数值计算方法与 FORTRAN 语言

华南工学院　郑咸义　编

電子工業出版社

内 容 简 介

本教材以讲述计算机的科学计算应用为目的，一方面介绍 FORTRAN 程序设计语言，包括从 FORTRAN IV 到 FORTRAN 77；一方面又较系统地介绍了各种常用的数值计算方法，包括方法的推导、建立、误差分析、稳定性讨论、收敛性研究，一直到给出主要算法的通用 FORTRAN 程序以及上机算例.

本教材是自动控制专业的教科书，但对于电子、机械、土建、化工、管理以及物理、化学、力学、生物、经济等专业也是适用的．此外，也可作为从事科学计算的科技人员的学习参考书.

数值计算方法与FORTRAN语言

华南工学院　郑咸义　编

责任编辑　王惠民

*

电子工业出版社出版(北京市万寿路北口)

北京科技印刷厂印刷

新华书店北京发行所发行　　各地新华书店经售

*

开本: 787×1092　1/16　印张: 25.75　字数: 595 千字

1986 年 8 月第 1 版　1986 年 10 月第 1 次印刷

印数: 10000　定价: 4.25 元

统一书号: 15290·311

出 版 说 明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校工科电子类专业课教材的编审、出版的组织工作．从一九七七年底到一九八二年初，由于各有关院校，特别是参与编审工作的广大教师的努力和有关出版社的紧密配合，共编审出版了教材159种．

为了使工科电子类专业教材能更好地适应社会主义现代化建设培养人才的需要，反映国内外电子科学技术水平，达到"打好基础、精选内容、逐步更新、利于教学"的要求，在总结第一轮教材编审出版工作经验的基础上，电子工业部于一九八二年先后成立了高等学校《无线电技术与信息系统》、《电磁场与微波技术》、《电子材料与固体器件》、《电子物理与器件》、《电子机械》、《计算机与自动控制》、中等专业学校《电子类专业》、《电子机械类专业》共八个教材编审委员会，作为教材工作方面的一个经常性的业务指导机构，并制定了一九八二～一九八五年教材编审出版规划．列入规划的教材、教学参考书、实验指导书等共217种选题．在努力提高教材质量，适当增加教材品种的思想指导下，这一批教材的编审工作由编审委员会直接组织进行．

这一批教材的书稿，主要是从通过教学实践、师生反映较好的讲义中评选择优和从第一轮较好的教材中修编产生出来的．广大编审者，各编审委员会和有关出版社都为保证和提高教材质量作出了努力．

这一批教材，分别由电子工业出版社、国防工业出版社、上海科学技术出版社、西北电讯工程学院出版社、湖南科学技术出版社、江苏科学技术出版社、黑龙江科学技术出版社和天津科学技术出版社承担出版工作．

限于水平和经验，这一批教材的编审出版工作肯定还会有许多缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评建议，共同为提高工科电子类专业教材的质量而努力．

电子工业部教材办公室

序　　言

本教材是在高等学校工科电子类计算机与自动控制教材编委会自动控制编审小组的组织和指导下编写的.

计算机对科学技术和工程设计的巨大影响已经越来越明显了. 计算机作为科学计算方面的应用，它要求我们至少掌握一种计算机程序设计语言和熟悉一些计算机用的数值计算方法. 本教材就是以讲述计算机的科学计算应用为目标，简要地讲解了 FORTRAN 程序设计语言，包括从 FORTRAN IV 到 FORTRAN 77；又比较系统地介绍了各种常用的数值计算方法，包括方法的推导、建立、误差分析、稳定性讨论、收敛性研究，一直到给出主要算法的通用 FORTRAN 程序以及上机算例. 我们企图把程序语言与数值计算方法的学习和应用更紧密地结合起来.

本教材主要是为大学自动控制专业编写的，但对于电子、机械、土建、化工、管理以及物理、化学、生物、经济等专业也是适用的. 如果不是把教材中的所有章节都拿到课堂上讲授的话，估计 60 学时可讲完本教材的基本内容. 尽可能多争取上机实习，无疑是学好程序语言和数值计算方法的重要途径.

