何使算法在合理的时间内收敛，如何选择适当的初值、如何对各种方法进行结合等等，从这一点上说科学计算更加重要。

Jeffery J. Leader 教授的这本教材采用数值分析与科学计算并重的思想，很好地解决了大部分现有教材没有很好解决的问题。书中没有过多地进行枯燥的定理论述，也没有过多地分析算法的细节，重点介绍了方法基本思想以及在 MATLAB 平台上的使用，其目的在于通过数值实验提高学生对算法的“鉴赏”能力，使学生熟练使用标准的计算机软件，了解各种算法的优缺点，最终能“拥有”这些算法。

本书每小节后面的习题可以使读者加深理解本小节所介绍的基本问题，这些问题一般不需要其他的附加资源；MATLAB 部分介绍了与本小节内容相关的 MATLAB 命令以及相应的数值实验，这部分的内容可以使读者逐渐认识这个软件并有机会通过数值实验获得对科学计算的直观认识，这些内容即使不在教学中讲授也可以作为自学 MATLAB 的参考书，在翻译此书的过程中译者就曾在数学建模培训中以这部分内容作为教材介绍过 MATLAB 软件，收到了很好的效果；附加问题特别是序号 10 以后的附加

题可能完成起来要有一定的难度，可有选择的完成。

本书结构合理，可读性强，除了可以作为本科高年级或研究生的“数值分析”教材，对以科学计算为工具的科技人员也有很大的参考价值。

翻译过程中我们力求忠实、准确地反映原著的内容和风格，但由于水平所限，难免会有疏漏及不妥之处，恳请读者批评指正。

在翻译此书的过程中得到了众多同仁的热情帮助和支持，在此特别感谢高福顺教授和杨月婷教授，他们在百忙中仔细审阅了原稿并提出了不少有价值的意见和建议；赵雪、姜晓威、赵阳几位老师参与了部分章节的翻译工作；张明达、姜元政、路云龙、姚卓、吕学哲、张羽、聂思奇等同志对书中 MATLAB 部分的内容进行了上机实验，在此一并表示感谢。最后对本书审稿人和编辑的出色工作表示诚挚的谢意。

前言

FOREWORD

数值分析是对连续数学问题的算法的研究。 Lloyd Trefethen(1992)

初等数值分析是一门非常具有吸引力的学科。只需花费很少的精力利用很少的代数知识就可以得出重要的算法、写出代码并执行。 G. W. Stewart (1998)

本书是在过去9年我所讲授的数值分析课程的基础上编写而成的，该课程最初是工科大学二年级的一门必修课，目前已成为数学、计算机科学、物理学和工科各年级学生的选修课。近年来我暑期在美国陆军研究实验室(Adelphi MD)、海军作战中心(Dahlgren VA)和空军研究实验室(Dayton OH)所做的计算咨询顾问工作以及其他咨询和研究工作的经历都对此书产生了很大影响，同时与其他同事的座谈以及所翻阅的其他教材也对本书有很大帮助。

我希望本书在以下方面与其他数值分析教材有所不同：

- 首先给出数值方法，将截断误差和计算机运算推迟到学生具有一定实际计算经验之后。
- 采用数值线性代数的最新方案。
- 用学生在实际中愿意使用的主要计算程序(包括 MATLAB 以及 Maple 和 Netlib 库)解释数值技术。
- 将数值分析理论和实际的科学计算原理有机地结合。
- 更注重最优化的分析。
- 通过数值实验提高学生的能力。
- 自然而有效地介绍 MATLAB。

每节后面以附加内容的方式介绍了 MATLAB 的相关内容，如果希望学生在学习本教程的同时学会 MATLAB，那么这部分的内容可以使他们逐渐认识这个软件并有机会通过数值实验获得对科学计算的直观认识。即使在教学中不使用 MATLAB 也应鼓励学生大致浏览一下这部分的内容以便将来使用。当然使用本教材时也可以不使用包含 MATLAB 的部分。我曾利用上述方案在课堂上进行了 MATLAB 的教学，效果很好。

