

# 计算方法复习提纲

网安 1903 史圣久

2020 年 12 月 19 日

学号：U201915584

# 目录

<b>1</b>	<b>引论</b>	<b>4</b>
1.1	计算方法的主要内容	4
1.2	常用近似计算方法	4
1.2.1	离散化方法	4
1.2.2	递推化方法	5
1.2.3	近似替代法	5
1.3	误差的基本概念	6
1.3.1	误差的来源	6
1.3.2	误差、误差限、相对误差和有效数字	6
1.3.3	有效数字与相对误差限的关系	7
1.3.4	算术运算的误差及误差限	7
1.4	近似计算注意原则	8
<b>2</b>	<b>插值方法与曲线拟合</b>	<b>9</b>
2.1	插值多项式的存在唯一性	9
2.2	Lagrange 插值	9
2.2.1	线形插值	9
2.2.2	抛物插值	10
2.2.3	Lagrange 插值公式	10
2.2.4	插值余项	11
2.3	Newton 插值	11
2.3.1	基函数	11
2.3.2	差商的概念	12
2.3.3	差商的性质	13
2.3.4	Newton 插值公式	13
2.4	Hermite 插值	14
2.5	分段插值	15
2.5.1	高次插值 Runge 现象	15
2.5.2	分段插值概念	15

2.5.3	线性分段插值 . . . . .	15
2.5.4	分段三次 Hermite 插值 . . . . .	16
<b>3</b>	<b>数值积分</b>	<b>17</b>
3.1	机械求积公式和代数精度 . . . . .	17
3.1.1	求积公式 . . . . .	17
3.1.2	代数精度 . . . . .	18
3.2	求积公式的构造方法 . . . . .	18
3.2.1	用解代数方程组的方法构造求积公式 . . . . .	18
3.2.2	用插值方法构造求积公式 . . . . .	18
3.3	Newton-Cotes 求积公式 . . . . .	19
3.3.1	Newton-Cotes 公式 . . . . .	19
3.3.2	Newton-Cotes 公式的数值稳定性 . . . . .	19
3.3.3	Newton-Cotes 求积公式截断误差 . . . . .	20
3.4	复化求积法 . . . . .	20
3.4.1	复化求积公式 . . . . .	20
3.4.2	复化求积公式截断误差 . . . . .	21
3.5	Romberg 求积公式及算法 . . . . .	21
3.5.1	梯形法的递推化 . . . . .	21
3.5.2	Romberg 公式 . . . . .	21
<b>4</b>	<b>常微分方程数值解法</b>	<b>23</b>
4.1	尤拉法、隐式尤拉法和二步尤拉法 . . . . .	23
4.1.1	尤拉法 . . . . .	23
4.1.2	隐式尤拉法 . . . . .	23
4.1.3	二步欧拉法 . . . . .	24
4.1.4	局部截断误差与精度 . . . . .	24
4.2	改进的尤拉法 . . . . .	24
4.2.1	梯形公式 . . . . .	24
4.2.2	改进的尤拉法 . . . . .	24
4.3	Runge-Kutta 法 . . . . .	25
4.3.1	Runge-Kutta 基本思想 . . . . .	25
4.3.2	二阶 Runge-Kutta 方法 . . . . .	25
4.3.3	高阶 Runge-Kutta 法 . . . . .	26
4.3.4	变步长 Runge-Kutta 法 . . . . .	27
4.4	收敛性和稳定性 . . . . .	27
4.4.1	收敛性 . . . . .	27
4.4.2	稳定性 . . . . .	28

<b>5</b>	<b>方程求根</b>	<b>29</b>
5.1	根的隔离与二分法	29
5.1.1	根的隔离	29
5.1.2	二分法	29
5.2	迭代法及其收敛性	30
5.2.1	迭代法的基本概念	30
5.2.2	迭代法过程的收敛性	30
5.3	收敛速度及收敛过程的加速	31
5.3.1	迭代的收敛速度	31
5.3.2	收敛过程的加速	32
5.4	牛顿法	32
5.4.1	牛顿法的构造及牛顿迭代公式	32
5.4.2	牛顿法的收敛性和收敛速度	33
5.4.3	初始值的选取	33
5.4.4	牛顿法下山法	33

# Chapter 1

## 引论

### 1.1 计算方法的主要内容

1. 面向计算机，根据计算机的特点为计算机提供可行有效的算法，使其可以直接处理。
2. 有可靠的理论分析，可以任意逼近达到精度要求，对误差进行分析，建立在数理理论上。
3. 有比较好的计算复杂度，即算法的时间复杂度和空间复杂度尽可能低。
4. 有数值实验，满足前三者后通过数值实验进行验证。

### 1.2 常用近似计算方法

#### 1.2.1 离散化方法

把连续型问题转化为离散型问题，因为计算机无法对连续型问题进行求解，为了使计算机可以正常运算，所以需要将连续型问题转化为离散型问题。最为常见的例子就是求定积分了。

例 1.1: 计算定积分

$$I = \int_a^b f(x) dx$$

将区间  $[a, b]$   $n$  等分,  $a = x_0 < x_1 < \dots < x_n = b, x_i = x_0 + h_i; i = 1, 2, \dots, n$ ; 则对于每个小区间  $[x_i, x_{i+1}]$ ,  $S_i$  我们可以用近似替代

$$S_i \approx \frac{h}{2}[f(x_{i-1}) + f(x_i)]$$

$$S = \sum_{i=1}^n S_i \approx \frac{h}{2} \sum_{i=1}^n [f(x_{i-1}) + f(x_i)] = \frac{h}{2}[f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b)]$$

$$\int_a^b f(x) dx \approx \frac{h}{2}[f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b)]$$

### 1.2.2 递推化方法

具体思路就是，将一项复杂的运算归纳为若干次简单的计算，且递推化的算法更加适合计算机进行运算。

例 1.2:  $p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

$$\begin{aligned} p_n(x) &= a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \\ &= (a_n x^{n-1} + a_{n-1} x^{n-2} + \dots + a_2 x + a_1) x + a_0 \\ &= ((a_n x^{n-2} + a_{n-1} x^{n-3} + \dots + a_3 x + a_2) x + a_1) x + a_0 \\ &= (\dots ((a_n x + a_{n-1}) x + a_{n-2}) x + \dots + a_1) x + a_0 \end{aligned}$$

其中，可以令：

$$\begin{aligned} v_0 &= a_n \\ v_1 &= a_n x + a_{n-1} \\ v_2 &= v_1 x + a_{n-2} \\ v_3 &= v_2 x + a_{n-3} \\ &\vdots \\ v_n &= v_{n-1} x + a_0 \end{aligned}$$

