

计算方法与算法语言

(修订本)

南京航空学院《计算数学》编写组

1976.11.

前 言

电子数字计算机是一种现代化自动计算工具，它具有……等特点，因此可以利用它来完成大量繁复的计算工作，解决科学技术、工程设计以及其它各个领域中的计算问题。（手算需要几年的计算量，计算机只要几秒就能完成）。

近代的电子数字计算机，由“硬件”与“软件”两大部分组成。“硬件”又称“硬设备”，是由电子器件等构成的计算机本体（X—2、709这类单通道的中、小型机器的硬件一般有运算器、存储器、控制器以及输入设备、输出设备等五个部分；输入设备与输出设备统称为计算机的外部设备。）“软件”又称“软设备”，是由编译系统等非电子器件的设备构成。编译系统是用机器语言（又称机器指令）编写成的一个加工程序，它能够把用算法语言编写的源程序、编排并翻译成用机器语言表示的目标程序（又称目的程序）。

本讲义包括两部分：第一部分、计算方法——研究数学问题数值计算的方法，因而又称数值分析；第二部分、算法语言（算法语言有许多种，我们这里介绍的是国产X—2机、709机、719机、DJS—6机、DJS—21机等机器上用的Algol—60算法语言）。

在进入这两方面具体内容的学习和研究以前，让我们简要地介绍一下电子数字计算机的发展概况及应用电子数字计算机解题的工作流程。

1. 电子数字计算机发展概况：

1945年为了满足计算弹道的需要制造成功了世界上第一台电子数字计算机，三十多年来电子计算机日新月异、发展极为迅速。它的基本电路从第一代采用的电子管发展到第二代采用的晶体管再发展到第三代采用的集成电路，目前已发展到采用大规模集成电路（采用大规模集成电路的计算机通称为第四代）；信息的通道也从单通道发展到多通道（多通道机可有多个通道同时进行运算）；运算速度从开始的每秒几十次发展到目前已出现每秒能运算几千万次甚至上亿次；存储量也越来越大。特别是六十年代以来，软设备被引入机器，在程序编制方面发生了根本的改革，有力地促进了计算机的应用。目前，电子数字计算机的功能已远远越出了“计算”这一范围，电子数字计算机不仅能用来解决各种繁复的数学计算问题，而且在工业生产的自动控制，国民经济的综合平衡、最佳设计方案的选择、文字翻译、资料检索、火炮自动瞄准等等各个领域都有着广泛应用。

在航空事业中，电子计算机的用途也极为广泛。不管是研制新的飞行器还是空中交通管制、导航、制导、发射人造卫星，……，电子计算机都是必不可少的工具。就拿研制一架新飞机来讲吧，在飞机的初步设计阶段要根据所提出的使用要求确定总体方案，进行基本气动力计算，初步进行结构设计，性能分析，模型的风洞吹风试验的数据处理，确定机翼的几何形状，机身的安排、发动机安装部位和气动载荷分布状况，检验飞机的安定性，操纵性是否符合要求等等，每一步都会遇到大量复杂的数学计算问题。运用电子计算机就可以在很短的时间内迅速获得正确的结果，加快飞机设计的进度，赢得时间。所以学会运用电子计算机的方法对我们从事航空事业和帝、修、反争时间，“备战、备荒、为人民”具有很大的意义。

毛主席教导我们：“中国人民有志气，有能力，一定要在不远的将来，赶上和超过世界先进水平。”

在以毛主席为首的党中央的正确领导下，我国的电子计算机事业得到了迅速的发展。根据1974年北京举办的一次“电子计算机展览”有关资料汇编介绍：

1958年、制成功电子管数字计算机（每秒运算一万次级）；

1964年、研制成功全晶体管电子数字计算机（每秒运算一至十万次）；

1970年我国第一颗人造地球卫星发射成功，它标志着我国电子计算机事业达到了新的水平；

1971年、研制成功集成电路数字计算机（每秒运算二十万次级）；

1972年、北京大学、北京有线电厂和燃化部等有关单位共同协作研制成大型集成电路电子数字计算机（每秒运算一百万次级），这是毛主席革命路线的伟大胜利，是无产阶级文化大革命的伟大胜利。我们深信一定能在不远的将来，赶上和超过世界先进水平。

2.应用电子数字计算机解题的工作流程：

在未配备有软设备的机器上解题时，只能用手编程序（手编程序是用机器指令编制的算题程序），用手编程序解题的工作流程大致如下：

构造数学模型→选择计算方法→用机器指令人工编制计算程序→程序和原始数据的信息化→计算机自动计算（程序及数据代码输入→计算→计算结果输出）→分析结果。

所谓数学模型，是指对实际问题所作的数学描述。无论工程设计活动或自然界的某种基本运动规律等等，总是经过反复试验，分析之后用一系列数学算式来描述，这种对实际问题进行的数学描述就称为数学模型。

构造了数学模型之后，就得选择运算简单、工作量节省，并能保证精度要求的计算方法。目前，计算方法学科的研究成果，已经为各种数学问题的数值计算提供了适于机器运算的大量行之有效的办法，供选用。

