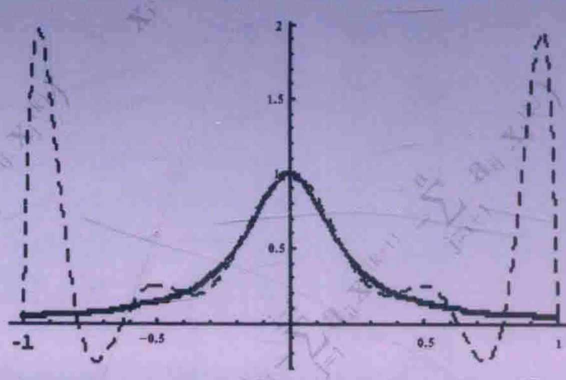
计算方法简明教程

李开宁 编



南京航空航天大学

计算方法简明教程

李开宁　编

南京航空航天大学
1999.8.

绪　论

1. 计算方法的意义和特点

数值计算是科学研究与工程技术中经常遇到的基本问题，但在如下情形采用传统的数学方法却无法获得所需要的解。

(1)所涉及的数学模型无系统的理论解法。例如，定积分 $\int_0^1 \sqrt{1+\cos^2 x}\,dx$ 无初等意义下的解的表达式，因此可积但积不出来。又如，对于多项式方程 $P_n(x)=0$ 的求根问题，理论上已证明当 $P_n(x)$ 为 n 次实系数多项式时，它在复数域内有且仅有 n 个根，且它若有复根必为共轭复根，但对于一般的高次多项式没有通用的求根公式。

(2)所涉及的数学模型有一套完整的理论解法，但其解法只适用于规模较小的情形，尤其当计算过程中出现误差时，将会导致误差严重积累淹没真解，以致方法完全失效。例如，解 n 阶线性方程组 $Ax=b$ 的克莱姆(Cram)法则 $x_j=A_j/|A|(j=1,\cdots,n)$ 的计算涉及 $(n+1)!$ 次乘除，当 n 增大时，计算量迅速增加，以致不堪承受，而且该方法易造成舍入误差的严重积累导致解失真。一般来说，使用计算机解 n 阶线性方程组时所能承受的计算量为 n^p 量级。

(3)基于离散数据建立数学模型时，常规的解析方法已不再适用，必须采用数值分析的方法。

实际上，除很少一类简单问题外，大多数源于工程实际的问题均涉及规模较大的复杂模型，需要借助于计算机才能进行数值求解。计算方法(又称为数值分析)的任务就是研究如何对给定的问题构建只涉及有限步四则运算的计算模型，以便有效地借助于计算机迅速求出所需要的数值解。这种计算模型通常又称为计算格式。

除纯粹的实验研究外，当今人们进行科学研究通常经过如下过程：

分析整理所确定的问题→构建数学模型→建立计算模型→进行程序设计(或仿真计算的设计)→利用计算机解出数据进行分析。

由此可见，计算方法是科学研究中不可或缺的一个重要工具。

计算方法是一门应用性很强的研究分支，其内容涉及面十分广泛，各种数值方法因依附于相应的数学分支而表现得相对独立，如数值代数是基于对代数学的数值计算研究，数值微积分是基于对微积分的数值计算研究，微分方程数值解是基于对微分方程的数值计算研究等等。但作为一门独立的研究分支，它自有贯穿于其中的精髓，这就是数学模型离散化所要遵循的相容性原则、控制误差积累的数值稳定性要求，以及反映计算格式效率的计算复杂性因素(存贮量，计算量(收敛速度)和为了适应大型计算所要求的并行性等)。希望读者在学习时应从这些方面深刻理解把握所建立的各种计算格式。

作为派生于纯粹数学的研究分支，在计算方法中理论的严谨分析固然必不可少，它为评价数值计算公式提供了可靠的保证，但作为一门应用性很强的分支，实际应用中往往不可能满足理论分析所要求的苛刻条件，因此常常是基于理论分析的结论而代之以近似的实用准则，并以数值试验辅证之。这自然会导致在特殊情况下计算的失败，因此需注意切莫把近似原则当作普适原则来随意使用。

2. 计算格式的相容性与稳定性

为便于读者更好地理解后面各章所介绍的计算格式，在此对计算格式的相容性与稳定性概念作一简要陈述。

定义 如果一个计算格式在取某种极限后可还原成某数学模型，则称该计算格式与此数学模型相容。

由此定义易证计算格式$\dfrac{y(a+h)-y(a)}{h}=2a$ 与数学模型 $y'=2x$ 相容。

证：因为

$$\frac{y(a+h)-y(a)}{h} \to y'(a) \ (h \to 0 \text{ 时})$$

从而当 $h\to 0$ 时，该计算格式还原为 $y'=2x$ 的特例 $y'(a)=2a$，所以结论成立。

又对于计算格式 $S_n=\sum\limits_{k=0}^{n}\dfrac{1}{k!}x^k$，我们有

$$S_n \to \sum_{k=0}^{\infty}\frac{1}{k!}x^k \quad (n \to \infty \text{ 时})$$

故 S_n 与数学模型 $e^x=\sum\limits_{k=0}^{\infty}\dfrac{1}{k!}x^k$ 相容，用 S_n 可近似计算 e^x。

相容性保证了计算格式的理论解充分接近精确解。在本教材中各种基于数学模型离散化而构建的计算格式均满足相容性。

定义 如果在用某一计算格式进行数值计算的过程中，误差不会严重积累，从而保证解满足所要求的精确度(简称精度)，则称该计算格式数值稳定，反之则称为不稳定。

为了简便，对于计算格式的稳定性分析通常基于对初始误差传播状况的讨论。

例 试建立计算定积分 $I_n=\int_0^1 x^n e^{x-1}\mathrm{d}x$ 的稳定的计算格式。

解：分部积分得

$$I_n = x^n e^{x-1}\big|_0^1 - n\int_0^1 x^{n-1}e^{x-1}\mathrm{d}x = 1 - nI_{n-1}$$