显然  $v_n = p_n$

### 1.2.3 近似替代法

许多的数学问题在有限次的运算里是无法求出解的，但可以化为在有限次运算可以得到有一定误差的解。

例 1.3: 求 e 的近似值：

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots \\ e &= 1 + 1 + \frac{1}{2!} + \dots + \frac{1}{n!} + \dots \\ e &\approx 1 + 1 + \frac{1}{2!} + \dots + \frac{1}{n!} \end{aligned} \tag{1.1}$$

使用余项进行误差估计

$$R_n(x) = e^x - \left(1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}\right) = \frac{f^{n+1}(\zeta)}{(n+1)!}, \zeta \in (0, 1) \quad (1.2)$$

$$|R_n(1)| \leq \frac{e}{(n+1)!}$$

## 1.3 误差的基本概念

### 1.3.1 误差的来源

1. 模型误差，建立数学模型时的理想化误差。
2. 观测误差，观测结果与实际大小的误差。
3. 截断误差，理论精确值需要无数次运算，实际只使用了有限次运算所造成的误差。
4. 舍入误差，保留有限长的近似值时四舍五入造成的误差。

### 1.3.2 误差、误差限、相对误差和有效数字

假设某一量的精确值为  $x$ ，其近似值为  $x^*$

【定义 1.1】 绝对误差：

$$\varepsilon(x) = |x - x^*|$$

存在一个数  $\eta$  满足

$$\varepsilon(x) = |x - x^*| \leq \eta$$

称  $\eta$  为近似值  $x^*$  的绝对误差限：

$$x = x^* \pm \eta$$

【定义 1.2】 相对误差：

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

或：

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

【定义 1.3】 相对误差限：

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

### 1.3.3 有效数字与相对误差限的关系

【定义 1.4】 若近似值  $x^*$  的绝对误差限小于等于某一位的半个单位，则称  $x^*$  精确到该位。

$$x^* = \pm 0.a_1 a_2 \dots a_n \times 10^m$$

$$|x^* - x| \leq \frac{1}{2} \times 10^{m-l}, 1 \leq l \leq n$$

则称  $x^*$  有  $l$  位有效数字。

【定理 1.1】 设近似值  $x^* = \pm 0.a_1 a_2 \dots a_n \times 10^m$ ，共有  $n$  位有效数字， $a_1 \neq 0$ ，则其相对误差限为  $\frac{1}{2a_1} \times 10^{-n+1}$ 。

【定理 1.2】 设近似值  $x^* = \pm 0.a_1 a_2 \dots a_n \times 10^m$ ，其相对误差限为  $\frac{1}{2(a_1+1)} \times 10^{-n+1}$ ， $a_1 \neq 0$ ，则其有  $n$  位有效数字。

### 1.3.4 算术运算的误差及误差限

和的误差是误差之和，差的误差是误差之差。

$$(x^* \pm y^*) - (x \pm y) = (x^* - x) \pm (y^* - y)$$

因为

$$|(x^* \pm y^*) - (x \pm y)| \leq |x^* - x| + |y^* - y|$$

所以，误差限之和是和或差的误差限。

乘法误差

$$|x^* y^* - xy| \leq |x^*| \eta(y^*) + |y^*| \eta(x^*) + \eta(x^*) \eta(y^*)$$

忽略  $\eta(x^*) \eta(y^*)$

$$|x^* y^* - xy| \leq |x^*| \eta(y^*) + |y^*| \eta(x^*)$$

除法误差

$$\begin{aligned} \left| \frac{x^*}{y^*} - \frac{x}{y} \right| &= \left| \frac{x^* y^* - x^* \varepsilon(y^*) - x^* y^* + y^* \varepsilon(x^*)}{y^* (y^* - \varepsilon(y^*))} \right| \\ &\leq \frac{|x^*| \eta(y^*) + |y^*| \eta(x^*)}{|y^*|^2 \left( 1 - \frac{\eta(y^*)}{|y^*|} \right)} \end{aligned}$$



因为  $\frac{\eta(y^*)}{|y^*|}$  可忽略, 所以:

$$\left| \frac{x^*}{y^*} - \frac{x}{y} \right| \leq \frac{|x^*|\eta(y^*) + |y^*|\eta(x^*)}{|y^*|^2}$$

综上所述

$$\begin{aligned}\eta(x^* + y^*) &= \eta(x^*) + \eta(y^*) \\ \eta(x^* - y^2) &= \eta(x^*) + \eta(y^*) \\ \eta(x^* y^*) &\approx |x^*|\eta(y^*) + |y^*|\eta(x^*) \\ \eta\left(\frac{x^*}{y^*}\right) &\approx \frac{|x^*|\eta(y^*) + |y^*|\eta(x^*)}{|y^*|^2} \quad y \neq 0, y^* \neq 0\end{aligned}$$

## 1.4 近似计算注意原则

### 1. 加减运算

近似数进行加减运算时, 把小数中位数多的先四舍五入, 使其比小数位数最少的数多一位, 计算结果保留小数位数与原数位数最少的数相同。

### 2. 乘除运算

近似数相乘除时, 各因子保留位数应比有效数字位数最少的多一位, 所得结果的有效数位数与原近似值中有效数字位数最少的位数至多一位。

### 3. 乘方与开方运算

近似数进行乘方或开方运算时, 原有几位有效数字, 结果就保留几位有效数字。

### 4. 对数运算

进行对数运算时, 所取对数的位数应与其真数有效数字位数相等。

# Chapter 2

## 插值方法与曲线拟合

### 2.1 插值多项式的存在唯一性

所谓多项式插值问题就是要确定一个次数不高于的代数多项式

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (2.1)$$

使其满足

$$p_n(x_i) = y_i \quad i = 0, 1, 2, \dots, n \quad (2.2)$$

点  $x_i (i = 0, 1, 2, \dots, n)$  互异, 称为插值节点, 包含插值节点的区间  $a, b$  称为插值区间。

**【定理 2.1】** 在  $n+1$  个互异的插值节点  $x_0, x_1, x_2, \dots, x_n$  上满足条件式 (2.2) 的次数不高于  $n$  的代数多项式 (2.1) 存在且惟一。

### 2.2 Lagrange 插值

#### 2.2.1 线形插值

**【问题 2.1】** 设函数  $y = f(x)$  在给定的互异节点  $x_0, x_1$  上的函数值为  $y_0, y_1$  求作一个次数小于等于 1 的多项式

$$p_1(x) = a_0 + a_1x$$

使它满足:

$$p_1(x_0) = y_0 \quad p_1(x_1) = y_1$$

问题 2.1 所定义的插值问题称为线性插值。

其解为:

$$p_1(x) = y_0 l_0(x) + y_1 l_1(x) \quad (2.3)$$

其中:

$$l_0(x) = \frac{x - x_1}{x_0 - x_1} \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}$$