科学计算的实践告诉我们，从某种程度上说，几乎任何问题最终都可以化简为求根、非线性优化或数值线性代数问题。前两个问题中往往也涉及线性方程组的求解。本书从

求根问题(第 1 章)开始,主要介绍一维求根,这是因为学生在微积分中已经了解了这个问题,然后介绍数值线性代数(第 2、3 章),因为这部分内容不仅本身重要,而且后面的章节中还要使用,第 4 章介绍作为推导其他方法的工具的多项式插值以及描绘由数据给出的曲线的工具的样条,第 5 章介绍数值求积的基本技术并重点介绍这些方法在 MATLAB 和 Maple 中的使用,第 6 章主要处理常微分方程组的求解,非线性优化的相关内容在第 7 章介绍,第 8 章讨论逼近理论的基本思想和方法。

本书试图避免落入那种先介绍一种完全新的数学思想,然后再给出关于这种思想的计算技术的套路中。如果学生还没有了解相关的数学问题,像求根、最大值和最小值、微积分中的微分和积分以及微积分或常微分方程课程中的一阶常微分方程,那么对于他们来说就很难理解为什么要了解关于这些问题的数值方法。例如对于富里叶变换,本书并没有先介绍富里叶变换,然后再介绍其离散形式,最后再给出快速傅里叶变换的计算方法,但是在介绍 QR 分解和奇异值时却违背了这个原则,因为这部分的内容在矩阵或线性代数教程中并没有标准化。

在课堂上,我经常让学生思考有关求根问题的具体方案或让学生对所学到的方法进行改进。为了模拟书中的方法,有些章节中还简要讨论了所介绍方法的替换和改进,即使在课堂上及教科书中没有详细的介绍,这样做是很有必要的。到本课程结束,学生们应该能够提出一些诸如用二次逼近代替线性逼近等等的问题,当然也可以跳过这些讨论。

为了强调方法背后的基本思想以及将各种方法进行融合,我总是鼓励学生使其感觉到他们“拥有”了这些方法。通常情况下,学生们总是感觉他们无法完全掌握现有的方法和软件,这时就需要多进行试探,以我的经验,随意地去“篡改”那些方法也许是大有好处的,这就是为什么人们总是使用而不是编写算法的主要原因,数值软件需要这样的过程。必须知道怎样才是好的初始值,怎样才能逼近导数,怎样才能选择正确的方法和参数。

在过去的十几年中,数值分析的实践和教学已经发生了巨大的变化,几乎很少有人再去写那些繁杂的代码和程序,人们更多的是依赖像 MATLAB 以及类似像 Octave、GAUSS 那样的软件包,同时人们所解决问题的规模也越来越大,从这方面说,还要感谢那些高级的计算机硬件以及 Netlib、NAG 那些强有力的软件。但即使这样也还是要进一步了解算法背后的基本思想以及怎样使这些算法在合理的时间内收敛,其中包括选择适当的初值、各种方法的结合以及调整算法的参数,当然对如何详细地写出算法的代码则可以有很少的了解(事实上,书写代码的工作可以留给专业的团队去完成),与数值分析相比,科学计算更加重要。

从教学层面上看,许多学校都在课程中减少了微积分的理论并很大程度上引入了计算机代数,这样就产生了两方面的问题:首先,教师必须按学生的知识背景去考虑问题;其次,学生往往习惯于看到计算机代数系统所给出的答案是数值形式的,而不是符号的。今天学习数值分析的学生可能不像从前那样具有数值计算的经验,这一点要归功于计算机代数系统。本书的目的在于通过强调形象思维(如线性和二次逼近)以及列举一些计算机代数系统不能解决的例子说明学习数值方法的必要性。

书中也提到了几种方法的并行问题,但没有做详细介绍。虽然学生应该知道这方面

内容的存在,但我认为数值分析的初等课程中并不应该过分强调这一点。即使学生能接触到这样的机器,但编程却可能超出了他们的能力。

本书不属于参考书,所以学生只有彻底读懂才能学会这门课程。书中尽量避免枯燥的定理论述,尽量通过严格的推导说明方法背后的思想。我认为学生在具有一定的数值计算经验之前不可能对那些枯燥的理论感兴趣,所以本书的目的在于让学生能够熟练地使用标准的计算机软件,其中包括选择完成任务的方法,同时也阐述了其他方法的主要优缺点。虽然我已经成功地做到了这一点,但还是希望你们也同样能够做到这一点。