$$I_0 = \int_0^1 e^{x-1}\mathrm{d}x = 1 - e^{-1},\ I_1 = 1 - I_0 = e^{-1}$$

显然，我们有

(1)对任何 n，$I_n>0$；

又由

$$\begin{cases} I_n < \int_0^1 x^n e^{1-1}\mathrm{d}x = \dfrac{1}{n+1} \\ I_{n-1} = \dfrac{1-I_n}{n} > \dfrac{1}{n}(1-\dfrac{1}{n+1}) = \dfrac{1}{n+1} \\ I_n > \int_0^1 x^n e^{0-1}\mathrm{d}x = \dfrac{e^{-1}}{n+1} \end{cases}$$

得

(2) $\dfrac{e^{-1}}{n+1} < I_n < \dfrac{1}{n+1} < I_{n-1}$；

利用性质(2)易证

(3)$I_n \downarrow 0$。

因此我们可建立如下两种计算格式。

格式(A)：$I_n = 1 - nI_{n-1}(n=1,2,\cdots)$

格式(B)：$I_{n-1} = \frac{1}{n}(1-I_n)\ (n=N,N-1,\cdots)$

为分析其稳定性，我们记$\tilde{I}_n$为在初始值有误差时计算得到的近似值，记 $e_n = I_n - \tilde{I}_n$为误差，则由格式(A)得到 $e_n = -ne_{n-1}$，故$|e_n| = n!\ |e_0|$，随着 n 增加，误差迅速增长，计算不稳定；而由格式(B)得到 $e_{n-1} = -\frac{1}{n}e_n$，故 $|e_n| = \frac{1}{n+1}\frac{1}{n+2}\cdots\frac{1}{N}|e_N|$，随着计算的推进，误差逐渐减小，计算稳定。

实际上，若取$\tilde{I}_0 = 40.6321$，初始误差 $|e_0| \approx \frac{1}{2}10^{-4}$，用格式(A)计算得到$\tilde{I}_8 = -0.728$，不满足性质(1)；而当取 $N=9$，$\tilde{I}_9 = \frac{1}{2}(\frac{1}{10}+\frac{e^{-1}}{10}) \approx 0.0684$ 时，用格式(B)计算得到的$\tilde{I}_8$，$\tilde{I}_7,\cdots,\tilde{I}_0$均大于零。

一般来说，若一个计算格式满足如下误差关系式(以 e 表示误差)

$$|e_{后}| \leqslant C|e_{初}|(C\ 为常数)$$

则认为该计算格式数值稳定。

注意到

|精确解－近似解|≈|精确解－计算格式的理论解|＋|近似解－计算格式的理论解|

可知，计算格式的稳定性和相容性是一个计算格式可行的必要条件。

练习

给定计算格式 $y_{n+1} = (1-20h)y_n$ （$h = x_{n+1} - x_n$，y_n 为 $y(x_n)$的近似值）

(1)问：它与何种数学模型相容？

(2)确定使该计算格式稳定的 h 的取值范围。

目　录

第1章　非线性方程的解法

数学、物理和工程实际中提出的问题常常归结为求解非线性函数方程

$$f(x)=0$$

其解 x^* 称为该方程的根或 $f(x)$的零点。

常见的非线性函数方程为：

代数方程，即 $f(x)$为代数多项式。例如：

$$x^{100}-x^7+8x^4+90=0$$

对于 n 阶实系数代数方程，虽然理论上已经确证其在复数域内有且仅有 n 个根，但却难以直接求解。

更为复杂的非线性方程为超越方程，即 $f(x)$中含有超越函数(非代数多项式的解析函数)，如指数函数、对数函数、三角函数等，这时方程解的个数、性态等更加难以直接确定或求解。

由于对绝大多数非线性函数方程的解难以直接求出简洁的解析表达式，因此必需寻求数值求其近似解的方法。

非线性函数方程的数值计算方法主要分为两大类。

第一类是区间收缩法。其方法是首先确定初始含根区间$[a_0,b_0]$，例如从某个点 x_0 出发以 h 为步长依次考察点 $x_i=x_0+ih(i=1,2,\cdots)$，直至搜索出一个含根区间$[x_{i-1},x_i]$为止，这时$[a_0,b_0]=[x_{i-1},x_i]$即为所要选择的区间。然后根据某种原则构造区间序列$\{[a_k,b_k]\mid x^*\in[a_k,b_k],k=1,2,\cdots\}$，使得

$$[a_0,b_0]\supset\cdots\supset[a_k,b_k]\supset\cdots\cdots，且\lim_{k\to\infty}(b_k-a_k)=0$$

