引　论

§1 算法设计思想引论

1 科学计算离不开算法设计

算法：计算机上使用的计算方法。

例：Crammer 法则解线性方程组，当 n=20 时，求 n+1 个行列式的值，共做(n+1)n!(n-1)次乘除操作，大约 10^21 次乘除操作，由每秒 3 千亿次的计算机承担运算需要如下时间：

$$\frac{10^{21}}{3\times10^{11}\times60\times24\times365}\approx100\text{（年）}$$

然而用消元法，使用普通的计算器也能轻易解决。

结论：人类的计算能力是计算机的研制能力与算法设计能力两者的结合；高效算法的设计是高性能计算的灵魂。

2 算法设计要有“智类之明”

三千多年前上古先贤陈子说：“算法之术，是用智矣!”

陈又说：“夫道术，言约而用博者，智类之明。问一类而以万事达者，谓之知道。”

即为：算法设计要用智慧。

算法设计的基本技术讲起来很简单，但应用却很广泛；要学好算法，关键在于将各色各样的具体算法进行归纳分类，并能融会贯通。只要掌握了基本算法设计技术，就能设计出许许多多具体算法。

3 数学思维的化归策略

数学被誉为思维的体操，数学思维是算法设计的基础。

数学家的思维特征：他们往往不是对问题进行正面的“攻击”，而是不断地

将问题加工变形，直到把它转化归纳为能够解决的问题。

化归策略：就是转化归纳的策略。

迪卡尔提出的万能法则：

（1）将实际问题化归为数学问题；

（2）将数学问题化归为代数问题；

（3）将代数问题化归为解方程。

数值算法设计基于化归策略提出三种基本的算法设计技术：

（1）化大为小的缩减技术

（2）化难为易的校正技术

（3）化粗为精的松弛技术

§2 化大为小的缩减技术

1 Zeno 悖论的启示

古希腊的 Zeno 在 2000 多年前提出一个耸人听闻的命题，一个人不管跑得多快，也永远追赶不上爬在他前面的一只乌龟。这就是著名的 Zeno 悖论。耐人寻味的是：Zeno 悖论的论断及其荒谬，但从算法设计思想的角度来看却是极为精辟的。

追的过程：龟不动，人追上龟的时间 $\Delta t_k = \frac{S_{k-1}}{V}$

赶的过程：人不动，龟在这段时间爬的路程 $S_k = v\Delta t_k = \frac{v}{V} S_{k-1}$

经过追和赶问题规模压缩了 $\frac{v}{V}$ 倍

Zeno 算法悖论的算法描述：

$$\begin{cases} S_k = \frac{v}{V} S_{k-1} & , \quad k = 1,2,... \\ S_0 = S \end{cases}$$

该种“化大为小”的设计策略称作规模缩减技术。

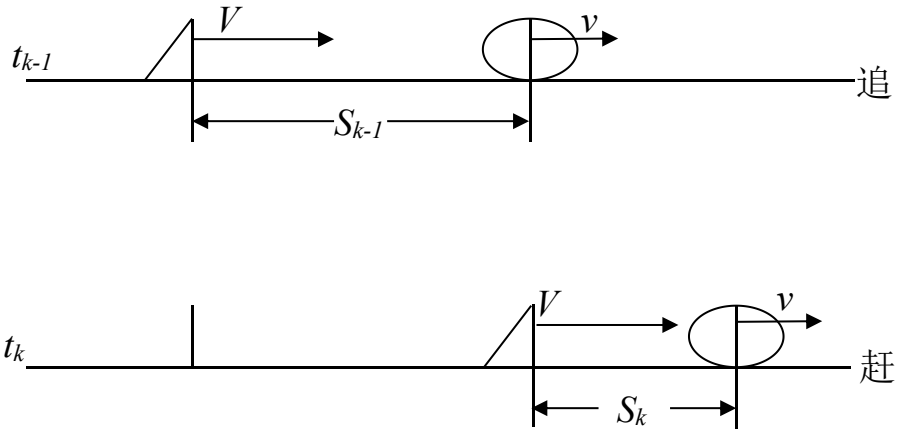


图 2-1　人龟追赶过程

2 数列累加求和

$$S = a_0 + a_1 + \ldots + a_n$$

算法描述：

$$\begin{cases} b_0 = a_0 \\ b_k = b_{k-1} + a_k \quad , \quad k = 1,2,...,n \end{cases}$$