书中的每个小节可以作为一讲,但也因人而异。习题部分包括用来加深理解内容的基本问题,除了前一小节刚刚讲过的内容之外,这些问题一般不需要其他的附加资源;MATLAB 部分介绍 MATLAB 命令以及数值实验的问题,学生可以将这部分内容作为自学 MATLAB 的参考书。通常我要求学生输出并上交他们的命令日志以及对结果进行简单的评价;有些附加问题可能需要与 MATLAB 中所介绍的内容等价的其他计算工具,序号在 10 以后的附加题可能更具有挑战性,最后两三个附加题通常相当困难。

本书所包含的教学内容可能会超过一个季度或一个学期。前 10 周我通常讲授下面的部分:1.1～1.9;2.1～2.9;3.1～3.6;4.1～4.4;5.1～5.3 以及 5.6 和 5.7 的基本思想;6.1～6.3 以及 6.5 和 6.6 的基本思想;7.1～7.3(如果时间允许可以加上 7.4 和 7.5)。我在讲授的过程中比较偏重于数值线性代数,因为许多其他的问题,像偏微分方程数值解等通常都需要这方面的知识,其他领域的许多问题也都需要进行线性代数方程组的求解。第 5、6 章中的许多内容都是以第 2 章的后半部分为基础的,由于存在这种依赖关系,所以 1.1～1.4,1.7,2.1～2.3 和 4.1 是必讲的。

本课程的先修课程包括一年的微积分、矩阵代数基础以及第 6 章所涉及的一阶常微分方程。第 8 章内容要求的条件可能更高一些,通常需要熟练地掌握一些其他数学知识。另外本课程还需要一些编程经验(如果使用 MATLAB)。

我的岳父曾经在他撰写的一本书的序言中写道:“一本书的完成需要众人的力量。”对此我深信不疑。我要感谢出版这本书的 Addison-Wesley 出版社,特别还要感谢 Addison-Wesley 的 Cindy Cody、Joe Vetere 和 RoseAnne Johnson; Cris Miller 和她的团队;责任编辑 Louise Gache;Michael Brown 编写了本书的习题解答并对本书进行了详细的检查;其他还有许多人也为本书做了大量的工作。

在此还要提及许多评论家对本书的评论,他们对内容的顺序、各段落的繁简程度都给出了很好的建议,同时还指出了书中的一些错误和疏漏。Rose-Hulman 工程学院的 4 届学生都曾使用过本书的手稿,他们对本书的进一步加工也有很大的帮助,在此也一并表示感谢,我希望在此提到他们中的所有人。书中所遗留的所有错误都归咎于我自己。

在完成本书的过程中,Rose-Hulman 工程学院的同事们给了我很大的帮助,在此特别要感谢 S. Allen Broughton、Ralph P. Grimaldi 和 Robert Lopez(目前在 Maplesoft)对我的鼓励和建议。

多年来我有幸和众多数值分析学者共同进行研究和工作,这些经历使我在这个领域不断成长,我特别要感谢我的导师 Brown 大学的 Philip J. Davis;Brown 大学的 Gottlieb

以及 Naval Postgraduate 大学的 Bill Gragg。

还要感谢的是在我职业生涯中对我帮助极大的两个人：Harvey Mudd 学院的 Robert L. Borrelli 和 Courtney S. Coleman。

最后还要感谢我妻子 Meg 对我的帮助，感谢她和我们的两个孩子 Derek 和 Corrinne 在我完成本书期间对我的耐心。

Jeffery J. Leader

Terre Haute, IN

目录

CONTENTS

第 1 章 非线性方程

1.1 对分法和反线性插值

科学或工程问题的求解和模拟最终往往都要解决求根或优化问题. 前一种情形要求出方程或方程组的解;后一种情形则要找出使函数取最大或最小值的点. 即使是对实验数据进行拟合或数值求解微分方程,也总是将问题简化成上述两类问题. 本章主要考虑非线性方程的求根问题,第 2 章和第 3 章考虑线性方程组,第 7 章讨论优化问题.