本教材承浙江大学储璇雯副教授主审，并经自动控制教材编审小组审定. 衷心感谢主审和编审小组对书稿的认真审阅，以及就书稿的编写和修改提出的许多宝贵意见. 感谢李树英同志的真诚帮助；本教材原是和他合写的，因他另有任务，未能具体执笔，但编写大纲是和他共同讨论确定的. 感谢严树森、张必佐、李时锦、许小谷、林复华、蔡岳忠、叶元龄等同志，以及我系 81 级应用数学专业的同学们和我院 84 届部分研究生，他们分别协助选配书中部分习题、测试书中的程序和算例、抄写了书稿. 感谢我院有关领导、教务处和应用数学系对本教材编写工作的支持和关心.

限于编者才疏学浅，实践经验不足，教材中缺点和错误一定不少. 恳切希望使用本教材的老师和同学们随时指正.

编　者

1985 年初于

华南工学院数学力学系

目　录

第1章 引 论

1-1 本书的任务

大致说来，数学模型的求解有两条途径：第一，求解析解，即解是解析表达式的形式，这是准确解；第二，如果解析解不可能或不容易求得，则求数值解，即解只是在一些离散点上取值的形式，多数情况下，数值解是近似解①．求数值解主要利用电子计算机．

我们把求数值解的问题称为数值计算问题．求解数值计算问题的方法称为**数值计算方法**，简称**数值方法**或**计算方法**．

那么，假如我们遇到数值计算问题，又有电子计算机可供我们使用的话，是不是把问题直接交给计算机，计算机就立刻把问题计算出来呢？不是的．这里有两件事情要做．第一，必须先为该数值计算问题选择合适的数值计算方法，并把它设计成可以让计算机一步步执行的所谓算法；第二，采用计算机能认识的所谓程序设计语言，把算法编码成计算机语言程序．这时，把语言程序和初始数据送入计算机，即上机计算，计算机才能（如果没有其它错误的话）把问题计算出来．这个过程可用图1-1表示．

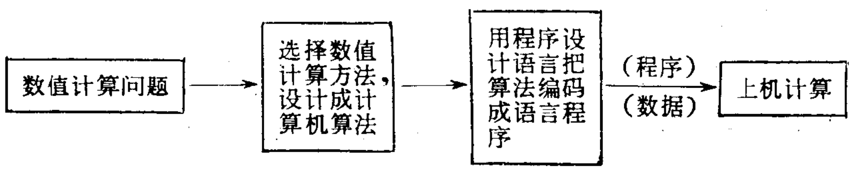


图 1-1

本书的任务是：

1）介绍一些计算机上适用的数值计算方法，研究如何把它们设计成计算机算法，讨论算法的数值稳定性、收敛性和误差估计．

2）介绍数值计算中广泛使用的 FORTRAN 程序设计语言，讨论如何把算法编码成 FORTRAN 语言程序，给出主要算法的通用 FORTRAN 程序．

必须说明的是，随着计算机的广泛应用，在数值计算方法不断发展的同时，人们已经为许多行之有效的数值计算方法设计好计算机算法，并编写了计算机语言程序．这些程序或者汇编成册（见 [c1] [c2] [c3] [c5]），供用户引用；或者直接库存在计算机系统中，供用户直接调用（CALL）．因此，实际上我们并不需要对每一个数值计算方法都去研究它的算法及其程序设计．但是，另一方面，我们也不能因此而认为，连学习数值计算方法的原理，掌握程序设计的技术，也不必要了．事实是，如果你没有为具体问题选择和使用数值计算方法的能力和知识，如果你不会动手编写自己需要的程序，或必要时读懂别人的程序，那么，你所能处理的问题在范围、深度和效率方面，将是极其有限的．

① 也有关于近似解析解的研究，例如见[c9]．

本章先讨论数值计算中的一些基本概念，然后是程序设计语言的概述，最后提供一些有关的数学补充材料.

1-2 数值计算中的一些基本概念

1. 误差概念

首先讨论误差的来源.

采用数值计算方法求数值解的整个过程，随处都有可能引进误差. 所谓误差，就是一个量的真实值与其近似值之差.

为实际问题建立数学模型时，往往进行简化和抽象，因此，所建立的数学模型与实际现象之间便可能存在误差，这种误差称为**模型误差**.

数学模型中通常包含一些根据观测或实验得到的参量，这些参量的值与真实值之间也可能存在误差，这种误差称为**观测误差**.

当数学模型不能获得准确解时，通常便采用数值方法求数值解；但数值方法中通常又采用一些近似处理，例如用收敛无穷级数前有限项之和来代替无穷级数之和，即截去无穷级数的后段. 因此，采用数值方法求得的数值解与模型本来存在的准确解之间便存在着误差，这种误差称为**截断误差**(或**方法误差**).