公式 (2.3) 也可以简单表示为:

$$p_1(x) = \frac{x - x_1}{x_0 - x_1} l_0(x) + \frac{x - x_0}{x_1 - x_0} l_1(x) \quad (2.4)$$

## 2.2.2 抛物插值

**【问题 2.2】** 设函数  $y = f(x)$  在给定的互异节点  $x_0, x_1, x_2$  上的函数值分别为  $y_0, y_1, y_2$  求作一个次数小于等于 2 的多项式。

$$p_2(x) = a_0 + a_1 x + a_2 x^2$$

使它满足:

$$p_2(x_0) = y_0, \quad p_2(x_1) = y_1, \quad p_2(x_2) = y_2$$

问题 2.2 所定义的插值问题称为抛物插值。

其解为:

$$p_2(x) = y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x) \quad (2.5)$$

其中:

$$\begin{aligned} l_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \\ l_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \\ l_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

## 2.2.3 Lagrange 插值公式

**【问题 2.3】** 设函数  $y = f(x)$  在给定的互异节点  $x_0, x_1, x_2, \dots, x_n$  上的函数值分别为  $y_0, y_1, y_2, \dots, y_n$  求作一个次数小于等于  $n$  的多项式。

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

使它满足:

$$p_n(x_i) = y_i, \quad i = 0, 1, 2, \dots, n$$

问题 2.2 所定义的插值问题称为抛物插值。

其解为:

$$p_n(x) = \sum_{i=0}^n y_i l_i(x) \quad (2.6)$$

其中:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}$$

公式(2.6)也可以简单表示为:

$$p_n(x) = \sum_{i=0}^n y_i \left( \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)} \right) \quad (2.7)$$

## 2.2.4 插值余项

依函数  $f(x)$  构造出的插值多项式  $p_n(x)$  仅为一种近似表示式。

令:

$$R_n(x) = f(x) - p_n(x) \quad (2.8)$$

并称  $R_n(x)$  为  $f(x)$  的截断误差或插值余项。

**【定理 2.2】** 设区间  $[a, b]$  含有节点  $x_0, x_1, \dots, x_n$ , 而  $f(x)$  在  $[a, b]$  内有  $1 \sim n+1$  阶的导数, 且  $f(x_i) = y_i (i = 0, 1, \dots, n)$  已给, 则当  $x \in [a, b]$  时, 对于式 (2.8) 给出的  $p_n(x)$  满足

$$f(x) - p_n(x) = \frac{f^{n+1}(u)}{(n+1)!} \prod_{i=0}^n (x - x_i) \quad (2.9)$$

## 2.3 Newton 插值

### 2.3.1 基函数

**【问题 2.4】** 求作  $n$  次多项式

$$\begin{aligned} N_n(x) = & c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots \\ & + c_n(x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned}$$

使满足

$$N_n(x_i) = f(x_i) \quad i = 0, 1, 2, \dots, n$$

问题 2.4 所定义的插值问题称为 **Newton 插值**。为了使  $N_n(x)$  的形式得到简化, 引进如下记号:

$$\begin{cases} \varphi_0(x) &= 1 \\ \varphi_i(x) &= (x - x_{i-1})\varphi_{i-1}(x) \\ &= (x - x_0)(x - x_1) \cdots (x - x_{i-1}) \quad i = 0, 1, 2, \dots, n \end{cases} \quad (2.10)$$

【定义 2.1】 由式(2.10)定义的  $n + 1$  个多项式

$$\varphi_0(x), \varphi_1(x), \varphi_2(x), \dots, \varphi_n(x)$$

于是  $N_n(x)$  可以表示为如下形式

$$N_n(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_n\varphi_n(x)$$

当  $x = x_0$  得

$$c_0 = N_n(x_0) = f(x_0)$$

当  $x = x_1$  得

$$c_1 = \frac{N_n(x_1) - c_0}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

当  $x = x_2$  得

$$c_2 = \left[ \frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right] / (x_2 - x_0)$$

### 2.3.2 差商的概念

给定  $[a, b]$  中互异点  $x_0, x_1, \dots$ , 以及函数  $f(x)$  对应得函数值  $f(x_0), f(x_1), \dots$ , 则称

$$f[x_0, x_1] = \frac{f(x_0) - f(x_1)}{x_0 - x_1}$$

为  $f(x)$  在  $x_0, x_1$  的一阶差商。

$$f[x_0, x_1, x_2] = \frac{f[x_0, x_1] - f[x_1, x_2]}{x_0 - x_2}$$

称为  $f(x)$  在  $x_0, x_1, x_2$  的二阶差商。

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_0, x_1, \dots, x_{k-1}] - f[x_1, x_2, \dots, x_k]}{x_0 - x_k}$$

称为  $f(x)$  在  $x_0, x_1, \dots, x_k$  的  $k$  阶差商。

### 2.3.3 差商的性质

(1)  $k$  阶差商是函数值  $f(x_0), f(x_0), f(x_0)$  的线性组合

$$f[x_0, x_1, \dots, x_k] = \sum_{j=0}^k \frac{f(x_j)}{(x_j - x_0)(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_k)}$$

令

$$w_k(x) = \prod_{i=0}^k (x - x_i)$$

则

$$w'_k(x) = \sum_{l=0}^k \prod_{\substack{i=0 \\ i \neq l}}^k (x - x_i)$$

$$w'_k(x_j) = \prod_{\substack{i=0 \\ i \neq j}}^k (x - x_i)$$

所以  $k$  阶差商又可以写成

$$f[x_0, x_1, \dots, x_k] = \sum_{j=0}^k \frac{f(x_j)}{w'_k(x_j)}$$

(2) 对称性: 在差商  $f[x_0, x_1, \dots, x_k]$  中, 可以改变结点的顺序而不改变差商的值, 差商的值和结点的排列顺序无关。

(3) 如果  $f[x, x_0, x_1, \dots, x_k]$  是  $x$  的  $m$  次多项式, 则  $f[x, x_0, x_1, \dots, x_{k+1}]$  是  $x$  的  $m-1$  次多项式。

(推论) 若  $f(x)$  是  $n$  次多项式, 则  $f[x, x_0, x_1, \dots, x_n]$  恒等于 0

### 2.3.4 Newton 插值公式

问题 2.4 的解为:

$$\begin{aligned} N_n(x) = & f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ & + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned} \quad (2.11)$$