计算方法中的计算公式，原始数据和计算步骤，构成计算问题的完整计算过程。在未配备有软设备的机器上解题时，为了让计算机来实现这个过程，必须把它们用机器能认识的语言（即机器指令）编写成计算程序，这一工作通称程序设计，在软设备未引入机器的时期，程序设计的工作要由一支队伍相当庞大的专业人员（程序设计员）来进行。

计算程序编好后（或者说程序设计工作完成后）就要将程序以及原始数据按一定的规则在约定的纸带上进行穿孔，这就是程序和原始数据的信息化。

只有上述各项工作，依次完成后才能进入计算机自动计算的阶段。

从上述流程可以看出：人工干预频繁，有的环节还必须有专业人员来承担，特别是程序设计工作，效率低，周期长，易出错，检查和调整程序的工作量很大。有的问题程序设计的工作量要按月计算，随着计算机的计算速度越来越快，而上机计算所要的时间则往往是按分计算即够。所以手编程序早已完全不适应计算机高速发展的形势，必须来一个根本性的改变，简化程序的编制，便于使用者的掌握，算法语言又称程序设计语言就是为了解决这一矛盾而产生和发展起来的。

Algol—60是程序设计语言中的一种，它是1960年在一次国际会议上定为国际语言的。Algol及Fortran这两种程序设计语言，是国际上普遍公认的科技计算方面的重要语言，目前应用的范围也最广泛。

前曾指出，用算法语言编制的算题程序称为源程序。

在装备有编译系统的机器上解题时，只要用相应的算法语言编制源程序，而不再需要进行手编程序的程序设计工作。

用源程序解题的工作流程大致如下：

构造数学模型→选择计算方法→用算法语言编制源程序→计算机自动计算（源程序代码输入→由编译程序把源程序编译成目标程序→机器根据目标程序实现计算→计算结果输出）→分析结果。

采用算法语言编制源程序的显著优点是，程序编制工作简便，便于学习掌握，计算机的使用者可自己编制源程序有利于计算机应用的普及和提高机器的使用效率。且机器的指令是每种类型的机器各不相同的，在这一种机器上能用的手编程序，到另一类型机器上则毫无用处。而源程序是用算法语言编制的。只须稍加修改就可在具有同一种算法语言的编译程序的另一种类型机器上应用。

且熟悉一种算法语言之后，对学习和掌握另一种算法语言有很大的启示作用。

还必须指出：编译程序是软件的必备部分；随着计算机的发展，目前，一台性能良好的电子数字计算机应配备有面向用户的，面向维修、管理人员的以及面向计算机本身的软件系统。

目 录

第一部分 计 算 方 法

第一章 插值与曲线拟合.....	(1)
§ 1.1. 引言.....	(1)
§ 1.2. 线性插值与二次插值.....	(2)
§ 1.3. 均差插值多项式 (牛顿插值多项式).....	(5)
§ 1.4. 拉格兰日插值多项式.....	(10)
§ 1.5. 关于数值微分.....	(13)
§ 1.6. 建立线型经验公式的最小二乘法.....	(14)
§ 1.7. 多项式经验函数的正规方程组.....	(16)
§ 1.8. 应用的例.....	(17)
第一章 习题.....	(21)
第二章 数值积分.....	(24)
§ 2.1. 矩形公式与梯形公式.....	(24)
§ 2.2. 抛物线公式 (辛甫生公式).....	(28)
§ 2.3. 变步长抛物线公式.....	(31)
第二章 习题.....	(32)
第三章 常微分方程初值问题数值解.....	(34)
§ 3.1. 引论.....	(34)
§ 3.2. 尤拉法.....	(35)
§ 3.3. 改进的尤拉法.....	(37)
§ 3.4. 龙格—库塔方法 (R—K 方法).....	(39)
第三章 习题.....	(40)
第四章 线性代数方程组的解法.....	(41)
§ 4.1. 矩阵的一些简单知识.....	(41)
§ 4.2. 行列式的概念和它的一些性质.....	(49)
§ 4.3. 逆矩阵的概念及几种特殊类型的矩阵.....	(55)
§ 4.4. 主元消去法.....	(61)

§ 4.5. 普通迭代法 (简单迭代法)	(69)
§ 4.6. 塞德尔迭代法	(74)
§ 4.7. 追赶法求实三对角线性方程组的解	(77)
第四章 习题	(79)
第五章 样条插值简介	(83)
§ 5.1. 引论	(83)
§ 5.2. 三次样条插值	(84)

第二部分 算法语言 (ALGOL—60)

第一章 电子数字计算机概述	(90)
§ 1. 电子数字计算机的基本结构	(90)
§ 2. 电子数字计算机中数的表示形式	(92)
§ 2.1. 十进制、二进制、八进制	(92)
§ 2.2. 数制的转换	(95)
§ 2.3. $x-2$ 机中数的表示形式	(98)
§ 2.4. 数的浮点二——十进制形式	(100)
§ 3. $X-2$ 机中指令的表示形式	(101)
第二章 程序的编写方法	(103)
§ 1. 程序的初步认识	(103)
§ 2. 算法语言的基本成分	(104)
§ 2.1. 基本符号	(104)
§ 2.2. 标识符、常量、变量、标准函数	(105)
§ 2.3. 表达式	(107)
§ 3. 基本语句	(110)
§ 3.1. 赋值语句	(110)
§ 3.2. 条件语句	(111)
§ 3.3. 复合语句	(114)
§ 3.4. 循环语句	(114)
§ 3.5. 转向语句	(119)
§ 3.6. 空语句	(121)
§ 3.7. 停机语句	(121)
§ 3.8. 输入语句	(122)
§ 3.9. 输出语句	(123)
§ 4. 说明和分程序	(130)
§ 4.1. 简单变量说明	(130)