所谓求根就是要求出 $f(x)=0$ 的解 x^*(也可能有多个解,但只需找出其中的一个). 方程的解也叫作方程的**根**或函数 f 的**零点**. 通常对特殊的情形有特殊的方案:如果 f 是二次式,则可以使用二次求根公式,$\sin(x)$ 的零点大家都知道. 但不久之后,需要求解的将不再是这些特殊的情形. 最简单的例子是像

$$\cos(x) - x = 0$$

这样的方程以及 5 次或 5 次以上的多项式方程. (3 次和 4 次多项式都有求根公式,但可以证明 5 次及 5 次以上的多项式则没有类似的公式,除非多项式有明显的分解式,否则只能用数值方法.) 更为复杂但却非常普遍的情形是求微分方程的解,而微分方程本身必须用数值方法求解.

一种可能的方法是试探法. 这种方法本身并没有问题,但缺乏对收敛速度估计的理论(在给定误差限内),而且很难自动进行. 在实际操作中,每运行一次程序,不同的求根问题都可能求解几十或几百次.(例如,用计算机辅助设计软件包确定曲线的交点.) 因此,需要找到快速、可靠而且简单的方法,最好是在处理过程中不需要人工干预.

我们考虑的是第一种方法是**穷举搜索**(也称**直接**、**图解**或**增量搜索**). 假定 f 在某个区间(不必有限)上是连续的. 由介值定理,如果能够找到两个点 a,b,使 $f(a)$ 和 $f(b)$ 的符号相反,则 (a,b) 中一定存在 f 的零点. 找出这样两个点的方法之一是先选定某个 x_0,以这个值作为根的初始近似,依次计算函数在

$$x_0, \quad x_1 = x_0 + h, \quad x_2 = x_0 + 2h, \quad x_3 = x_0 + 3h, \quad \cdots$$

上的值,其中 $h>0$ 称为**步长**(或**网格尺寸**),一旦检测到符号变化,则在这个区间内得到了长度为 h 的有限区间(称符号发生变化的区间 $[x_i, x_{i+1}]$ 为**有根区间**,见图 1.1),在这个小一些的区间 $[x_i, x_{i+1}]$ 上可以用更小的 h 重复上述过程,以便更精确地确定根的位置.

穷举搜索法等价于先画出函数的图像,然后找出图像穿过 x 轴的区间. 这种方法效率很低,下面要寻求更好的方法.

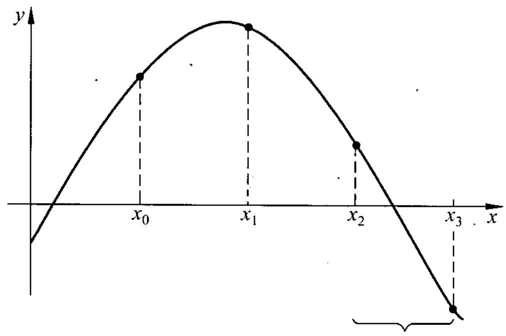

图 1.1 零点的划界

假定已通过某种手段得到了关于连续函数的有根区间$[a,b]$. 这里并不使用更小的步长对整个区间进行搜索,而是将中点

$$m=\frac{a+b}{2}$$

看成是 f 的零点位置的近似，这个值比 a 和 b 的近似效果更好. 因为如果 x^* 接近 b,那么$|a-x^*|$可能会和区间长度 $\omega=(b-a)$一样大,反之对于$|b-x^*|$也有类似的结论,但由于 x^* 不是位于 m 的左边,就是 m 的右边(除了极端情形 $x^*=m$),所以

$$|m-x^*|\leqslant\frac{1}{2}\omega$$

这就是**对分法**(或**二分法**)的基本思想:如果 f 在所考虑的区间上连续,且$[x_0,x_1]$是有根区间,即 $f(x_0)f(x_1)<0$,则取

$$x_2=\frac{x_0+x_1}{2}$$

并计算 $f(x_2)$. 如果 $f(x_2)=0$ 则结束,否则或者 $f(x_2)$与 $f(x_0)$符号相反,此时$[x_0,x_2]$是新的有根区间,长度为原来的一半，或者 $f(x_2)$与 $f(x_1)$符号相反,此时$[x_2,x_1]$是新的有根区间,长度也为原来的一半. 不管是哪种情形,都通过计算一个新的函数值($f(x_2)$的计算)将零点 x^* 位置的不确定性降低了 50%. 接下来在这个新的区间上重复这个过程,即求出区间的中点 x_3,然后找出以 x_3 为端点的更小的有根区间. 继续这个过程,直到有根区间的长度充分小.