其余项  $R_n(x)$  为:

$$\begin{aligned} R_n(x) = & f(x) - N_n(x) \\ = & f[x, x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_n) \end{aligned} \quad (2.12)$$

【定理 2.3】 若  $f(x)$  在  $[a, b]$  上有  $n+1$  阶导函数, 则存在一点  $\xi \in [\min x_i, \max x_i]$ , 使得:

$$f[x_0, x_1, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}$$

## 2.4 Hermite 插值

【问题 2.5】 设函数  $y = f(x)$  在节点  $x_0, x_1, \dots, x_n$  上的函数值  $f(x_i) = y_i$  及导数  $f'(x_i) = y'_i$  已知, 求作一个插值多项式函数  $H(x)$ , 使满足条件:

$$H(x_i) = y_i, \quad H'(x_i) = y'_i \quad i = 0, 1, \dots, n$$

在上式共给了  $H(x)$  应满足的  $2n+2$  个条件, 故可以确定一个次数不超过  $2n+1$  次的多项式:

$$H(x) = H_{2n+1}(x) = a_0 + a_1x + \dots + a_{2n+1}x^{2n+1}$$

设基函数  $\alpha_i(x), \beta_i(x)$  满足如下调节:

$$\begin{cases} \alpha_i(x_j) = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \\ \alpha'_i(x_j) = 0 \\ \beta_i(x_j) = 0 \\ \beta'_i(x_j) = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \end{cases}$$

可以解得:

$$H_{2n+1}(x) = \sum_{i=0}^n \left[ y_i \left( 1 - 2(x - x_i) \sum_{\substack{k=0 \\ k \neq i}}^n \frac{1}{x_i - x_k} \right) l_i^2(x) + y'_i(x - x_i) l_i^2(x) \right] \quad (2.13)$$

【定理 2.4】 设  $f(x)$  在区间  $[a, b]$  内有  $2n+2$  阶导函数, 且  $f(x_i) = y_i, f'(x_i) = y'_i$  已给, 则下式成立:

$$R_{2n+1}(x) = f(x) - H_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \omega_{n+1}^2(x) \quad (2.14)$$

其中

$$\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i)$$

【定理 2.5】 满足条件的插值多项式  $H_{2n+1}(x)$  是唯一的。

三次 Hermite 插值  $H_3(x)$  如下:

$$H_3(x) = y_i \varphi_0\left(\frac{x-x_i}{h_i}\right) + y_{i+1} \varphi_1\left(\frac{x-x_i}{h_i}\right) + h_i y'_i \Psi_0\left(\frac{x-x_i}{h_i}\right) + h_i y'_{i+1} \Psi_1\left(\frac{x-x_i}{h_i}\right) \quad (2.15)$$

其中

$$\begin{cases} x_i \leq x \leq x_{i+1} \\ \varphi_0(x) = (x-x_{i+1})^2(2x-x_i-3x_{i+1}), & \varphi_1(x) = (x-x_i)^2(2x-x_{i+1}-3x_i) \\ \Psi_0(x) = x(x-x_{i+1}), & \Psi_1(x) = x(x-x_i) \\ h_i = x_{i+1} - x_i \end{cases}$$

## 2.5 分段插值

### 2.5.1 高次插值 Runge 现象

在大范围内使用高次插值, 效果往往不理想

### 2.5.2 分段插值概念

随着插值结点的增多, 插值多项式的次数必然也会随之而增加, 而高次插值多项式会产生 Runge 现象, 所以应该尽力避免高次插值多项式。

可以采取分段插值的方式, 将一个大区间分成若干个小区间, 而在每一个小区间  $[x_i, x_{i+1}]$  构造一个插值多项式  $p_i(x) (i = 0, 1, \dots, n-1)$ 。

### 2.5.3 线性分段插值

【问题 2.6】 在划分  $\Delta$  的每个结点  $x_i$  上给出了相应的  $y_i$ , 求一次多项式  $s_1(x)$  使得:

$$s_1(x) = y_i \quad x = x_i, x_{i+1}$$

此问题的解为:

$$s_1(x) = y_i \frac{x-x_{i+1}}{x_i-x_{i+1}} + y_{i+1} \frac{x-x_i}{x_{i+1}-x_i} \quad x_i \leq x \leq x_{i+1} \quad (2.16)$$

【定理 2.6】 若  $f(x)$  有 2 阶导数, 且是问题 2.6 的解, 则有:

$$|f(x) - s_1(x)| \leq \frac{1}{8} h_i^2 \max_{x_i \leq x \leq x_{i+1}} |f''(x)| \quad (2.17)$$



### 2.5.4 分段三次 Hermite 插值

【问题 2.6】 在划分  $\Delta$  的每个结点  $x_i$  上给出了相应的函数值  $y_i$  和导数值  $y'_i$ ，求三次多项式  $s_3(x)$  使得：

$$\begin{aligned} s_3(x_i) &= y_i & s_3(x_{i+1}) &= y_{i+1} \\ s'_3(x_i) &= y'_i & s'_3(x_{i+1}) &= y'_{i+1} \end{aligned}$$

此问题的解为：

$$s_3(x) = y_i \varphi_0\left(\frac{x-x_i}{h_i}\right) + y_{i+1} \varphi_1\left(\frac{x-x_i}{h_i}\right) + h_i y'_i \Psi_0\left(\frac{x-x_i}{h_i}\right) + h_i y'_{i+1} \Psi_1\left(\frac{x-x_i}{h_i}\right) \quad (2.18)$$

其中

$$\begin{cases} x_i \leq x \leq x_{i+1} \\ \varphi_0(x) = (x-1)^2(2x+1), & \varphi_1(x) = x^2(-2x+3) \\ \Psi_0(x) = x(x-1)^2, & \Psi_1(x) = x^2(x-1) \\ h_i = x_{i+1} - x_i \end{cases}$$

【定理 2.7】 若  $f(x)$  有 4 阶导数，且是问题 2.7 的解，则有：

$$|f(x) - s_3(x)| \leq \frac{h^4}{384} \max_{x_i \leq x \leq x_{i+1}} |f^{(4)}(x)| \quad h = \max h_i \quad (2.19)$$

# Chapter 3

## 数值积分

### 3.1 机械求积公式和代数精度

#### 3.1.1 求积公式

1. 梯形公式

$$\int_a^b f(x)dx \approx \frac{1}{2}(b-a)[f(a) + f(b)]$$

2. 中矩形公式

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

3. Simpson 公式

$$\int_a^b f(x)dx \approx \frac{1}{6}(b-a)\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