§ 4.2. 数组说明	(132)
§ 4.3. 分程序	(136)
§ 4.4. 开关说明	(140)
§ 5. 过程说明和过程调用	(145)
§ 5.1. 无参数过程	(145)
§ 5.2. 带参数过程	(147)
§ 5.3. 函数过程	(155)
§ 5.4. 常用算法的标准过程	(158)
第三章 上机计算	(162)
§ 1. 穿孔	(162)
§ 2. 控制台介绍	(164)
§ 3. 上机步骤	(165)
§ 4. 语法检查的输出信息和编译过程中输出内容	(168)
§ 4.1 语法检查和编排的输出信息	(168)
§ 4.1 编译过程中输出的内容	(168)
§ 5. 一些操作技术	(170)
附录一 语法错误性质对照表	(173)
附录二 编排过程中出现的错误性质表	(174)
附录三 出错停机表	(175)
附录四 目标程序的内存分配	(176)
附录五 汉语方案	(177)
附录六 常用算法标准过程举例	(178)
(一)拉格朗日一元 n 点插值	(178)
(二)拉格朗日一元三点插值	(179)
(三)拉格朗日二元 n 点插值	(181)
(四)拉格朗日二元三点插值	(183)
(五)辛浦生法(定步长)求积分值	(184)
(六)辛浦生法(变步长)计算定积分值	(186)
(七)辛浦生法(变步长)计算二重积分值	(188)
(八)高斯全主元消去法解线性代数方程组	(190)
(九)线性代数方程组的高斯——赛德尔迭代法	(192)
(十)全主元高斯——约当消去法求逆矩阵	(194)
(十一)龙格——库塔法解常微分方程组(定步长)	(196)
(十二)龙格——库塔法解常微分方程组(变步长)	(198)
(十三)求高次代数方程全部实根的牛顿法	(201)
(十四)曲线拟合问题的最小二乘法	(202)
(十五)三次样条插值(追赶法)的程序	(206)

第一部分 计算方法

第一章 插值与曲线拟合

§ 1—1 引言

在生产和科研中,函数是表达现实空间物质运动规律的数量关系的。实际问题遇到的函数 $y=f(x)$,虽然从原则上说,它在某个区间 $[a, b]$ 上是存在的。但是,通常是通过实验观察得到的,因此只能知道在 $[a, b]$ 区间上一系列点 x_i 上的相应函数值 $y_i=f(x_i)$,对于 x 的其他值, $y=f(x)$ 的变化情况是不知道的,也就是说我们往往只知道函数 $y=f(x)$ 的一张表格 (x_i, y_i) , $i=1, \dots, n$; 用它来研究函数的性质是不方便的,也不能知道表格以外的点上的值,若函数能用公式表示,即使是近似的,对于我们研究、分析函数的性质、计算表中没有列出的函数值都是非常有用的。问题是:这种近似式有没有?若有,我们怎么来构造?

例1. 某号飞机模型进行风洞吹风试验。对同一攻角 α 测得不同 M 数所对应的阻力系数 C_x 值如下表

M	0.527	0.727	0.807	0.927	0.973	1.005	1.066
C_x	0.01075	0.01219	0.01188	0.01426	0.01668	0.02214	0.02895

现在的问题是:怎样根据上表求出 M 与 C_x 的近似关系式,从而计算出表中未给出的 M 值处相应的 C_x 值。

插值法就是为了解决上面问题提出的,根据已经给出的函数在一些点上的值寻求函数近似表达式的一种方法。

由于代数多项式是简单的函数,人们很早就用它去近似地表示复杂的函数,代数插值法就是以代数多项式作为工具,去近似地表达函数的方法,既然寻求的代数多项式能近似地表达用列表法给出的函数关系,当然要求在表中给出的自变量点上,代数多项式的值与相应的表中给出的函数值相等,插值多项式的构造就是从这一点出发的。现用数学语言表述如下:

设 $y=f(x)$ 是区间 $[a, b]$ 上的连续函数,已知它在 $[a, b]$ 上 $n+1$ 个互不相同的点 $x_0, x_1, \dots, x_n$ 上取值 $y_0, y_1, \dots, y_n$ 。若代数多项式 $p(x)$ 在点 x_i 上满足 $p(x_i)=y_i$, ($i=0, 1, \dots, n$), 则称 $p(x)$ 是函数 $y=f(x)$ 的插值多项式。 $x_0, x_1, \dots, x_n$ 是插值点或基点又稱节点,包括基点的区间 $[a, b]$ 称为插值区间。 $y=f(x)$ 称为被插函数,它通常是列表函数。插值法就是求满足上述条件的多项式 $p(x)$ 的方法。