例 1.1.1 考虑函数 $f(x)=\cos(x)$,这个函数的一个零点为 $\pi/2\doteq1.5708$(符号$\doteq$表示值已按给定的有效数位进行了舍入),由于 $f(1)\doteq0.5403$,$f(2)\doteq-0.4161$,且 f 连续,因此$[1,2]$为有根区间,中点 $x_2=1.5$. 由

$$f(1.5)\doteq0.0707,$$

$f(1.5)f(2)<0$,因此可以用 $x_2=1.5$ 代替 $x_0=1$,即$[1.5,2]$是改进的有根区间. 下一个中点$x_3=1.75$,且

$$f(1.75)\doteq-0.1782,$$

所以新有根区间为$[1.5,1.75]$. 如果不再继续,则$[1.5,1.75]$中的任意值都可以作为零点位置的近似. 若选择这个区间的中点 $x_4=1.625$,则可以使最差误差最小化. ■

由于舍入误差的影响，在上述过程中某些正值可能会变成负值，负值可能会变成正值，除非我们因此会选择错误的区间，否则上述方法一定会收敛到 f 在初始有根区间中的某个零点. 也就是说，随着迭代次数 k 的增加，对于位于初始有根区间中的某个满足 $f(x^*)=0$ 的 x^*，一定有

$$\lim_{k\to\infty} x_k = x^*.$$

进一步，由于有根区间的长度每次减半，所以还可以预测出达到所需精度所需要的迭代次数. 如果初始有根区间的长度为 w，则第一次对分后新有根区间的长度为 $w/2$，第二次对分后长度为 $w/4$. 一般的进行 n 步之后，有根区间的长度为 $w/2^n$. 这样为使初始长度为 1 的有根区间的长度降低为10^{-4}，n 必须足够大，即

$$\begin{aligned}\frac{1}{2^n} &\leqslant 10^{-4}\\ 10^4 &\leqslant 2^n\\ n &\geqslant \log_2(10^4)\\ &\doteq 13.2877.\end{aligned}$$

所以 $n=14$ 次迭代就够了. 由于误差最多为10^{-4}，因此结果中小数点后面的四位数字都是正确的(也可能第 4 位小数会少一个单位). 当然，虽然这样做可以保证最终区间的长度是 2^{-14}(注意 $2^{-14}\leqslant 10^{-4}\leqslant 2^{-13}$)，并且中点是精度在 2^{-15}之内的近似，但实际的误差可能更小. 事实上，如果以[1,2]为初始区间，对 $f(x)=\cos(x)$进行 14 次迭代，并用最终区间的中点 $x_{16}\doteq 1.57077$ 作为零点$\pi/2=1.57079\cdots$的近似值，则实际误差为 2.6×10^{-5}，误差上界为 3.1×10^{-5}，误差上界表示的是最差情形的误差，通常我们的结果相对而言是最佳的.

由对分法产生的序列一定能收敛到根；但穷举搜索则不然(如果 h 不是充分小，这种方法可能会恰好越过两个紧挨着的根，见图 1.2). 通常对分法的速度很快，但为此也要付出一定代价：对分法需要给出初值的有根区间，而穷举搜索法只需提供一个能落在事先假定的根的左边的一个初始值. 为了得到初始有根区间可以使用初始的图解搜索(或试探).