从而当 k 充分大，以至 $b_k-a_k<\varepsilon$(ε 为精度要求)时，便可取$[a_k,b_k]$中的点作为 x^* 的近似解。对这类方法本章仅介绍常用的二分法。

第二类是基于点逼近思想的迭代法。其方法是首先选定 x^* 的若干个近似值 $x_0,x_{-1},\cdots x_{-m}$，然后按某种原则计算出点序列$\{x_k\mid k=1,2,\cdots\}$，使得 $\lim\limits_{k\to\infty}x_k=x^*$，这样当 k 充分大时 x_k 便可作为解 x^* 的近似值，对这类方法本章主要介绍简单迭代法、牛顿迭代法和弦截法。

1.1　二分法

二分法又称为对分法，其基本假设是在给定的区间$[a,b]$上，$f(x)$连续且 $f(a)\cdot f(b)<0$。从而由闭区间上连续函数的性质知：存在 $x^*\in[a,b]$，使 $f(x^*)=0$。

1.1.1　二分法的计算步骤

二分法的计算步骤是首先选取$[a_0,b_0]=[a,b]$，然后对 $k=0,1,2,\cdots\cdots$依次计算

$$x_{k+1}=\frac{1}{2}(a_k+b_k)\qquad(区间中点)$$

$$
若\ f(a_k)\cdot f(x_{k+1})\begin{cases} >0 & 取[a_{k+1},b_{b+1}]=[x_{k+1},b_k] \\ <0 & 取[a_{k+1},b_{k+1}]=[a_k,x_{k+1}] \\ =0 & 终止计算 \end{cases}
$$

这样，除碰巧出现上面第三种情况而得到解 x^* 外，便构造出长度逐渐减半的区间序列 $\{[a_k,b_k],k=1,2,\cdots\}$。

实际计算时所采用的终止原则为：对于给定的精度要求 $|\bar{x}-x^*|<\varepsilon$（以下不加说明，即以 $\bar{x}$ 表示近似值），当计算到 $b_k-a_k<\varepsilon$ 时，便终止计算，取 $\bar{x}=a_k$ 或 b_k；或当计算到 $b_k-a_k<2\varepsilon$时，终止计算，取 $\bar{x}=\frac{1}{2}(a_k+b_k)$ 。

1.1.2 二分法的收敛性与事前误差估计

由区间收缩过程易知，$b_k-a_k=\frac{1}{2}(b_{k-1}-a_{k-1})=\cdots=\frac{1}{2^k}(b-a)\to 0(k\to\infty 时)$，所以二分法总是收敛的。若按上述终止原则的第二种方法取近似值 $\bar{x}=\frac{1}{2}(a_k+b_k)$，我们有

$$
|\bar{x}-x^*|\leqslant\frac{1}{2}(b_k-a_k)=\frac{1}{2^{k+1}}(b-a)
$$

故对给定的精度要求 $|\bar{x}-x^*|<\varepsilon$，可从 $\frac{1}{2^{k+1}}(b-a)<\varepsilon$，预先估计出所需迭代次数 $K>\frac{\lg\left(\frac{b-a}{\varepsilon}\right)}{\lg 2}-1$，即须计算 $K=\left[\frac{\lg\frac{b-a}{\varepsilon}}{\lg 2}\right]$ 步，这里[]为取整符号（例如[2.5]=2），特别当 $\varepsilon=10^{-m}$时（m 为正整数）

$$
K=\left[\frac{\lg(b-a)+m}{\lg 2}\right] \tag{1.1.1}
$$

例 1.1 试用二分法求 $f(x)=x^3-2x-5=0$ 的一个正根，使误差小于 10^{-3}。

解：取 $x_0=0$，步长 $h=1$，向右搜索得到 $f(0)=-5,f(1)=-6,f(2)=-1,f(3)=16$，故可取初始区间$[a_0,b_0]=[2,3]$，且由(1.1.1)式预先算出

$$
K=\left[\frac{\lg(3-2)+3}{\lg 2}\right]=[9.97]=9
$$

所以 $\bar{x}=\frac{1}{2}(b_9+a_9)$ 即为所要求的近似值。

计算过程见下表：

k	a_k	b_k	x_k	$f(x_k)$的符号
0	2	3	2.5	+
1	2	2.5	2.25	+
2	2	2.25	2.125	+
3	2	2.125	2.0625	−
4	2.0625	2.125	2.0938	−
5	2.0938	2.125	2.0194	+
6	2.0938	2.1094	2.1016	+
7	2.0938	2.0016	2.0977	+
8	2.0938	2.0977	2.0958	+
9	2.0938	2.0958	2.0949	

所以 $\bar{x}=2.095$。

1.1.3 二分法评述

二分法的优点是方法简单可靠，易于编程实现，它对函数要求低，工程实际中提出的问题绝大多数都可满足连续性要求，二分法适用于 x^* 为 $f(x)=0$ 的奇数重根情形。

但是当 x^* 为偶数重根时，对任何充分小的 $\delta_1,\delta_2>0$，$f(x^*-\delta_1)f(x^*+\delta)>0$，所以无法直接用二分法求根。这时如果 $f(x)$ 可微，则可根据 $f(x)=0$ 的根为 $F(x)=\dfrac{f(x)}{f'(x)}=0$ 的单根而变通地对 $F(x)=0$ 用二分法求根。