用 $p(x)$ 算出的插值区间内不是基点的 x 上的值就作为被插函数在该点的近似值,这样的点 x , 通常称被插值点。这类插值, 通称内插。

在区间 $[a, b]$ 上用 $p(x)$ 近似 $f(x)$, 其几何意义就是通过 $n+1$ 个点 $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ 作代数曲线 $y=p(x)$ 来近似曲线 $y=f(x)$, 如图1.1所示。当 $n=1$ 时, 曲线就是通过 $(x_0, y_0), (x_1, y_1)$ 的直线。当 $n=2$ 时, 就是通过 $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ 的抛物线。

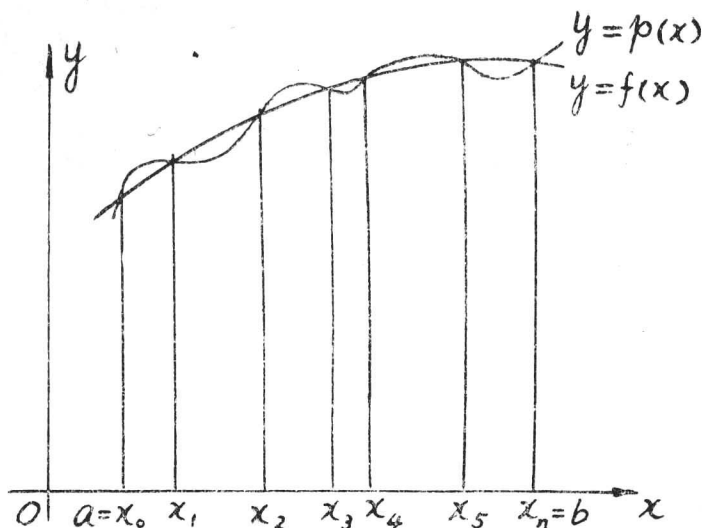


图 (1.1)

由图1.1可见，在基点上， $p(x_i) = f(x_i)$ ， $i=0, 1, \dots, n$ 。而在其他点上往往会有误差。一般来说， $|f(x) - p(x)|$ 越小则近似程度就越好。

还必须指出，用插值多项式来近似地表示被插函数，不仅因为 $f(x)$ 是由一张表格给出而需要，有时在 $f(x)$ 的解析式子已经给出，但计算起来相当复杂，工作量极大，而又无法简化该式，也可以根据该函数在一些点上的值构造插值多项式来近似表达它，以减少工作量。用插值多项式来表达函数也是研究数值微分等问题的基础，可以说它是计算方法中最基本的方法。在实际工作中，经常会遇到这类问题。插值多项式多种多样，本章我们着重介绍数字电子计算机上常用的两种：牛顿均差插值多项式与拉格朗日插值多项式。关于插值问题的进一步讨论可参考书目[1]，[2]，[3]。

§ 1—2 线性插值

1. 线性插值：

已知函数 $f(x)$ ，在点 x_0, x_1 上的值为 y_0, y_1 ；要求多项式 $p_1(x)$ ，使得 $p_1(x_0) = y_0$ ， $p_1(x_1) = y_1$ 。

显然通过 $A(x_0, y_0)$ ， $B(x_1, y_1)$ 两点的直线方程就满足上述要求（图 1.2）

由初等数学可知：过 A, B 两点的直线方程是：

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}.$$

即：

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0).$$

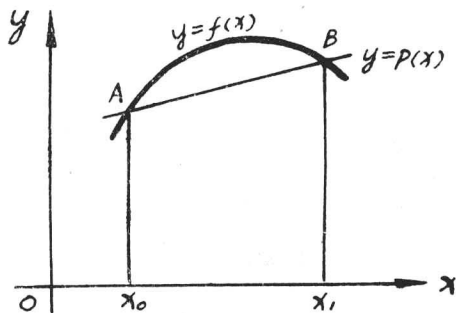


图 (1.2)

所以

$$p_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0). \quad (1.1)$$

(1.1)式又可写为:

$$p_1(x) = \frac{x - x_1}{x_0 - x_1}y_0 + \frac{x - x_0}{x_1 - x_0}y_1. \quad (1.1)^*$$

例1. 设函数 C_x 由表给出:

M	0.527	0.727
C_x	0.01075	0.01219

试求 $C_x(0.7)$ 的值,

解: 应用线性插值多项式(1.1)*

这里 $x_0 = 0.527$, $y_0 = 0.01075$, $x_1 = 0.727$, $y_1 = 0.01219$, 求 $y(0.7)$ 的值.

$$\begin{aligned} p_1(0.7) &= \frac{0.7 - 0.727}{0.527 - 0.727} \times 0.01075 + \frac{0.7 - 0.527}{0.727 - 0.527} \times 0.01219 \\ &= 0.0107 \times \frac{0.027}{0.2} + 0.01219 \times \frac{0.173}{0.2} \\ &\doteq 0.01110. \end{aligned}$$

$$\therefore y(0.7) = C_x \doteq (0.7) 0.01110$$

例2. 设函数 y 由下表给出:

x	y
12	1.07918
13	1.11394

试用线性内插, 求 $x = 12.7$ 处的值.