由于计算机只能表示有限位数并对有限位数进行运算，因此，在计算机的计算过程中需要进行舍入，这种由舍入而产生的误差称为**舍入误差**.

此外，在有的计算过程中，初始数据的误差对计算结果的影响可能很大，这种由初值引起的误差称为**初值误差**.

在数值计算方法的应用中，主要考虑截断误差、舍入误差和初值误差，研究它们的产生、传播和对计算结果的影响，尽量控制它们，使得计算结果有足够的精确度.

下面，讨论数值运算中近似值的误差概念.

定义 1 设 x 为某量的准确值，x^* 为 x 的近似值，则 x 与 x^* 之差 e^*

$$e^* = x - x^*$$

称为近似值 x^* 的**绝对误差**，简称**误差**. 显然，绝对误差可能正也可能负，而并非误差绝对值之意.

一般情况下，我们不能算出准确值 x，因而也就不能算出绝对误差 e^*. 但我们可以根据具体测量或计算情况估计出误差 e^* 的绝对值 $|e^*|$ 的某个上界 ε^*，即

$$|e^*| = |x - x^*| \leqslant \varepsilon^*$$

ε^* 称为近似值 x^* 的**绝对误差限**，简称**误差限**.

可以通过近似值 x^* 及其误差限 ε^* 来表示准确值 x 的范围：

$$x^* - \varepsilon^* \leqslant x \leqslant x^* + \varepsilon^*$$

或

$$x = x^* \pm \varepsilon^*$$

例如，近似值为 7.5304、误差限为 0.0001 的准确值 x 可表示为

$$7.5304 - 0.0001 \leqslant x \leqslant 7.5304 + 0.0001$$

或

$$x = 7.5304 \pm 0.0001$$

绝对误差在一定意义下反映了近似值的准确程度,但还未能完全刻划近似值的好坏.例如, $x_1 = 5 \pm 0.1$ 与 $x_2 = 5000 \pm 1$, 虽然近似值 $x_2^* = 5000$ 的误差限 1 是近似值 $x_1^* = 5$ 的误差限 0.1 的 10 倍,但若考虑所讨论的值本身的大小,在 5000 之内差 1 ,比在 5 之内差 0.1,显然前者的准确程度更高. 为此,我们引入相对误差的概念.

定义 2 近似值的绝对误差 e^* 与准确值 x 之比

$$e_r^* = \frac{e^*}{x} = \frac{x - x^*}{x}$$

称为近似值 x^* 的**相对误差.**

同样,准确值 x 一般是不知道的,因此,实际计算中常取相对误差为

$$e_r^* = \frac{e^*}{x^*} = \frac{x - x^*}{x^*}$$

与绝对误差类似,我们也可估计出相对误差绝对值的一个上界

$$|e_r^*| \leqslant \varepsilon_r^*$$

并称 ε_r^* 为近似值 x^* 的**相对误差限.**

由上述定义我们可知:

1) 绝对误差和误差限与所讨论量的单位有关,而相对误差和相对误差限则与所讨论量的单位无关,因此,后者常用百分比表示.

2) 同一个近似值 x^* 的绝对误差限与相对误差限有关系

$$\varepsilon_r^* = \frac{\varepsilon^*}{|x^*|}$$

例如,值 $x_1 = 5 \pm 0.1$ 的近似值 5 的相对误差限为

$$\varepsilon_r^*(x_1) = \frac{0.1}{5} = 0.02 = 2\%$$

值 $x_2 = 5000 \pm 1$ 的近似值 5000 的相对误差限为

$$\varepsilon_r^*(x_2) = \frac{1}{5000} = 0.0002 = 0.02\%$$

可见 x_2 的近似值比 x_1 的近似值相对地准确.

下面,再引入准确数字与有效数字的概念.

当数 x 有很多位时,我们常按"四舍五入"的规则取前面若干位数 x^* 作为 x 的近似值.如 $\pi = 3.14159265\cdots\cdots$,常取 3.14 或 3.1416 作近似值. 这种取法的特点是近似值的误差不超过其末位的半个单位①.

定义 3 若近似值 x^* 的误差不超过某一个位数的半个单位,便称该位数为近似值 x^* 的**准确数字**;同时,若该位数到 x^* 的第一位非零数字共有 n 位,便称这 n 位数字为**有效数字**,或称近似值 **x^* 有 n 位有效数字.**

例如, $x = 0.7136\cdots$. 若取 $x^* = 0.714$, 由于 $|x - x^*| \leqslant 0.0005$, 故 x^* 准确到小数后第三位;若取 $x^* = 0.715$, 由于 $|x - x^*| \leqslant 0.005$, 则 x^* 只准确到小数后第二位.