二分法不能用于求复根，也难以向方程组推广使用，比如对二阶非线性方程组，若采用二分法，则须对矩形区域分割、收缩，不仅计算量大大增加，而且问题的讨论也变得更加复杂。

二分法的收敛速度比较慢，本质上是线性收敛的。

1.2 一般迭代法

1.2.1 迭代法的算法思想

对于给定的非线性方程

$$f(x)=0 \tag{1.2.1}$$

迭代法的算法思想为：

(1)把方程(1.2.1)等价变换为如下形式

$$x=g(x)$$

从而 $x^*=g(x^*)$，我们称 x^* 为函数 $g(x)$ 的不动点。

(2)建立迭代格式

$$x_{k+1}=g(x_k) \tag{1.2.2}$$

或更一般地，建立格式

$$x_{k+1}=g(x_k,x_{k-1},\cdots x_{k-m}) \quad (m\geqslant 1) \tag{1.2.2$'$}$$

格式(1.2.2)称为单步迭代法，格式(1.2.2)$'$称为多步迭代法，它们统称为不动点迭代法，$g(x)$称为迭代格式的迭代函数。

(3)选取初始值 x_0(或更一般地，选取 $x_0,x_{-1},\cdots,x_{-m}$)，用迭代格式递推计算直至得到满意的解。

迭代法的关键问题是如何构造迭代格式以保证迭代得到的解 $x_k\to x^*$($k\to\infty$时)。

1.2.2 迭代法的收敛性

本章仅限于对单步迭代法讨论收敛性，为此，我们先对迭代法作一几何解释。

求解方程 $x=g(x)$ 在几何上即为求直线 $y=x$ 与曲线 $y=g(x)$ 的交点的横坐标 x^*。对于 x^* 的初始近似值 x_0，取点 $P_0(x_0,g(x_0))$，其纵坐标即为 $x_1=g(x_0)$，过 P_0 引平行于 x 轴的直线交 $y=x$ 于 Q_1，再过 Q_1 作垂直于 x 轴的直线交 $y=g(x)$ 于 P_1，其横坐标即为 x_1，如此继续做下去便得到点序列 $P_2,P_3,\cdots$，它们相应的横坐标即为 $x_2,x_3,\cdots\cdots$(见图 1-1)。

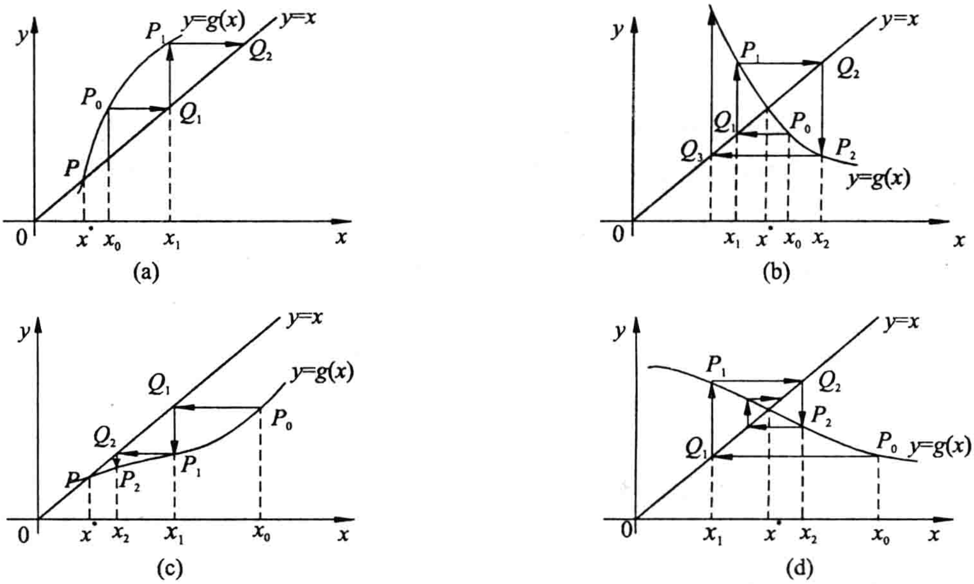


图 1-1

由图 1-1 可见，当 P_k 与 P 的纵坐标之差 $|g(x^*)-g(x_k)|\geqslant|x^*-x_k|$（$P_k$ 与 P 的横坐标之差）时，即图 1-1 中(a)、(b)情形，迭代格式不收敛；反之，在(c)、(d)情形迭代格式收敛。实际上，进一步分析可见，当 $|g(x^*)-g(x_k)|\geqslant|x^*-x_k|$（$k=0,1,2,\cdots$）时，有

$$|x^*-x_k|\geqslant|x^*-x_0|$$

格式不收敛；而若 $|g(x^*)-g(x_k)|\leqslant L|x^*-x_k|$ （$0<L<1,\ k=0,1,2,\cdots$），则

$$|x^*-x_k|\leqslant L^k|x^*-x_0|\to 0(k\to\infty\text{ 时})$$

格式收敛。

由此我们引进如下李普希兹连续的定义，进而建立收敛性定理。

定义 1.1 设在某个区间 Δ 内，函数 $g(x)$ 满足下述李普希兹(Lipschitz)条件：

$$|g(x)-g(y)|\leqslant L|x-y| \quad (\forall\ x,y\in\Delta,\ 0<L<1\text{ 为常数})$$

则称 $g(x)$ 在 Δ 内李普希兹连续。

命题 1.1 若 $g'(x)$ 在区间 Δ 内连续且 $|g'(x)|\leqslant L<1(\forall\ x\in\Delta)$，则 $g(x)$ 在 Δ 内李普希兹连续。