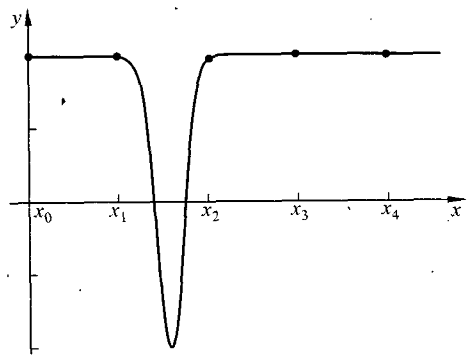

图 1.2 穷举搜索遗漏一个脉冲

再考虑函数 $f(x)=\cos(x)$，这里假定已经得到的初始有根区间为[0, 1.6]，这时 $f(0)\doteq 1.0000$，$f(1.6)\doteq -0.0292$，对分法可以进行，但要注意 $|f(1.6)|$ 非常小于 $|f(0)|$，所以选择的新点应更接近 $x_1=1.6$，如果取 $x_2=1.4$，则 $f(x_2)\doteq 0.1700$，所以[1.4,1.6]是一个新的更小的有根区间；若用对分法，则得到的有根区间为[0.8,1.6]，长度是这个区间的 4 倍. 由此得到另一种方法，通常称为**(反)线性插值**(也称**试位法**)：给定一个连续函数 f 及关于 f 零点的初始有根区间$[x_0, x_1]$，对点$(x_0, f(x_0))$和$(x_1, f(x_1))$作直线拟合(见图 1.3)，这条直线称为 f 在这些点上的插值，如果 f 在这个区间上近似为线性的，则可以用直线的零点近似函数的零点(这种

方法之所以称为反插值，是因为我们是用插值直线去求 x 的值，而不是 y，这里将 x 看成 y 的线性函数)，由于两个 y 值的符号相反，$f(x_0)$不等于 $f(x_1)$，所以可以用点斜式将直线方程写成：

$$x=\frac{x_1-x_0}{f(x_1)-f(x_0)}y+\left(x_1-\frac{x_1-x_0}{f(x_1)-f(x_0)}f(x_1)\right) \tag{1.1}$$

并通过设 $y=0$，解出它的 x 截距 x_2：

$$x_2=x_1-f(x_1)\frac{x_1-x_0}{f(x_1)-f(x_0)}. \tag{1.2}$$

与前面的讨论一样，$[x_0,x_2]$或$[x_2,x_1]$将成为新的更小的有根区间. 继续这个过程，直到满足某种**收敛准则**，即满足能够确定已充分接近真解的某种准则. 因为这些区间的长度可能不趋近于零(当单侧逼近方程的根时，见习题 1)，所以不能用区间长度作为唯一的收敛准则. 同样道理，除非有根区间的长度非常小，否则最后一次计算的端点一般是根的最好的近似，而不是中点.

反线性插值虽然可以收敛到函数在有根区间内的某个零点；但其收敛速度取决于 $f(x)$在接近零点时的线性程度. 由微积分知识，如果 $f(x)$是充分可微的，则在很小的区间上，可以用直线进行很好的近似. 对于实际中碰到的一般类型的函数，这个结论说明反线性插值通常比对分法快一些(对于图 1.4 中的函数和有根区间，只要有根区间没有达到充分小，收敛就是极其慢的)，这样做失去了对分法的收敛保障以及虽然速度慢但却稳定的性质，采用了另外一种类似的但即没有保障也不容易改进的方法.

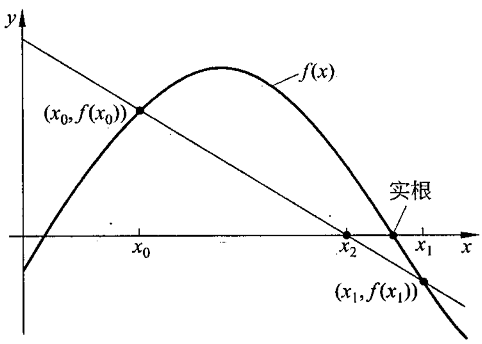

图 1.3 反线性插值

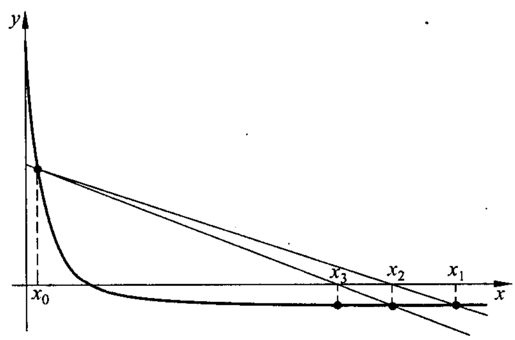

图 1.4 刚性函数的反线性插值

下一节将考虑一种收敛非常快的方法——牛顿(Newton)法. 使用这种方法同样要付出代价：要求函数可微. 快速方法往往需要更多的假定条件，同时提供的保障也会更少一些.