解: 应用线性插值多项式(1.1)

由于这里

$$x_0 = 12, \quad y_0 = 1.07918;$$

$$x_1 = 13, \quad y_1 = 1.11394.$$

$$\begin{aligned} \therefore p_1(x) &= 1.07918 + \frac{1.11394 - 1.07918}{13 - 12}(x - 12) \\ &= 1.07918 + 0.03476(x - 12). \end{aligned}$$

从而

$$\begin{aligned} p_1(12.7) &= 1.07918 + 0.03476(12.7 - 12) \\ &= 1.10351. \end{aligned}$$

倘若 x 在区间 $[x_0, x_1]$ 之外, 又离 x_0 或 x_1 不远, 也可用 $p_1(x)$ 的值作为 $f(x)$ 的近似值. 这类插值, 通称外插 (或外推).

当 $|x_1 - x_0|$ 很小或问题的要求不高时, 线性插值是经常使用的.

2. 二次插值 (抛物插值):

线性插值只利用 (x_0, y_0) , (x_1, y_1) 两组值来求 $f(x)$ 的近似值, 由误差分析可知误差较大. 为了减少误差, 我们设想再利用一组值来求 $y = f(x)$ 的近似值, 其结果比线性插值要好些. 因此现在的问题就是已知在点 x_0, x_1, x_2 上相对应的值为 y_0, y_1, y_2 , 要求一个多项式 $p_2(x)$, 使得 $p_2(x_0) = y_0$, $p_2(x_1) = y_1$, $p_2(x_2) = y_2$.

前已看到, 过两点的插值多项式是一次式, 可以设想过三点的插值多项式是二次式, 设为 $p_2(x) = ax^2 + bx + c$ 。在几何上这是过 (x_0, y_0) , (x_1, y_1) , (x_2, y_2) 的一条抛物线。现在的问题是: 求系数 a, b, c 使满足 $p_2(x_0) = y_0$, $p_2(x_1) = y_1$, $p_2(x_2) = y_2$ 。虽然可以由联立方程:

$$\begin{cases} ax_0^2 + bx_0 + c = y_0, \\ ax_1^2 + bx_1 + c = y_1, \\ ax_2^2 + bx_2 + c = y_2. \end{cases}$$

来解出 a, b, c , 但这样作既比较麻烦, 且不利于在插值问题中作深入的理论研究, 因此我们由另一途径来处理这一问题。

前面已得到了满足 $p_1(x_0) = y_0$, $p_1(x_1) = y_1$ 的一次插值多项式 $p_1(x)$ 。

现在把所要求的 $p_2(x)$ 分为两项, 一项为 $p_1(x)$ 。剩下的一项称为 $p_2^*(x)$, 则

$$p_2(x) = p_2^*(x) + p_1(x).$$

由 $p_1(x_0) = y_0$, $p_1(x_1) = y_1$, 要使 $p_2(x_0) = y_0$, $p_2(x_1) = y_1$, 必然要 $p_2^*(x_0) = p_2^*(x_1) = 0$, 故可取 $p_2^*(x) = A(x-x_0)(x-x_1)$, 问题归结为求 A 。

由 $p_2(x_2) = y_2 = p_2^*(x_2) + p_1(x_2)$ 得:

$$\begin{aligned} A &= \frac{y_2 - p_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{y_2 - y_0 - \frac{y_1 - y_0}{x_1 - x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{\frac{y_2 - y_0}{x_2 - x_0} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_1}. \end{aligned}$$

故有:

$$\begin{aligned} p_2(x) &= y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + \\ &\quad + \frac{\frac{y_2 - y_0}{x_2 - x_0} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_1}(x - x_0)(x - x_1). \end{aligned} \quad (1.2)$$

显然

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

所以 $p_2(x)$ 为我们所要求的二次插值多项式。二次插值又称抛物插值。

例3. 设函数 y 由下表给出:

x	25	36	49
y	5	6	7

试求通过上述三点的二次插值多项式, 并求 $x=41$ 时的近似值。

解: 应用二次插值公式(1.2)

由于

$$x_0 = 25, \quad x_1 = 36, \quad x_2 = 49;$$

$$y_0 = 5, \quad y_1 = 6, \quad y_2 = 7.$$

∴ 有

$$y_1 - y_0 = 6 - 5 = 1,$$

$$x_1 - x_0 = 36 - 25 = 11,$$

$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{1}{11};$$

$$y_2 - y_0 = 7 - 5 = 2,$$

$$x_2 - x_0 = 49 - 25 = 24,$$

$$\frac{y_2 - y_0}{x_2 - x_0} = \frac{2}{24} = \frac{1}{12};$$

$$x_2 - x_1 = 49 - 36 = 13.$$

$$\frac{\frac{y_2 - y_0}{x_2 - x_0} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_1} = \frac{\frac{1}{12} - \frac{1}{11}}{13} = \frac{-\frac{1}{132}}{13} = -\frac{1}{132} \times \frac{1}{13} = -\frac{1}{1716}.$$