定理 1.1 设 $x^*=g(x^*)$，$g(x)$ 在区间 Δ：$|x-x^*|<\delta$ 内李普希兹连续，则对任何初值 $x_0\in\Delta$，由迭代格式 $x_{k+1}=g(x_k)$ 计算得到的解序列 $\{x_k\}$ 收敛于 x^*（这时我们称迭代格式 $x_{k+1}=g(x_k)$ 在 x^* 的邻域 Δ 上局部收敛）。

证：(1)首先用数学归纳法证明：对 $\forall\ k\in N=\{0,1,2,\cdots\}$，$x_k\in\Delta$。

由假设知 $x_0\in\Delta$。

又设 $x_k\in\Delta$，则 $|x^*-x_k|<\delta$，所以

$$|x^*-x_{k+1}|=|g(x^*)-g(x_k)|\leqslant L|x^*-x_k|<x^*-x_k<\delta$$

即 $x_{k+1}\in\Delta$。

综上，由归纳法原理知，结论成立。

(2)由(1)的结论和 $g(x)$ 在 Δ 内李普希兹连续的假设,可递推得到

$$|x^* - x_k| \leqslant L|x^* - x_{k-1}| \leqslant \cdots \leqslant L^k|x^* - x_0| \to 0(k \to \infty \text{ 时})$$

因此,$\lim\limits_{k\to\infty} x_k = x^*$,定理得证。

注: 1° $g(x)$ 在 Δ 内李普希兹连续的条件保证了 x^* 为 $f(x)=0$ 在 Δ 内的唯一根。

证:反设存在 $\tilde{x}\in\Delta,\tilde{x}\not\approx x^*$ 且 $f(\tilde{x})=0$,即 $\tilde{x}=g(\tilde{x})$,则

$$0 < |\tilde{x} - x^*| = g|(\tilde{x}) - g(x^*)| \leqslant L|\tilde{x} - x^*| < |\tilde{x} - x^*|$$

矛盾,所以结论成立。

2° 迭代函数在 x^* 附近李普希兹连续从而收敛的迭代格式统称为皮卡(Picard)迭代。

推论 设 $x^* = g(x^*)$,若 $g(x)$ 在 x^* 附近连续可微且 $|g(x^*)|<1$,则迭代格式 $x_{k+1}=g(x_k)$ 在 x^* 附近局部收敛。

证:因为 $g(x)$ 在 x^* 附近连续可微,所以对 $\varepsilon_0=\frac{1}{2}(1-|g'(x^*)|)$,存在 $\delta>0$,使得

$$|g'(x) - g(x^*)| < \varepsilon_0(\text{当 } x \in \Delta = \{x||x - x^*| < \delta\} \text{ 时})$$

因此 $$|g'(x)| \leqslant |g'(x^*)| + \varepsilon_0 = \frac{1}{2}(1 + |g'(x^*)|) \xlongequal{\text{记为}} L < 1$$

由命题 1.1 得:$g(x)$ 在 Δ 内李普希兹连续。再由定理 1.1 便得:对$\forall$ $x_0\in\Delta$,解序列$\{x_k\}$收敛于 x^*,证毕。

注:由于 x^* 事先未知,故实际应用时,代之以近似判则 $|g'(x_0)|<1$,这仅当 x_0 充分接近 x^* 时才可保证迭代格式的收敛性。

定理 1.2 (非局部收敛定理)如果 $g(x)$ 在$[a,b]$上连续可微且以下条件满足:

(1)若 $x\in[a,b]$,则 $g(x)\in[a,b]$;

(2)$\forall$ $x\in[a,b]$,$|g'(x)|\leqslant L<1$。

那么,对任何 $x_0\in[a,b]$,$x_{k+1}=g(x_k)$的解序列$\{x_k\}$收敛于 x^*。

证:只须证明存在解 $x^*\in[a,b]$。

实际上,记 $\varphi(x)=x-g(x)$,则由条件(1)得 $f(a)\leqslant 0,f(b)\geqslant 0$。若 $f(a)=0$,则 $x^*=a$;若 $f(b)=0$,则 $x^*=b$;若 $f(a)\neq 0,f(b)\neq 0$,则 $f(a)f(b)<0$,故有 $x^*\in(a,b)$。所以总有 $x^*\in[a,b]$,证毕。

注:虽然定理 1.1 的条件是充分条件,但其条件并不很强,实际上,我们易证如下命题。

命题 1.2 若在区间$[a,b]$内$|g'(x)|\geqslant 1$,则对$\forall$ $x_0\in[a,b]$,迭代格式 $x_{k+1}=g(x_k)$不收敛。