其一般形式如下所示

$$\int_a^b f(x)dx \approx (b-a)[c_0f(x_0) + c_1f(x_1) + \dots + c_nf(x_n)]$$

其中

$$c_0 + c_1 + \dots + c_n = 1$$

即

$$\int_a^b f(x)dx \approx \sum_{i=0}^n A_i f(x_i) \quad (3.1)$$

求积公式(3.1)也可写成如下形式

$$\int_a^b f(x)dx \approx \sum_{i=0}^n A_i f(x_i) + R \quad (3.2)$$

其中  $R$  为余项。

### 3.1.2 代数精度

【定义 3.1】 如果求积公式(3.1) 对于一切次数小于等于  $m$  的多项式是准确的，且对于  $m+1$  次多项式不准确，则称该公式有  $m$  次代数精度。

【定理 3.1】 对于给定的  $n+1$  个互异结点

$$a = x_0 \leq x_1 \leq \cdots \leq x_n = b$$

总存在系数  $A_0, A_1, \dots, A_n$ ，使得求积公式(3.1)的代数精度最低为  $n$ 。

## 3.2 求积公式的构造方法

### 3.2.1 用解代数方程组的方法构造求积公式

$$\begin{cases} A_0 + A_1 + \dots + A_n = b - a \\ A_0 x_0 + A_1 x_1 + \dots + A_n x_n = \frac{1}{2} (b^2 - a^2) \\ \vdots \\ A_0 x_0^n + A_1 x_1^n + \dots + A_n x_n^n = \frac{1}{n+1} (b^{n+1} - a^{n+1}) \end{cases} \quad (3.3)$$

### 3.2.2 用插值方法构造求积公式

用  $p_n(x)$  代替  $f_n(x)$

$$\int_a^b f(x) dx \approx \int_a^b p_n(x) dx \quad (3.4)$$

$$\int_a^b p_n(x) dx = \int_a^b \left[ \sum_{i=0}^n f(x_i) l_i(x) \right] dx = \sum_{i=0}^n \int_a^b [f(x_i) l_i(x)] dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x) dx$$

所以其求积系数为：

$$A_i = \int_a^b l_i(x) dx$$

【定理 3.2】 形如式子(3.1)的求积公式至少具有  $n$  此代数精度的充要条件是它是插值型的。

### 3.3 Newton-Cotes 求积公式

#### 3.3.1 Newton-Cotes 公式

将积分区间  $[a, b]$   $n$  等分,  $h = (b - a)/n$  为各小区间长度,  $x_k = a + kh$  作为插值节点, 得到如下求积公式:

$$\int_a^b f(x)dx \approx (b - a) \sum_{k=0}^n C_k f(x_k) \quad (3.5)$$

称式(3.5)为 Newton-Cotes 公式, 其中  $C_k$  为 Cotes 系数。

$$C_k = \frac{1}{b - a} \int_a^b \prod_{\substack{j=0 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j} dx$$

$$C_k = \frac{(-1)^{n-k}}{n \cdot k!(n - k)!} \int_0^n \prod_{\substack{j=0 \\ j \neq k}}^n (t - j) dt \quad (3.6)$$

$n$  从 1  $\rightarrow$  4 的 Cotes 系数如下:

$n$	$C_0(n)$	$C_1(n)$	$C_2(n)$	$C_3(n)$	$C_4(n)$
1	1/2	1/2			
2	1/6	2/3	1/6		
3	1/8	3/8	3/8	1/8	
4	7/90	16/45	2/15	16/45	7/90

#### 3.3.2 Newton-Cotes 公式的数值稳定性

一个算法如果在计算的过程中舍入误差在一定的条件下能够得到控制, 即舍入误差的增长不影响产生可靠的结果, 则称它是数值稳定的。

误差

$$\varepsilon_n = -(b - a) \sum_{i=0}^n C_i \varepsilon_i$$

设  $\varepsilon = \max_{0 \leq k \leq n} |\varepsilon_k|$  则有:

$$|\varepsilon_n| \leq \varepsilon(b - a) \sum_{k=0}^n |C_k|$$

当  $1 \leq k \leq 8$  时,  $C_k$  都是正数, 有

$$|\varepsilon_n| \leq \varepsilon(b - a)$$

故当  $n$  小于 8 时, Newton-Cotes 公式有数值稳定性, 而高阶的 Newton-Cotes 公式不是数值稳定的。

### 3.3.3 Newton-Cotes 求积公式截断误差

$$\begin{aligned}
 R &= \int_a^b f(x)dx - \sum_{i=0}^k A_k f(x_k) = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x) dx \quad \xi \in [a, b] \\
 &= \frac{h^{n+2}}{(n+1)!} \int_0^n f^{(n+1)}(\xi) \prod_{j=0}^n (t-j) dt \quad \xi \in [a, b]
 \end{aligned} \tag{3.7}$$

【定理 3.3】 当  $n$  为偶数时， $n$  阶 Newton-Cotes 求积公式代的数精度至少为  $n+1$ 。  
几种低阶 Newton-Cotes 公式的截断误差

#### 1. 梯形公式截断误差

$$R = \frac{1}{2}(b-a)^3 \int_0^1 f''(\xi) t(t-1) dt = -\frac{1}{12}(b-a)^3 f''(\xi), \quad \xi \in [a, b]$$

#### 2. Simpson 公式截断误差

$$R = \frac{f^{(4)}(\eta)}{4!} \int_a^b (x-a) \left(x - \frac{a+b}{2}\right)^2 (x-b) dx = -\frac{1}{90} \left(\frac{b-a}{2}\right)^5 f^{(4)}(\eta) \quad \eta \in [a, b]$$

#### 3. Cotes 公式截断误差

$$\begin{aligned}
 \int_a^b f(x) &\approx \frac{1}{90}(b-a) \left[ 7f(a) + 32f(a+h) + 12f\left(\frac{a+b}{2}\right) + 32f(a+3h) + 7f(b) \right] \\
 R &= -\frac{8}{945} \left(\frac{a+b}{4}\right)^7 f^{(6)}(\eta) \quad \eta \in [a, b]
 \end{aligned}$$