又例如, $x = 30.4500364\cdots$. 若取 $x^* = 30.45004$, 则它有 7 位有效数字; 若取

---

① 这里均按绝对值而言,下面的定义也同.

$x^* = 30.4500$，则它有 6 位有效数字；若取 $x^* = 30.45$，则它有 4 位有效数字.

根据定义，任意一个具有 n 位有效数字的近似值 x^* 总可以表示成标准形式

$$x^* = \pm 0.a_1a_2\cdots a_n \times 10^m \tag{1.1}$$

其中 m 是一个整数 $a_1, a_2, \cdots, a_n$ 是 0 到 9 中的一个数字而 $a_1 \neq 0$.

必须指出的是，"四舍五入"的规则有时改为：四以下舍，五以上入，刚好五时，若前面是偶数，则将五舍去，若前面是奇数，则将五进一. 实践表明，这种舍入规则有可能减少误差积累.

有效数字与绝对误差和相对误差都有密切的关系：

1）对任意一个具有 n 位有效数字的近似值 x^*，若把它写成标准形式 (1.1)，则得 x^* 的绝对误差估计式

$$|e^*| = |x - x^*| \leqslant \frac{1}{2} \times 10^{-n} \times 10^m = \frac{1}{2} \times 10^{m-n} \tag{1.2}$$

可见，近似值的有效数字的位数越多，绝对误差越小；而且只要知道了有效数字的位数，就可写出它的绝对误差限 $\varepsilon^* = \frac{1}{2} \times 10^{m-n}$.

2）对于写成标准形式(1.1)的近似值 x^*，若它具有 n 位有效数字，则由于

$$a_1 \times 10^{m-1} \leqslant |x^*| \leqslant (a_1 + 1) \times 10^{m-1}$$

于是可得相对误差估计式

$$|e_r^*| = \frac{|x - x^*|}{|x^*|} \leqslant \frac{\frac{1}{2} \times 10^{m-n}}{a_1 \times 10^{m-1}} = \frac{1}{2a_1} \times 10^{-n+1} \tag{1.3}$$

即可取相对误差限 $\varepsilon_r^* = \frac{1}{2a_1} \times 10^{-n+1}$. 可见，近似值的有效数字的位数越多，相对误差也越小.

反之，若 x^* 的相对误差限 $\varepsilon_r^* = \frac{1}{2(a_1 + 1)} \times 10^{-n+1}$，则由于

$$|x - x^*| = |x^*||e_r^*| \leqslant (a_1 + 1) \times 10^{m-1} \times \frac{1}{2(a_1 + 1)} \times 10^{-n+1} = \frac{1}{2} \times 10^{m-n}$$

故知 x^* 至少具有 n 位有效数字.

例 1 查表得$\sqrt{3} = 1.7320581\cdots$，问需取几位有效数字，才使近似值的相对误差限小于 0.1%.

解 把$\sqrt{3}$写成$\sqrt{3} = 10 \times 0.17320581\cdots$. 根据(1.3)式，要求

$$\varepsilon_r^* \leqslant \frac{1}{2a_1} \times 10^{-n+1} = \frac{1}{2 \times 1} \times 10^{-n+1} \leqslant 0.1\%$$

可知取 $n = 4$，即有

$$\varepsilon_r^* \leqslant \frac{1}{2 \times 1} \times 10^{-4+1} = 0.5 \times 10^{-3} \leqslant 10^{-3} = 0.1\%$$

故需取四位有效数字$\sqrt{3} \approx 1.732$.

我们指出，实践中通常约定：**原始数据都用有效数字写**. 也就是说，凡不标明绝对或相对误差界的近似数，都将被认为是有效数字. 例如，某产量五万八千四百二十万吨，应