1.2.3 迭代法的误差估计

对于 Picard 迭代,我们由李普希兹连续条件得:

$$|x_{k+1} - x_k| \leqslant L|x_k - x_{k-1}| \qquad (k = 1,2,\cdots)$$

故对正整数 p,有 $|x_{k+p}-x_k|\leqslant|x_{k+p}-x_{k+p-1}|+|x_{k+p-1}-x_{k+p-2}|+\cdots+|x_{k+1}-x_k|$

$$\leqslant(L^p+L^{p-1}+\cdots+L)|x_k-x_{k-1}|=\frac{L}{1-L}(1-L^p)|x_k-x_{k-1}|$$

令 $p\to\infty$ 取极限得 $$|x^* - x_k| \leqslant \frac{L}{1-L}|x_k - x_{k-1}| \tag{1.2.3}$$

进一步有 $$|x^* - x_k| \leqslant \frac{L^k}{1-L}|x_1 - x_0| \tag{1.2.4}$$

由此，我们可进行

1)事前误差估计。欲要求$|\bar{x}-x^*|<\varepsilon$，则由(1.2.4)知，只须

$$\frac{L^k}{1-L}|x_1-x_0|<\varepsilon$$

据此解得
$$k>\frac{1}{|\ln L|}\left(\ln\frac{|x_1-x_0|}{1-L}-\ln\varepsilon\right)$$

故所需迭代步数的上限为 $K_0=\left[\frac{1}{|\ln L|}\left(\ln\frac{|x_1-x_0|}{1-L}-\ln\varepsilon\right)\right]+1$

2)事先误差估计。根据式(1.2.3)，我们可选用$\frac{L}{1-L}|x_k-x_{k-1}|<\varepsilon$作为终止原则。但由于一般难以准确确定$L$，故实际计算时常简单地选用$|x_k-x_{k-1}|<\varepsilon$作为终止原则，当然这在$L<\frac{1}{2}$时$x_k$是不能满足精度要求的，故有时人们在没有把握时，凭经验(或通过参考解比较)设定一个常数$\alpha<1$(如取$\alpha=\frac{1}{2^n}$)以$\alpha|x_k-x_{k-1}|<\varepsilon$作为终止原则。

例 1.2 试建立迭代格式求解$f(x)=x^3-2x-5=0$在区间(2,3)内满足精度要求$\varepsilon=10^{-8}$的根。

解：(1)若把$f(x)=0$等价化为$x=\frac{x^3+5}{2}$，可建立迭代格式$x_{k+1}=\tilde{g}(x_k)$，$\tilde{g}(x)=\frac{x^3+5}{2}$，由$\tilde{g}'(x)=\frac{3}{2}x^3$知，在[2,3]内$|\tilde{g}(x)|>1$，所以该迭代格式在[2,3]内不收敛。

(2)若把$f(x)=0$等价化为

$$x=\sqrt[3]{2x+5}$$

建立迭代格式

$$x_{k+1}=g(x_k),\ g(x)=\sqrt[3]{2x+5}$$

易知在$x>0$时$g(x_k)$单调增，故$g(2)<g(x)<g(3)$，而$g(2)=\sqrt[3]{9}>2$，$g(3)=\sqrt[3]{11}<3$，所以在[2,3]内定理1.2的条件(1)满足；又由$g'(x)=\frac{2}{3}\frac{1}{\sqrt[3]{(2x+5)^2}}>0$在[2,3]内单调减得知，在[2,3]内$0<g'(x)<g'(2)=\frac{2}{9\sqrt[3]{3}}<1$。故由定理1.2得：任取$x_0\in[2,3]$，该迭代格式收敛。

取$x_0=2$计算得

k	x_k	k	x_k
1	2.080083823	6	2.094550838
2	2.092350678	7	2.094551303
3	2.094216996	8	2.094551454
4	2.094500652	9	2.094551477
5	2.094543758	10	2.094551481

由$|x_{10}-x_9|<10^{-8}$知，x_{10}即为所要求的解。

1.2.4 迭代法的收敛速度与加速收敛技巧

收敛性是一个数值计算方法可行的必要条件，而对方法优劣性的评价则主要依据其计算工作量。收敛速度即为相关的一个重要概念。

定义 1.2 设迭代格式 $x_{k+1}=g(x_k)$的解序列$\{x_k\}$收敛于 $x=g(x)$的根 x^*，如果迭代误差 $e_k=x_k-x^*$ 当 $k\to\infty$时满足渐近关系式

$$\frac{e_{k+1}}{e_k^p}\to C(C\neq 0\ \text{为常数})$$

则称该迭代格式是 p 阶收敛的。特别地，$p=1$ 时称为线性收敛，$p>1$ 时称为超线性收敛，$p=2$时称为平方收敛。

线性收敛的迭代格式是最慢的，为了提高收敛速度，我们以下介绍埃特肯(Aitken)加速技巧。

设序列$\{x_k\}$线性收敛于 x^*，即有 $\lim\limits_{k\to\infty}\frac{e_{k+1}}{e_k}=C$，则近似地有

$$\begin{cases}x_1-x^*\approx C(x_0-x^*)\\ x_2-x^*\approx C(x_1-x^*)\end{cases}$$

两式相除得
$$\frac{x_1-x^*}{x_2-x^*}\approx\frac{x_0-x^*}{x_1-x^*}$$

解得
$$x^*-\left[x_2-\frac{(x_2-x_1)^2}{x_2-2x_1+x_0}\right]\approx 0$$

据此，我们可取修正值 $\widetilde{x}_2=x_2-\dfrac{(x_2-x_1)^2}{x_2-2x_1+x_0}$ 作为 x^* 的新近似值以提高精度，这一技巧称为埃特肯加速技巧。