## 3.4 复化求积法

### 3.4.1 复化求积公式

#### 1. 复化梯形公式

$$\begin{aligned}
 \int_{x_k}^{x_{k+1}} f(x) dx &\approx \frac{1}{2}(x_{k+1} - x_k) [f(x_k) + f(x_{k+1})] \\
 \int_a^b f(x) dx &= \sum_{i=0}^{n-1} \sum_{x_k}^{x_{k+1}} f(x) dx \approx \sum_{i=0}^{n-1} \frac{h}{2} [f(x_k) + f(x_{k+1})] \\
 &= \frac{h}{2} \left[ f(a) + 2 \sum_{k=1}^{n-1} f(x_k) + f(b) \right] = T_n
 \end{aligned} \tag{3.8}$$

#### 2. 复化 Simpson 公式

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[ f(a) + 4 \sum_{k=1}^m f(x_{2k-1}) + 2 \sum_{k=1}^{m-1} f(x_{2k}) + f(b) \right] = S_n \tag{3.9}$$

### 3. 复化 Cotes 公式

$$\int_a^b f(x)dx = \frac{1}{90} \times 4h \left[ 7f(a) + 32 \sum_{k=1}^m f(x_{4k-3}) + 12 \sum_{k=1}^m f(x_{4k-2}) + 32 \sum_{k=1}^m f(x_{4k-1}) + 14 \sum_{k=1}^{m-1} f(x_{4k}) + 7f(b) \right] = C_n \quad (3.10)$$

#### 3.4.2 复化求积公式截断误差

##### 1. 复化梯形公式截断误差

$$R_T \approx -\frac{1}{12}(b-a)h^2 f''(\eta) \quad \eta \in [a, b] \quad (3.11)$$

##### 2. 复化 Simpson 公式截断误差

$$R_S \approx -\frac{1}{180}(b-a) \left(\frac{h}{2}\right)^4 f^{(4)}(\eta) \quad \eta \in [a, b] \quad (3.12)$$

##### 3. 复化 Cotes 公式截断误差

$$R_C \approx -\frac{2}{945}(b-a) \left(\frac{h}{4}\right)^6 f^{(6)}(\eta) \quad \eta \in [a, b] \quad (3.13)$$

## 3.5 Romberg 求积公式及算法

### 3.5.1 梯形法的递推化

$$\begin{cases} T_{k1} = \frac{h}{2} [f(x_k) + f(x_{k+1})] \\ T_{k2} = \frac{h}{4} [f(x_k) + 2f(x_{k+\frac{1}{2}}) + f(x_{k+1})] \end{cases}$$

因而

$$T_{k2} = \frac{1}{2}T_{k1} + \frac{h}{2}f(x_{k+\frac{1}{2}})$$

由此可以一步一步二分得到更精确的结果。

### 3.5.2 Romberg 公式

梯形法的递推化速度过慢，用 Romberg 公式能大大加快速度。首先以梯形法的递推化得到的几组值作为  $T_m^{(0)}$

#### 复化 Simpson 的积分值

$$T_m^{(1)} : S_n = \frac{4}{3}T_{2n} - \frac{1}{3}T_n \quad (3.14)$$

### 复化 Cotes 的积分值

$$T_m^{(2)} : \quad C_n = \frac{16}{15}S_{2n} - \frac{1}{15}S_n \quad (3.15)$$

$\vdots$

$$T_m^{(3)} : \quad C_n = \frac{64}{63}S_{2n} - \frac{1}{63}S_n$$

### Romberg 积分法

$$\left\{ \begin{array}{l} T_0^{(0)} = \frac{b-a}{2}[f(a) + f(b)] \\ \text{对于 } m = 1, 2, \dots, \text{依次计算} \\ (1) \quad h_m = \frac{b-a}{2^m} \\ (2) \quad T_m^{(0)} = \frac{1}{2}T_{m-1}^{(0)} + h_m \sum_{i=1}^{2^{m-1}} f(a + (2i-1)h_m) \\ (3) \quad T_{m-j}^{(j)} = \frac{T_{m-j+1}^{(j-1)} - \left(\frac{1}{2}\right)^{(2j)} T_{m-j}^{(j-1)}}{1 - \left(\frac{1}{2}\right)^{2j}} = \frac{4T_{m-j+1}^{(j-1)} - T_{m-j}^{(j-1)}}{4-1} \quad j = 1, 2, \dots, m \end{array} \right. \quad (3.16)$$

# Chapter 4

## 常微分方程数值解法

重点：一阶方程的初值问题

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

### 4.1 尤拉法、隐式尤拉法和二步尤拉法

#### 4.1.1 尤拉法

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) \\ y_0 = y(x_0) \end{cases} \quad (4.1)$$

#### 4.1.2 隐式尤拉法

$$\begin{cases} y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \\ y_0 = y(x_0) \end{cases} \quad (4.2)$$

隐式尤拉法不能直接求解，可以采用下面两种方法进行实际计算：

##### 1. 隐式化显式

如同解方程一样，将未知数的  $y_{n+1}$  都移到一边，产生一个显式递推公式。适用于  $f(x, y)$  比较简单的情形。

##### 2. 预报-校正法

$$\begin{cases} \bar{y}_{n+1} = y_n + hf(x_n, y_n) \\ y_{n+1} = y_n + hf(x_{n+1}, \bar{y}_{n+1}) \\ y_0 = y(x_0) \end{cases}$$



### 4.1.3 二步欧拉法

$$\begin{cases} y_{n+1} = y_{n-1} + 2hf(x_n, y_n) \\ y_0 = y(x_0) \end{cases}$$

### 4.1.4 局部截断误差与精度

【定义 4.1】 对于求解一阶常微分方程初值问题的某以计算方法，若假定在求  $y_{n+1}$  的递推公式中等式右边的所有的量为精确的，在此前提下  $y(x_{n+1} - y_{n+1})$  称为此方法的局部截断误差。若局部截断误差为  $O(h^{p+1})$ ，则称此方法的精度为  $p$  阶。

欧拉法、隐式欧拉法精度为 1 阶，二步欧拉法精度为 2 阶。

## 4.2 改进的尤拉法

### 4.2.1 梯形公式

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \quad (4.3)$$

性质:

1. 梯形公式是欧拉法和隐性欧拉法的算术平均
2. 梯形公式精度为 2 阶

### 4.2.2 改进的尤拉法

1. 嵌套形式

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n))]$$

2. 平均化公式

$$\begin{cases} y_p = y_n + hf(x_n, y_n) \\ y_c = y_n + hf(x_{n+1}, y_p) \\ y_{n+1} = \frac{1}{2} (y_p + y_c) \end{cases}$$