写成

$$58420 \times 10^4 \text{吨}$$

它表明有 5 位有效数字，绝对误差限为 0.5 万吨．如果把它写成

$$584200000 \text{ 吨或 } 5842 \times 10^5 \text{ 吨}$$

则前者表示有 9 位有效数字，绝对误差限为半吨；后者表示有 4 位有效数字，绝对误差限为 5 万吨．显然两者都是不符合实际情况的．

在科学计算中，常常采用科学记数法，即数的浮点表示法．例如把

$$25.01760 \quad \text{记成} \quad 0.2501760 \times 10^2$$

$$0.00013579 \quad \text{记成} \quad 0.13579 \times 10^{-3}$$

这时，0.2501760 是前者的有效数，0.13579 是后者的有效数．

2. 算法与数值稳定性

算法（Algorithm）这个概念可以从不同角度加以描述．一种最简单的说法是：算法是解决问题的一个逻辑顺序．稍为详细的说法是：算法是由一套明确的规则组成的若干步骤，它指定操作的顺序，将问题按一定数目的步骤加以解决．在数值计算领域中，我们所指的算法是：可供计算机执行的、对一些数据按某种规定的顺序进行运算的一个有穷序列．

以求一元二次方程

$$ax^2 + bx + c = 0 \qquad (a \neq 0)$$

的两个实根为例．我们知道，当 $b^2 - 4ac \geqslant 0$ 时，两个实根存在，计算公式为

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1.4}$$

由此，我们可以设计计算机算法如下：

算法 A

【步 1】输入 a, b, c；

【步 2】判断：如果 $b^2 - 4ac \geqslant 0$，则转步 4，否则执行下一步；

【步 3】输出无实根标志，转步 6；

【步 4】按公式(1.4)计算 x_1 与 x_2；

【步 5】输出 x_1 与 x_2；

【步 6】结束．

一般地，一个算法具有下列重要特性[c10]：

1） 输入，具有 0 个或 0 个以上由外界提供的量；

2） 输出，具有 1 个或多个输出；

3） 确定性，即算法的每一步都有确切、无二义性的定义；

4） 有穷性，即每个算法总是在执行有穷步之后结束；

5） 可行性，即算法中所有有待实现的运算都是基本的，原则上可由一个人仅用笔和纸就能完成的．

算法也可借助框图(或称流程图)来表示. 上述算法A的框图如图1-2.

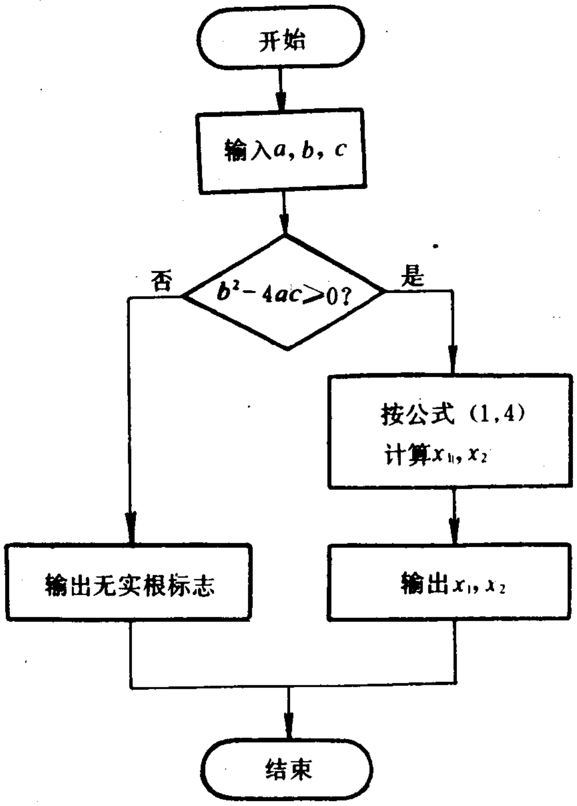
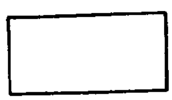
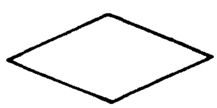


图 1-2

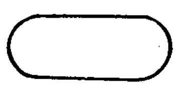
框图中各种符号的意义如下:



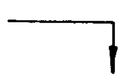
表示一般的计算处理过程，称工作框。



表示判断或检查。



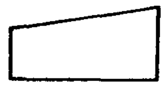
表示入口或出口，包括开始、结束、暂停和返回。



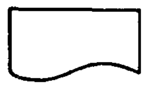
表示流程的路线及方向

在较详细的框图中，有时还采用如下符号：

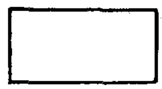
表示输入或输出。



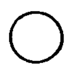
表示从键盘输入。



表示打印机输出。



表示过程的调用。



表示流程线连接点。

算法框图是直观地描述算法的一种工具.