从而有:

$$p_2(x) = 5 + \frac{1}{11}(x-25) - \frac{1}{1716}(x-25)(x-36).$$

这就是新要求的二次插值多项式。

用 $x=41$ 代入, 便得出:

$$\begin{aligned} p_2(41) &= 5 + \frac{1}{11}(41-25) - \frac{1}{1716}(41-25)(41-36) \\ &\approx 6.4079. \end{aligned}$$

§ 1—3 均差插值多项式(牛顿插值多项式)

前面, 从最简单的插值问题推出了一次、二次插值多项式。我们看到: 两个插值点可求出一插值多项式 $p_1(x)$, 而三个插值点可定出二次插值多项式 $p_2(x)$ 、(它有三个待定系数), 一般说, 当插值点增加到 $n+1$ 个时, 所求的插值多项式次数也不会超过 n 次, 其一般形式如下:

$$p_n(x) = a_0 + a_1x + \cdots + a_nx^n. \quad (1.3)$$

因此, 插值问题的一般提法是:

已知函数 $y=f(x)$ 在 $n+1$ 个不相同的点 $x_0, x_1, \cdots, x_n$ 上取值 $y_0, y_1, \cdots, y_n$, 要求一个次数不超过 n 次的多项式 $p_n(x)$, 使

$$p_n(x_j) = y_j, \quad (j=0, 1, \cdots, n),$$

现在的问题是用什么方法求出满足此条件的 $p_n(x)$, 一种做法是根据给出的条件从(1.3)中求出 $n+1$ 个待定系数 $a_0, a_1, \cdots, a_n$, 这样作需要解 $n+1$ 个联立方程, 计算复杂。另一种做法就是上节推导二次插值多项式的方法。我们采用后一方法。为此, 对(1.2)作进一步分析, 从中找出规律性的东西。

1. 均差的定义, 性质及均差表:

公式 $p_1(x)$ 中出现的 $\frac{y_1 - y_0}{x_1 - x_0}$, 实际上是函数 $y=f(x)$ 在区间 (x_0, x_1) 上的平均变化率,

我们称它为 $f(x)$ 的一阶均差 (或称差商)。记作

$$f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0};$$

类似地, 有:

$$f(x_0, x_2) = \frac{y_2 - y_0}{x_2 - x_0}, \quad f(x_1, x_2) = \frac{y_2 - y_1}{x_2 - x_1}; \dots$$

公式 $p_2(x)$ 中的二次因子的系数

$$\frac{\frac{y_2 - y_0}{x_2 - x_0} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_1}$$

实际上就是:

$$\frac{f(x_0, x_2) - f(x_0, x_1)}{x_2 - x_1}.$$

也就是说, 是一阶均差的均差, 记作

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

称为 $f(x)$ 的二阶均差 (或二阶差商)。

一般地, 可定义 $f(x)$ 的 n 阶均差为:

$$f(x_0, x_1, \dots, x_n) = \frac{f(x_1, x_2, \dots, x_n) - f(x_0, x_1, \dots, x_{n-1})}{x_n - x_0}, \quad (1.4)$$

通过计算, 易验证

$$\begin{aligned} f(x_0, x_1) &= f(x_1, x_0); \\ f(x_0, x_1, x_2) &= f(x_0, x_2, x_1) = f(x_2, x_1, x_0) = f(x_2, x_0, x_1) \\ &= f(x_1, x_2, x_0) = f(x_1, x_0, x_2); \\ &\dots\dots\dots \end{aligned}$$

上述性质通称为均差的对称性,

也就是说: 一阶均差 $f(x_0, x_1)$ 的值, 只与插值点 x_0, x_1 的选择有关, 而与插值点的排列顺序无关。

二阶均差 $f(x_0, x_1, x_2)$ 的值, 只与插值点 x_0, x_1, x_2 的选择有关, 而与插值点的排列顺序无关。

一般说来, n 阶均差 $f(x_0, x_1, \dots, x_n)$ 只与插值点 $x_0, x_1, \dots, x_n$ 的选择有关, 而与插值点的排列顺序无关。

均差的对称性是均差的很重要的一个性质。

均差的另一个重要的性质是: $f(x)$ 的任意阶均差可以通过其插值点的坐标及函数值表示出来。

例如, 一阶差商 $f(x_0, x_1)$ 可表示为:

$$f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0}.$$

二阶差商 $f(x_0, x_1, x_2)$ 可表示为:

$$f(x_0, x_1, x_2) = \frac{f(x_0)}{(x_0-x_1)(x_0-x_2)} + \frac{f(x_1)}{(x_1-x_0)(x_1-x_2)} + \frac{f(x_2)}{(x_2-x_0)(x_2-x_1)}.$$

可以证明对一般的 n 阶均差, 有:

$$f(x_0, x_1, x_2, \dots, x_n) = \sum_{j=0}^n \frac{f(x_j)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)} \quad (1.5)$$