以上为两种等价的形式，改进的尤拉法精度为 2 阶

## 4.3 Runge-Kutta 法

### 4.3.1 Runge-Kutta 基本思想

$$\begin{cases} y'(x) = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

利用泰勒展开:

$$y(x_{n+1}) = y(x_n) + hy'(\xi) \quad (x_n \leq \xi \leq x_{n+1})$$

实际计算中由于  $\xi$  无法准确计算, 可以利用  $[x_n, x_{n+1}]$  间的导数值来近似代替  $\xi$

### 4.3.2 二阶 Runge-Kutta 方法

$$\begin{cases} y_{n+1} = y_n + h((1-\lambda)k_1 + \lambda k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + ph, y_n + phk_1) \end{cases}$$

其中

$$\lambda p = \frac{1}{2}$$

两种常用的二阶 Runge-Kutta 法:

#### 1. 改进欧拉法

当取  $\lambda = \frac{1}{2}, p = 1$  时

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + h, y_n + hk_1) \end{cases}$$

#### 2. 变形的欧拉法

当取  $\lambda = 1, p = \frac{1}{2}$  时

$$\begin{cases} y_{n+1} = y_n + hk_2 \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \end{cases}$$

### 4.3.3 高阶 Runge-Kutta 法

#### 3 阶 Runge-Kutta 方程

$$\begin{cases} y_{n+1} = y_n + c_1 k_1 + c_2 k_2 + c_3 k_3 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + a_2 h, y_n + b_{21} k_1) \\ k_3 = hf(x_n + a_3 h, y_n + b_{31} k_1 + b_{32} k_2) \end{cases}$$

其中:

$$\begin{cases} c_1 + c_2 + c_3 = 1 \\ a_2 = b_{21} \\ a_3 = b_{31} + b_{32} \\ c_2 a_2 + c_3 a_3 = \frac{1}{2} \\ c_2 a_2^2 + c_3 a_3^2 = \frac{1}{3} \\ c_3 b_{32} a_2 = \frac{1}{6} \end{cases}$$

#### 常用 3 阶 Runge-Kutta 公式

库塔公式:

$$\begin{cases} y_{n+1} = y_n + \frac{1}{6} (k_1 + 4k_2 + k_3) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\ k_3 = hf(x_n + h, y_n - k_1 + 2k_2) \end{cases}$$

#### 常用 4 阶 Runge-Kutta 公式

经典公式:

$$\begin{cases} y_{n+1} = y_n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 = hf(x_n + h, y_n + k_3) \end{cases}$$

Gill 公式:

$$\begin{cases} y_{n+1} = y_n + \frac{1}{6} [k_1 + (2 - \sqrt{2})k_2 + (2 + \sqrt{2})k_3 + k_4] \\ k_1 = hf(x_n, y_n) \\ k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{\sqrt{2}-1}{2}k_1 + \frac{2-\sqrt{2}}{2}k_2\right) \\ k_4 = hf\left(x_n + h, y_n - \frac{\sqrt{2}}{2}k_2 + \frac{2+\sqrt{2}}{2}k_3\right) \end{cases}$$

#### 4.3.4 变步长 Runge-Kutta 法

取  $h$  为步长, 截断误差为:

$$y = y(x_{n+1}) - y_{n+1}^{(h)} \approx ch^{p+1}$$

之后取  $\frac{h}{2}$  为步长, 截断误差为:

$$y = y(x_{n+1}) - y_{n+1}^{(h/2)} = 2c\left(\frac{h}{2}\right)^{p+1}$$

由此可得到下列估计式:

$$y(x_{n+1}) - y_{n+1}^{(h/2)} \approx \frac{1}{2^p - 1} [y_{n+1}^{(h/2)} - y_{n+1}^{(h)}]$$

前后两次的偏差为:

$$\Delta = \frac{1}{2^p - 1} [y_{n+1}^{(\frac{h}{2})} - y_{n+1}^{(h)}]$$

1. 对于给定的  $\epsilon$ , 如果  $\Delta \geq \epsilon$ , 将步长折半反复进行计算, 直至  $\Delta \leq \epsilon$  为止, 这时取最终得到的  $y_{n+1}^{(h/2)}$  为结果。
2. 如果  $\Delta \leq \epsilon$ , 若将步长反复加倍, 直到  $\Delta \geq \epsilon$  为止, 这时再将步长折半一次, 就得到所要的结果。

## 4.4 收敛性和稳定性

### 4.4.1 收敛性

**【定义 4.2】** 若单步法对任意固定的  $x_n = x_0 + nh$ , 如果数值有解, 数值解  $y_n$  当  $h \rightarrow 0$  时趋向于准确解  $y(x_n)$ , 则称该单步法为收敛。

**【定义 4.3】** 在单步法中定义  $|y(x_n) - y_n|$  为  $y_n$  整体的截断误差。

【定理 4.2】 假设单步法具有  $p$  阶精度，且满足利普希茨条件：

$$|\varphi(x, y, h) - \varphi(x, \bar{y}, h)| \leq L_\varphi |y - \bar{y}|$$

且初值  $y_0 = y(x_0)$  是准确的，则其整体截断误差为：

$$y(x_n) - y_n = O(h^p) \quad (4.4)$$

### 4.4.2 稳定性

【定义 4.4】 若一种数值方法在节点  $x_n$  处的数值解答  $y_n$  的初值解  $y_n$  有大小为  $\delta_n$  的扰动，而在以后的节点值  $y_m$  ( $m > n$ ) 产生的扰动绝对值不超过  $|\delta_n|$ ，即

$$|\delta_m| \leq |\delta_n|$$

则称该数值方法是稳定的。

部分稳定条件：

1. 尤拉法：  $0 \leq h \leq -\frac{2}{\lambda}$
2. 隐式尤拉法： 恒成立
3. 梯型公式： 恒成立
4. 标准 4 阶 Runge-Kutta 公式： 约为  $0 \leq h \leq -\frac{2.78}{\lambda}$

# Chapter 5

## 方程求根

### 5.1 根的隔离与二分法

#### 5.1.1 根的隔离

【定理 5.1】 若  $f(x)$  在  $[a, b]$  上连续，且  $f(a) * f(b) < 0$ ，则方程  $f(x) = 0$  在  $[a, b]$  内至少有一个根。