与算法有关的一个重要问题是，初始数据的误差或计算中产生的舍入误差，在计算过程中的传播和积累，常因算法而异. 因而，同一个数值计算问题，采用不同算法得到的结果，其精度可能差别很大. 这就是算法的数值稳定性问题. 一个算法，如果计算结果受上述的误差影响较小，就称这个算法是**数值稳定的**，否则，就称这个算法是**数值不稳定的**.

例如，对上述二次方程，考虑当 $a = 1, b = -1.000000001 \times 10^9, c = 10^9$ 这种比较特殊的情形. 由方程

$$\begin{aligned} ax^2 + bx + c &= x^2 - 1.000000001 \times 10^9 x + 10^9 \\ &= x^2 - (10^9 + 1)x + 10^9 \\ &= (x - 10^9)(x - 1) = 0 \end{aligned}$$

可知这时两个实根的准确值为 $x_1 = 10^9$ 与 $x_2 = 1$. 但如果按上述算法 A（即直接用求根公式计算）编写程序并上机计算，比如在 APPLE II 微型计算机上计算，则得

$$x_1 = 10^9, \qquad x_2 = 0.703125$$

可见 x_1 很好，x_2 很差. 这说明上述算法 A 是数值不稳定的. 它受计算中产生的舍入误差影响.

在电子计算机上求解二次方程（其中令 $a > 0$）的一种做法是，先用下列公式算出绝对值大的一个根，比如 x_1

$$x_1 = \begin{cases} -\dfrac{b}{2a} + \sqrt{\left(\dfrac{b}{2a}\right)^2 - \dfrac{c}{a}}, & \text{若} -\dfrac{b}{2a} > 0 \qquad (1.5) \\ -\dfrac{b}{2a} - \sqrt{\left(\dfrac{b}{2a}\right)^2 - \dfrac{c}{a}}, & \text{若} -\dfrac{b}{2a} \leqslant 0 \qquad (1.6) \end{cases}$$

然后根据根与系数的关系 $x_1x_2=\frac{c}{a}=10^9$，计算另一根

$$x_2=10^9/x_1 \tag{1.7}$$

即重新设计算法如下：

算法 B

【步 1】输入 a,b,c；

【步 2】判断：如果 $\left(\frac{b}{2a}\right)^2-\frac{c}{a}\geqslant 0$，则转步 4，否则执行下一步；

【步 3】输出无实根标志，转步 9；

【步 4】如果 $-\frac{b}{2a}>0$，则转步 6，否则执行下一步；

【步 5】按公式(1.6)计算 x_1，转步 7；

【步 6】按公式(1.5)计算 x_1；

【步 7】按公式(1.7)计算 x_2；

【步 8】输出 x_1,x_2；

【步 9】结束.

按这个算法编写程序并上机计算，同样在 APPLE II 上计算，则得

$$x_1=10^9,\qquad x_2=1$$

可见 x_1 与 x_2 都很好，说明算法 B 有较好的数值稳定性.

也可以举出受初始数据的误差影响的不稳定算法(例如习题一的第 11 题).

3. 条件问题与病态概念

什么是数值计算中的条件问题？我们以计算函数值 $f(x)(x\in[a,b])$为例来说明. 假定运算时 x 代以近似值 $x^*\in[a,b]$，相应地函数值 $f(x)$ 代以 $f(x^*)$；又假定 x^* 的绝对误差很小，因而高阶量可以略去不计，于是我们可得

$$e^*[f(x)]=f(x)-f(x^*)=f'(x^*)e^*(x)$$

这时，若 $|f'(x)|\leqslant 1\quad(x\in[a,b])$，则有

$$|e^*[f(x)]|\leqslant|e^*(x)|$$

即自变量的微小变动引起的函数值的变动更微小. 如果对某一点 $\bar{x}\in[a,b]$，使 $|f'(\bar{x})|$ 的值很大，就说函数 $f(x)$ 在 $\bar{x}$ 这一点的计算在绝对误差意义下是坏条件的.

相应地，若 $x\neq 0$，x^* 充分接近 x，由 $e_r^*(x)=\frac{e^*(x)}{x}=\frac{x-x^*}{x}$，$e_r^*[f(x)]=\frac{e^*[f(x)]}{f(x)}=\frac{f(x)-f(x^*)}{f(x)}$，有

$$\begin{aligned}|e_r^*[f(x)]|&=\left|\frac{e^*[f(x)]}{f(x)}\cdot\frac{x}{e^*(x)}\cdot e_r^*(x)\right|\\&\approx\left|\frac{xf'(x)}{f(x)}\right||e_r^*(x)|\end{aligned}$$