习题 1.1

1. 在初始有根区间[0.5,1]内用 4 次对分法迭代求 $\cos(x)-x=0$ 的唯一正根. 取同样的初始有根区间，用反线性插值重复上述过程，注意解的单侧逼近性.
2. 证明方程 $5x^7=1-x$ 有唯一的实数解，用对分法求这个解并精确到 4 位小数. 再用反

线性插值重复上述过程.

3. 函数 $f(x)=\cos(5x)$ 在区间 $[0,4.5]$ 内有 7 个零点. 在这个初始有根区间内(先证明其确实是有根区间),先用对分法再用反线性插值求出一个根并至少精确到 3 位小数. 用已知的余弦函数的零点位置求出近似值的误差并对结果加以说明.
4. 利用函数 $f(x)=x^2-5$,用对分法求 $\sqrt{5}$ 的近似值并至少精确到 4 位小数. 将近似误差与实际误差相比较.
5. 以 $[0.25,2]$ 为初始有根区间,用反线性插值求 $f(x)=1/x^3-10$ 的零点并精确到 3 位小数. 用对分法重复上述过程并加以说明.

MATLAB 1.1

数值分析主要研究的是如何设计能够解决科学和工程中的数学问题的计算方法,这些方法一般都是采用迭代的形式并且具有合理的计算时间及令人满意的误差性质. 这些方法也可以不用计算机完成,有些方法是与个人的名字联系在一起的——如牛顿(1642—1727)、欧拉(Euler)(1707—1783)和高斯(Gauss)(1777—1855). 这些个人所能使用的计算工具比我们现在使用的要简陋得多.

显而易见,通过在计算机或计算器上运行这些方法可以更好的积累经验. 本书主要在 MATLAB 环境下讨论,但也可以使用 C++、Fortran、Maple、Mathematica 或其他计算工具. 即使教学过程中不使用 MATLAB,阅读这些内容并在你所使用的计算环境中进行计算也是很重要的.

MATLAB 软件包是一种交互式计算环境和程序设计语言(由 MathWorks 公司出品). 这一节内容的学习可以在计算机上进行并在阅读时输入一些命令,也可以改变所使用的函数. 双击 MATLAB 图标(Windows 中)或在提示符下输入 matlab 或 matlab 6 (Unix 中)就可以运行 MATLAB. 在下面内容

```
Commands to get started:  intro,  demo,  help help
Commands for more information:  help,  whatsnew,  info,  subscribe
≫
```

之后可以看到一个交互窗口. ≫符号是 MATLAB 的提示符. 我们现在使用的是 MATLAB 的即时模式——本质上就像一个计算器.

现在做几个简单计算,在≫提示符下,输入 3+4 然后按回车键. 即输入:

```
≫ 3+4
ans=
    7
```

继续输入:

```
≫ 3-4
ans=
    -1
```

```
>> 3*4
ans=
     12
>> 3/4
ans=
     0.7500
>> ans ^2
ans=
     0.5625
>>
```

变量 ans 由 MATLAB 自动赋值．也可以将值直接赋给其他变量．输入：

```
>> x=5, y=2
x=
   5
y=
   2
>> x+y, x^y
ans=
     7
ans=
     25
>>
```

（一行内可以输入多个命令．）下面试一下关于函数 $f(x)=\sin(e^x)$ 的对分法．首先画出函数的图像．输入：

```
>> x=0: 0.1 : 3
```

建立一个元素范围为 0 到 3、间距为 0.1 的行向量 x（即列表值）．输入：

```
>> y=sin (exp(x));
```

计算向量中每个 x 的 $\sin(e^x)$ 并将这些值赋给 y．语句后面的分号表示不进行结果的显示．sin 的自变量采用弧度制．输入：

```
>> y
```

（没有分号）可以看到 y 的值．现在输入：

```
>> plot (x, y)
```

就可以看到函数的图像了（plot 命令能自动将点连成连续的分段线性曲线）．输入：

```
>> grid
```

可以在图像上加上网格．再看一下这个图：显然在区间内有 6 个零点，可以有效地进行穷举搜索．

这个图形非常粗糙．现在采用更细的网格．在 MATLAB 6 或稍后的版本中将会在历