为了计算各阶均差通常采用以下的均差表 (以 $n=4$ 为例)

x_i	$f(x_i)$	一阶均差 $f(x_{i-1}, x_i)$	二阶均差 $f(x_{i-1}, x_i, x_{i+1})$	三阶均差 $f(x_{i-1}, x_i, x_{i+1}, x_{i+2})$	四阶均差 $f(x_{i-1}, x_i, \dots, x_{i+4})$
x_0	$f(x_0)$				
x_1	$f(x_1)$	$f(x_0, x_1)$			
x_2	$f(x_2)$	$f(x_1, x_2)$	$f(x_0, x_1, x_2)$	$f(x_0, x_1, x_2, x_3)$	
x_3	$f(x_3)$	$f(x_2, x_3)$	$f(x_1, x_2, x_3)$	$f(x_1, x_2, x_3, x_4)$	$f(x_0, x_1, x_2, x_3, x_4)$
x_4	$f(x_4)$	$f(x_3, x_4)$	$f(x_2, x_3, x_4)$		

表 (1.1)

例1. 设有函数表如下:

x	0.361	0.409	0.547	0.446
$y=f(x)$	1.06587	1.08481	1.15337	1.10112

试造均差表:

解: 据均差的定义, 可得表(1.1)形式的均差表如次:

x_i	$f(x_i)$	一阶均差 $f(x_{i-1}, x_i)$	二阶均差 $f(x_{i-1}, x_i, x_{i+1})$	三阶均差 $f(x_{i-1}, x_i, x_{i+1}, x_{i+2})$
0.361	1.06587			
0.409	1.08481	0.39458		
0.547	1.15337	0.49681	0.54962	
0.446	1.10112	0.51733	0.55495	0.58471

例2. 设有函数表如次:

x	5	7	11	13	21
$f(x)$	150	392	1452	2366	9702

试造 $f(x)$ 的均差表:

解: 仿上, 可得:

x_i	$f(x_i)$	一阶均差	二阶均差	三阶均差	四阶均差
5	150				
7	392	121			
11	1452	265	24		
13	2366	457	32	1	
21	9702	917	46	1	0

2. 均差插值多项式:

引入均差的定义后, 公式 $p_1(x)$ 及 $p_2(x)$ 的规律就更明显了, 事实上, 这时(1.1)及(1.2)可分别写成:

$$p_1(x) = f(x_0) + f(x_0, x_1)(x - x_0), \quad (1.1)^{**}$$

$$p_2(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1). \quad (1.2)^*$$

上述两个公式应用均差的定义及对称性质极易验证。

为了得出 $n+1$ 个插值的均差插值多项式及其余项, 除了事先给出的 $n+1$ 个插值点 x_j ($j=0, 1, \dots, n$) 而外需再给出一个点 $x \neq x_j$, 它可以是插值区间 $[a, b]$ 上的任一点。

据均差的定义, 可知:

$$\left. \begin{aligned} f(x) &= f(x_0) + f(x, x_0)(x - x_0), & (1.6.1) \\ f(x, x_0) &= f(x_0, x_1) + f(x, x_0, x_1)(x - x_1), & (1.6.2) \\ f(x, x_0, x_1) &= f(x_0, x_1, x_2) + f(x, x_0, x_1, x_2)(x - x_2), & (1.6.3) \\ &\dots\dots\dots \\ f(x, x_0, x_1, \dots, x_n) &= f(x_0, x_1, \dots, x_n) \\ &\quad + f(x, x_0, x_1, \dots, x_n)(x - x_n). \end{aligned} \right\} \quad (1.6)$$

兹用(1.6.2)代入(1.6.1)可得:

$$f(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x, x_0, x_1)(x - x_0)(x - x_1). \quad (1.7)$$

即:

$f(x) = p_1(x) + R_1(x)$, 式中 $R_1(x) = f(x, x_0, x_1)(x - x_0)(x - x_1)$ 称为 $p_1(x)$ 的余项。

类似地, 用(1.6.3)代入(1.7), 可得:

$$f(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + f(x, x_0, x_1, x_2)(x - x_0)(x - x_1)(x - x_2).$$

即:

$f(x) = p_2(x) + R_2(x)$, 式中 $R_2(x) = f(x, x_0, x_1, x_2)(x - x_0)(x - x_1)(x - x_2)$ 称为 $p_2(x)$ 的余项。

由此, 可总结出一般规律: 每增加一个插值点, 只要将(1.6)中相应的高阶均差代入前一公式, 依此类推, 可得:

$$\begin{aligned} f(x) &= f(x_0) + f(x_0, x_1)(x - x_0) + \dots \\ &\quad + f(x_0, x_1, \dots, x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}) \\ &\quad + f(x, x_0, x_1, \dots, x_n)(x - x_0)(x - x_1) \dots (x - x_n) \end{aligned} \quad (1.8)$$

记

$$p_n(x) = f(x_0) + f(x_0, x_1)(x-x_0) + \dots + f(x_0, x_1, \dots, x_n)(x-x_0)(x-x_1)\dots(x-x_{n-1}). \quad (1.9)$$

$$R_n(x) = f(x, x_0, x_1, \dots, x_n)(x-x_0)(x-x_1)\dots(x-x_n). \quad (1.10)$$

则 $p_n(x)$ 称为 n 次均差插值多项式 (或 n 次牛顿插值多项式);

$R_n(x)$ 称为 n 次均差插值多项式的余项。

而 (1.8) 则称为均差插值公式 (或牛顿插值公式)。

必须指出, (1.8)、(1.9) 及 (1.5) 是插值问题最基本的结果, 从上面的讨论, 可以看出: 均差插值多项式在结构上有一个重要特点 (也是它的最大优点), 即每增加一个插值点, 只要多计算一项, 而 $p_n(x)$ 的各项系数恰好是各阶均差值, 很有规律, 且它们可在均差表 (1.1) 中第一条斜线上找到。