如果 $0<C<1$，用埃特肯加速技巧可使解序列平方收敛于 x^*。把埃特肯加速技巧应用于单步迭代法 $x_{k+1}=g(x_k)$便构成了如下 Steffensen 算法。

算法 1.1 (Steffensen 算法)

(1)输入精度要求 ε，初值 x_0，最大迭代步数 N 及稳定性要求 $\tilde{\varepsilon}$；

(2)令 $k=0$；

(3)计算 $\alpha=g(x_k)$，$\beta=g(\alpha)$；

(4)计算 $\gamma=\beta-2\alpha+x_k$，若$|\gamma|<\tilde{\varepsilon}$，则终止计算，输出 β 作为近似值；否则转入(4)；

(5)计算 $x_{k+1}=\beta-\dfrac{(\beta-\alpha)^2}{\gamma}$ (加速处理)；

(6)若$|x_{k+1}-x_k|<\varepsilon$，则终止计算，输出 x_{k+1}作为近似值；

(7)令 $k=k+1$，若 $k>N$，则停止计算，输出失败信息(这时须另选初值计算)，否则转入(3)。

需要注意的是，埃特肯加速技巧对超线性收敛的解序列并无大的改善。

例 1.3 对例 1.2 的迭代格式 $x_{k+1}=\sqrt[3]{2x_k+5}$ 取 $x_0=2$ 用 Steffensen 算法计算得

k	x_k	α	β
0	2 X_0	2.080083823 X_1	2.092350678 X_2
1	2.094577839 X_1	2.094555487	2.094552090
2	2.094551481	2.094551481	2.094551481
3	2.094551481		

由 $|x_3-x_2|<10^{-8}$ 知，x_3 即为所要求的解。与例 1.2 比较可见，Steffensen 算法把原来迭代格式的收敛速度大大提高了。

1.3 牛顿迭代法

1.3.1 牛顿迭代公式的构造

基于线性近似的思想，牛顿建立了一种著名的单步迭代法。下面具体介绍其格式的构造思想。

设 $f(x)$ 在其零点 x^* 附近连续可微，已知 x_k 为 x^* 的第 k 次近似值，则

$$\begin{aligned} f(x) &= f(x_k) + f'(x_k)(x - x_k) + O((x - x_k)^2) \\ &\approx f(x_k) + f'(x_k)(x - x_k) \xlongequal{\text{记为}} P_1(x) \end{aligned}$$

取 $P_1(x)$ 的根作为 x^* 的第 $k+1$ 次近似值，则从 $f(x)+f'(x_k)(x_{k+1}-x_k)=0$ 解出

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

这就是著名的牛顿迭代公式。其几何意义就是过点 $P(x_k, f(x_k))$ 作函数 $y=f(x)$ 的切线 l：

$$y = f(x_k) + f'(x_k)(x - x_k)$$

以切线 l 与 x 轴的交点 x_{k+1} 作为 x^* 的新近似值（见图 1-2）。

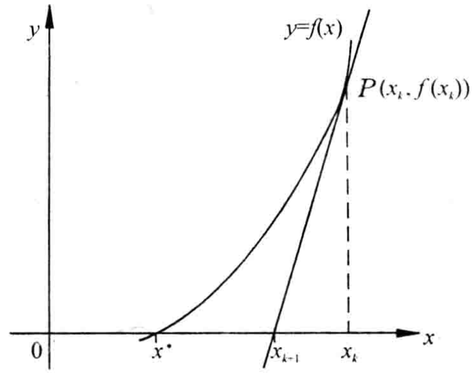


图 1-2

算法 1.2 （牛顿迭代法）

(1)输入迭代初值 x_0，精度要求 ε，最大迭代步数 L（通常取 $L=60$）；

(2)计算 $F=f(x_0)$，$DY=f'(x_0)$；

(3)若 $|DY|+1.0=1.0$ 则令 $L=0$，输出失败信息，终止计算；

(4)计算 $x_1 = x_0 - \dfrac{F}{DY}$；

(5) 令 $F = f(x_1)$，$DY = f'(x_1)$；

(6)若 $|x_1-x_0|<\varepsilon$ 或 $|F|<\varepsilon$ 转入(8)，否则令 $L=L-1$，$x_0=x_1$；

(7)若 $L\neq 0$ 转入(3)；

(8)打印 x_1。

1.3.2 牛顿迭代法的收敛性与收敛速度

基于1.2的讨论，我们易得如下定理。

定理1.3 给定$f(x)=0$，如果$f(x)$在根x^*附近二阶连续，且x^*为$f(x)=0$的单根，则牛顿迭代法在x^*附近至少是平方收敛的。

证：由牛顿迭代法的迭代函数表达式$g(x)=x-\dfrac{f(x)}{f'(x)}$得

$$g'(x)=\frac{f(x)f''(x)}{[f'(x)]^2}$$

而单根条件保证了$f'(x^*)\neq 0$，所以$g'(x^*)=0$。由定理1.1的推论知，牛顿迭代法在x^*附近局部收敛，当x_k处于x^*附近$f''(x)$连续的区域内时可作泰勒展式

$$0=f(x^*)=f(x_k)+f'(x_k)(x^*-x_k)+\frac{1}{2}f''(\xi)(x^*-x_k)^2 \quad (\xi\text{介于}x^*,x_k\text{之间})$$