##### 1. 图解法

画出  $y = f(x)$  的粗略图，从而确定曲线  $y = f(x)$  与  $x$  轴的交点的粗略位置  $x_0$ ，作为近似值。

##### 2. 试验法

在某一区间中，取一些适当的数来试验。

#### 5.1.2 二分法

**基本思路**：逐步将含有  $x^*$  的区间进行二分，通过判断函数值的符号，逐步对半缩小有根区间，直到区间缩小到误差允许范围内，然后以区间中点作为  $x^*$  的近似值。

取区间  $[a, b]$  的中点  $x_0 = \frac{1}{2}(a + b)$ ，区间  $[a_k, b_k]$  的长度为

$$b_k - a_k = \frac{1}{2^k}(b - a)$$

又有  $x_k = \frac{1}{2}(a_k + b_k)$ ，所以

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

误差估计式

$$|x^* - x_k| \leq \frac{1}{2^{1+k}}(b - a)$$

## 5.2 迭代法及其收敛性

### 5.2.1 迭代法的基本概念

迭代法是指，用某种收敛于所给问题的精确解的极限过程，来逐步逼近的一种计算方法。  
对于一般形式的方程

$$f(x) = 0$$

先转化成等价的

$$x = g(x)$$

再从  $x_0$  出发，作序列  $\{x_n\}$

$$x_{n+1} = g(x_n) \quad (n = 0, 1, 2, \dots)$$

设

$$a = \lim_{n \rightarrow \infty}$$

可得

$$a = g(a)$$

所以

$$f(a) = 0$$

【定理 5.2】若迭代函数连续，那么，当选取  $x_0$  后若生成的序列  $x_n$  收敛，则一定收敛到根  $x^*$  上。

### 5.2.2 迭代法过程的收敛性

【定理 5.3】设迭代函数  $\varphi(x)$  在  $[a, b]$  上具有连续的一阶导数，且

- (1) 当  $x \in [a, b]$  时， $a \leq \varphi(x) \leq b$
- (2) 存在正数  $L < 1$ ，使对任意  $x \in [a, b]$ ，有

$$|\varphi'(x)| \leq L < 1$$

成立，则  $\varphi(x)$  在  $[a, b]$  上具有唯一解  $x^*$ ，且对任意初始近似值  $x_0 \in [a, b]$ ，迭代过程收敛，且

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

【定理 5.4】误差估计式

$$|x^* - x_k| \leq \frac{1}{1-L} |x_{k+1} - x_k|$$

$$|x^* - x_k| \leq \frac{L^k}{1-L} |x_1 - x_0|$$

【定理 5.5】设  $\varphi(x)$  在  $x = \varphi(x)$  的根  $x^*$  邻近有连续的一阶导数，且

$$|\varphi'(x^*)| < 1$$

则迭代过程具有局部收敛性。

## 5.3 收敛速度及收敛过程的加速

### 5.3.1 迭代的收敛速度

【定义 5.1】设序列  $\{x_k\}$  是收敛于方程  $f(x) = 0$  的根  $x^*$  的迭代序列，即  $x^* = \lim_{k \rightarrow \infty} x_k$ 。记  $\varepsilon_k = x_k - x^*$  表示各步的迭代误差。

若有某个实数  $p$  和非零常数  $C$ ，使

$$\lim_{k \rightarrow \infty} \frac{\varepsilon_{k+1}}{\varepsilon_k^p} = C \quad (C \neq 0)$$

则称序列  $x_k$  是  $p$  阶收敛的。 $p$  越大收敛速度就越快。

【定理 5.6】对于迭代过程，如果迭代函数  $\varphi(x)$  在所求根  $x^*$  的邻近有连续的二阶导数，且

$$|\varphi'(x^*)| < 1$$

则有

1. 当  $\varphi'(x^*) \neq 0$  时，迭代过程为线性收敛。
2. 当  $\varphi'(x^*) = 0$ ，而  $\varphi''(x^*) \neq 0$  时，迭代过程为二次收敛。

推导到一般情况，有如下定理

【定理 5.7】设  $x^*$  是方程  $x = \varphi(x)$  的根，在  $x^*$  的某一邻域， $\varphi(x)$  的  $m(m \geq 2)$  阶导数连续，且

$$\varphi'(x^*) = \dots = \varphi^{(m-1)}(x^*) = 0$$

$$\varphi^{(m)}(x^*) \neq 0$$

则有  $m$  阶收敛。



### 5.3.2 收敛过程的加速

假定  $g'(x)$  在某范围内近似为  $a$ ，则误差：

$$x^* - \bar{x}_{k+1} \approx \frac{a}{1-a}(\bar{x}_{k+1} - x_k)$$

将误差结果作为补偿，则

$$\begin{aligned} x_{k+1} &= \bar{x}_{k+1} + \frac{a}{1-a}(\bar{x}_{k+1} - x_k) \\ &= \frac{1}{1-a}\bar{x}_{k+1} - \frac{a}{1-a}x_k \end{aligned}$$

此过程可归纳为：

$$\left\{ \begin{array}{ll} \text{迭代:} & \bar{x}_{k+1} = g(x_k) \\ \text{改进:} & x_{k+1} = \bar{x}_{k+1} + \frac{a}{1-a}(\bar{x}_{k+1} - x_k) \end{array} \right.$$

**Aitken 加速法：**

$$\left\{ \begin{array}{ll} \text{迭代:} & x_{k+1}^{(1)} = g(x_k) \\ \text{迭代:} & x_{k+1}^{(2)} = g(x_{k+1}^{(1)}) \\ \text{改进:} & x_{k+1} = x_{k+1}^{(2)} - \frac{(x_{k+1}^{(2)} - x_{k+1}^{(1)})^2}{x_{k+1}^{(2)} - 2x_{k+1}^{(1)} + x_k} \end{array} \right.$$

## 5.4 牛顿法

### 5.4.1 牛顿法的构造及牛顿迭代公式

$$\varphi(x) = x - \frac{f(x)}{f'(x)}$$

### 5.4.2 牛顿法的收敛性和收敛速度

【定理 5.8】在牛顿法中，若  $x^*$  是方程  $f(x) = 0$  的一个单根，并且  $f(x)$  在  $x^*$  及其附近有连续的二阶导数，则牛顿法具有局部收敛性。

【定理 5.9】若对于方程  $f(x) = 0$ ，如果在单根  $x^*$  附近有连续的二阶导，则牛顿法至少平方收敛，若  $f''(x) \neq 0$ ，则牛顿法平方收敛。

### 5.4.3 初始值的选取

如果

$$|f'(x_0)|^2 > \left| \frac{f''(x_0)}{2} \right| |f(x_0)|$$

且  $f''(x_0) \neq 0$ ，则可以选择  $x_0$  作为初始值。

### 5.4.4 牛顿法下山法

为满足函数值单调下降，将牛顿法计算结果

$$\bar{x}_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

与前一步的值  $x_k$  加权平均

$$x_{k+1} = \lambda \bar{x}_{k+1} + (1 - \lambda)x_k$$

或

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

其中  $0 < \lambda \leq 1$  称下山因子。