3. 实例:

例1. 给定 $f(x)$ 的函数表如下, 作插值多项式并计算 $f(0.596)$ 的值。

解:

(1) 根据给定函数表作均差表如下:

k	x_k	$f(x_k)$	一阶均差	二阶均差	三阶均差	四阶均差	$0.596 - x_k$
0	0.40	0.41075					0.196
1	0.55	0.57815	1.1160				0.046
2	0.65	0.69675	1.1860	0.2800			-0.054
3	0.80	0.88811	1.2757	0.3588	0.197		-0.204
4	0.90	1.02652	1.3841	0.4336	0.214	0.034	
5	1.05	1.25386	1.5156	0.5260	0.231	0.034	

(2) 决定插值多项式的阶:

从均差表看出, 四阶均差是常数, 故五阶均差必为零。

因此可作四阶插值多项式: $p_4(x)$ 。

(3) 选定插值点, 写出 $p_4(x)$:

为了计算 $f(0.596)$, 选靠近 $\bar{x} = 0.596$ 的五个点 x_0, x_1, x_2, x_3, x_4 作插值点。

则有:

$$p_4(x) = 0.41075 + 1.1160(x-0.4) + 0.2800(x-0.4)(x-0.55) + \\ + 0.197(x-0.4)(x-0.55)(x-0.65) + \\ + 0.034(x-0.4)(x-0.55)(x-0.65)(x-0.80).$$

因此

$$f(0.596) \doteq p_4(0.596).$$

(4) 补充 $(\bar{x} - x_k)$ 列, ($k=0, 1, 2, 3$):

$\therefore p_4(0.596) = p_4(\bar{x})$ 的计算中, 需要用到 $\bar{x} - x_k$ ($k=0, 1, 2, 3$)。因此将 $\bar{x} - x_k$ 写在均差表的最后一列。

(5) 计算 $p_4(0.596)$:

为了充分利用中间结果以节省运算时间及机器内存, 我们对多项式的计算采用下列程

序:

为了确切起见以一四次多项式为例阐明如下: (其思想适用于一般多项式)。

设 $F(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$

则可改写为:

$$F(x) = \{[(a_4x + a_3)x + a_2]x + a_1\}x + a_0.$$

仿此, 我们可将计算 $p_4(\bar{x})$ 的程序改写为:

$$\begin{aligned} p_4(\bar{x}) &= \left(\{ [0.034(\bar{x} - 0.80) + 0.197](\bar{x} - 0.65) + 0.28 \}(\bar{x} - 0.55) \right. \\ &\quad \left. + 1.116 \right)(\bar{x} - 0.40) + 0.41075 \\ &= \left(\{ [(0.034)(-0.204) + 0.197](-0.054) + 0.28 \}(0.046) \right. \\ &\quad \left. + 1.116(0.196) + 0.41075 \right) \\ &= 0.63192. \end{aligned}$$

§ 1—4 拉格兰日插值多项式

1. 问题的提出:

上节讨论的均差插值多项式, 虽然计算很有规律, 实际计算中也常用, 但有一个缺陷, 即不够直观,

兹介绍另一种形式的插值多项式,

已知: $y = f(x)$ 在 $n+1$ 个点 $x_0, x_1, \dots, x_n$ 上的值为 $y_0, y_1, \dots, y_n$. 试求一次数不高于 n 的插值多项式 $L_n(x)$ 使得

$$L_n(x_j) = y_j, \quad (j=0, 1, \dots, n) \quad (1.11)$$

我们希望对于新的形式的插值多项式能一眼看出条件(1.11)成立。

2. 拉格兰日插值多项式的构造:

不难看出, 如果能将均差插值多项式(1.9)中的系数(用均差表示的), 换为用已给参数 x_j, y_j 来表示, 则可得出一种新的形式的插值多项式, 而这是能够办得到的。回顾一下前述讨论可立知(1.5)式即可起到这个作用。

事实上, 由(1.9)式有:

$$\begin{aligned} p_n(x) &= f(x_0) + f(x_0, x_1)(x-x_0) + \dots \\ &\quad + f(x_0, x_1, \dots, x_n)(x-x_0)(x-x_1)\dots(x-x_{n-1}). \end{aligned}$$

而据(1.5)有:

$$f(x_0, x_1, \dots, x_n) = \sum_{j=0}^n \frac{1}{(x_j - x_0)(x_j - x_1)\dots(x_j - x_{j-1})(x_j - x_{j+1})\dots(x_j - x_n)} y_j.$$

用(1.5)代入(1.9)经过计算和整理, 可得下列(1.12)式:

$$L_n(x) = \sum_{j=0}^n \frac{(x-x_0)(x-x_1)\dots(x-x_{j-1})(x-x_{j+1})\dots(x-x_n)}{(x_j-x_0)(x_j-x_1)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_n)} y_j \quad (1.12)$$

式中引用符号 $L_n(x)$ 是为了与均差插值多项式加以区别。