整理得

$$\begin{aligned} x^* &= x_k-\frac{f(x_k)}{f'(x_k)}-\frac{1}{2}\frac{f''(\xi)}{f'(x_k)}(x^*-x_k)^2 \\ &= x_{k+1}-\frac{1}{2}\frac{f''(\xi)}{f'(x_k)}(x^*-x_k)^2 \end{aligned}$$

所以

$$\frac{e_{k+1}}{e_k^2}=\frac{1}{2}\frac{f''(\xi)}{f'(x_k)}$$

由$f'(x^*)\neq 0$便得

$$\lim_{k\to\infty}\frac{e_{k+1}}{e_k^2}=\frac{1}{2}\frac{f''(x^*)}{f'(x^*)}\text{为一有限常数}$$

因此，当$f''(x^*)\neq 0$时，牛顿迭代法为平方收敛；当$f''(x^*)=0$时，牛顿迭代法超平方收敛。

例1.4 对例1.2取$x_0=2$用牛顿迭代公式计算得

k	x_k
1	2.1
2	2.094568121
3	2.094551482
4	2.094551482

可见，只须迭代四步便得到满足要求$|x_k-x_{k-1}|<10^{-8}$的结果。

1.3.3 牛顿迭代法评述

牛顿迭代法的突出优点是收敛速度比较快，但它一般是局部收敛的方法，对初始值的要求高，这是迭代法通常存在的弊病。为解决这个问题，我们常常选用1.1所介绍的二分法来提供足够“好”的近似值作为初值，然后用牛顿迭代法来加快收敛过程。此外，可通过增加“下山”限制来放宽对初值的要求，也就是把牛顿迭代法修改为

$$\begin{cases} x_{k+1}=x_k-\lambda_k\dfrac{f(x_k)}{f'(x_k)} \\ x_0 \end{cases}$$

其中λ_k的选取使得$|f(x_{k+1})|<|f(x_k)|$（“下山”限制）。这种方法称为牛顿下山法，详见[1]。

另需注意的是，当 x^* 为 $f(x)=0$ 的 $m(\geqslant 2)$重根时，牛顿迭代法仅仅线性收敛，这时如果 $f(x)$三阶连续，则可变通地对 $F(x)=\dfrac{f(x)}{f'(x)}=0$ 用牛顿迭代法求根。

牛顿迭代法由于涉及 $f'(x)$的计算，导致了对函数的要求高，并增加了每一迭代步的计算量，这在一定程度上减弱了该迭代法收敛快的优越性，而且在向非线性方程组推广时，$[f'(x)]^{-1}$演变为对一导数矩阵 $DF=(\dfrac{\partial f_i}{\partial x_j})_{n\times n}$的求逆过程而使计算量和对函数的要求大大增加。因此，人们致力于研究建立牛顿迭代法的修改格式以回避对函数导数值的计算，由此派生出很多有效的算法。本章仅对非线性方程介绍一种较为有效的修改算法——弦截法。

1.4 弦截法

弦截法又称为割线法，其计算思想是：若已知 x^* 的两个近似值 x_k 和 x_{k-1}，则以 $f(x)$在 x_{k-1}与 x_k 之间的平均变化率（差商）$\dfrac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$ 近似代替 $f'(x_k)$，把牛顿迭代法修改为

$$\begin{cases} x_{k+1}=x_k-\dfrac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1}) \\ x_0,x_{-1} \end{cases} \qquad (k=0,1,\cdots)$$

这就是弦截法。它是一个多步迭代法，其几何意义是以过 $P(x_k,f(x_k))$和 $Q(x_{k-1},f(x_{k-1}))$两点作曲线 $y=f(x)$的弦线 l：

$$y=f(x_k)+\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}(x-x_k)$$

以 l 与 x 轴的交点作为 x^* 的新近似值 x_{k+1}（参见图 1-3）。

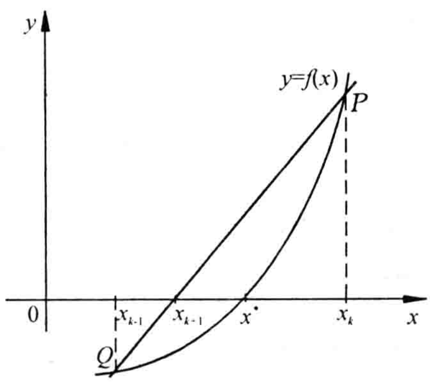


图 1-3

关于弦截法的收敛性质我们有如下定理。

定理 4.1 设 $f(x)$在根 x^* 的邻域 $\Delta=\{x\,|\,|x-x^*|<\delta\}$内二阶连续，且对任何 $x\in\Delta$，$f'(x)\neq 0$，则对任何 $x_0,x_{-1}\in\Delta$，相应的弦截法是 $p=\dfrac{1+\sqrt{5}}{2}\approx 1.618$ 阶收敛的。

该定理说明弦截法是超线性收敛的的算法，也是局部收敛的方法，其初始值亦可用二分法提供。