重要特征：具有多层嵌套结构。

$$S = (...((a_0 + a_1) + a_2) + ... + a_{n-1}) + a_n$$

因此可以用嵌套结构的层数来表达累加求和的规模。

3 缩减技术的设计思想

引进某个实数---刻划问题的规模（及问题的大小）；

问题的解---为规模足够小的退化情形；

缩减技术的设计思想---大事化小，小事化了。

大事化小：（1）结构递归；（2）规模递减。

小事化了：规模足够小则为解。

4 多项式求和的秦九韶算法

$$P = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n$$

逐项生成算法耗费的乘法次数：

$$Q=\sum_{k=0}^{n}(n-k)\approx n^2$$

n 为 P 的次数定为规模

$$P=(a_0x+a_1)x^{n-1}+\sum_{k=2}^{n}a_kx^{n-k}$$

令 $v_1=a_0x+a_1$

$$P=v_1x^{n-1}+\sum_{k=2}^{n}a_kx^{n-k}$$

（规模减少 1 次）

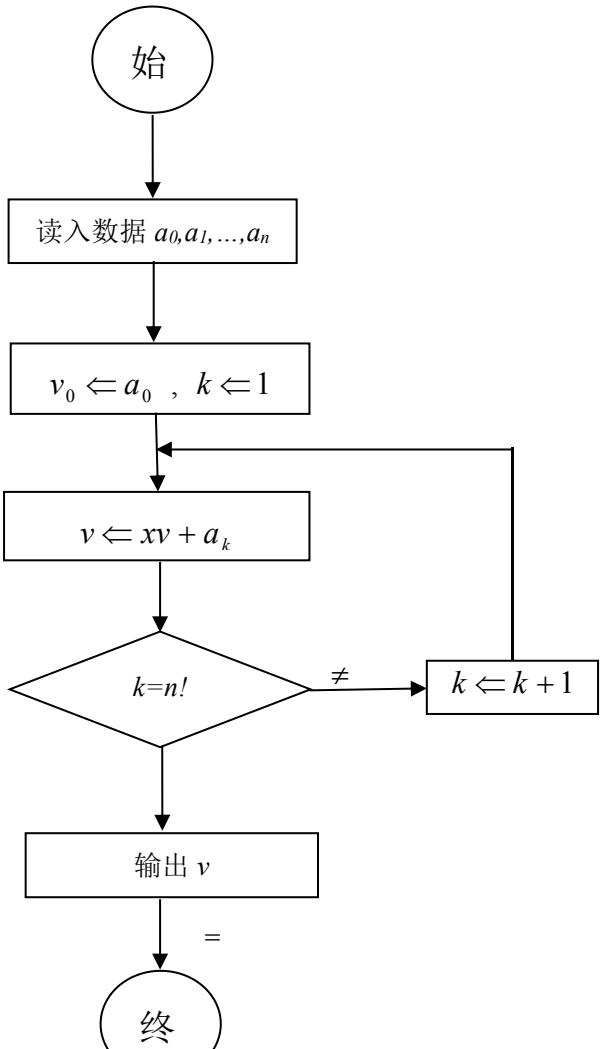
算法：令 $v_0=a_0$，对 $k=1,3,...,n$ 执行 $v_k=xv_{k-1}+a_k$

则结果：$P=v_n$ 为 $P=a_0x^n+a_1x^{n-1}+\ldots+a_{n-1}x+a_n$ 的值。

秦九韶算法的多重嵌套结构：

$$P=(...((a_0x+a_1)x+a_2)x+...+a_{n-1})x+a_n$$

5 计算流程图：



§3 化难为易的校正技术

1 Zeno 悖论中的“Zeno”钟

由 $Vt - vt = S$ 易得出人追上龟的实际时间：$t^* = \frac{S}{V - v}$

下面用预报校正技术处理这个简单问题，为将来求解一般非线性问题做准备。

设 t_0 位 t^*的某个预报值，希望提供校正量 Δt，使校正值 t_1 为：

$t_1 = t_0 + \Delta t$

$\Rightarrow V(t_0 + \Delta t) - v(t_0 + \Delta t) \approx S$

$\Rightarrow V(t_0 + \Delta t) - vt_0 \approx S$　　（v 是小量，Δt 也是小量）

$\Rightarrow$校正值：$t_1 \approx \dfrac{S + vt_0}{V}$

$\Rightarrow$迭代公式：$t_k = \dfrac{S + vt_{k-1}}{V}$　　，　$k=0,1,2,\ldots$

易知：$t_k \xrightarrow{k\to\infty} t^*$

这里，t_k 为日常钟，k 为 Zeno 钟。

2 求开方的迭代公式

给定 $a > 0$，求 $\sqrt{a}$ 的问题归结为解非线性方程 $x^2 - a = 0$。

设给定某个预报值 x_0，和校正量 Δx，则

$(x_0 + \Delta x)^2 \approx a$

$\Rightarrow x_0^2 + 2x_0\Delta x \approx a \Rightarrow \Delta x \approx \dfrac{a - x_0^2}{2x_0}$

所以，校正值 x_1 为：

$$x_1 = x_0 + \Delta x = x_0 + \frac{a}{2x_0} - \frac{x_0}{2} = \frac{1}{2}\left(x_0 + \frac{a}{x_0}\right)$$

导出开方的迭代公式：

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right)$$

直到 $|x_{k+1} - x_k| < \varepsilon$ 为止（ε 为给定精度）

例：求 $\sqrt{2}$

解：　取 $x_0 = 1$，迭代 5 次，可得到 $\varepsilon = 10^{-6}$ 的结果 1.414214。

k	0	1	2	3	4	5

x_k	1	1.500000	1.466667	1.414216	1.414214	1.414214

几何解释：由 $x_k \approx \sqrt{a} \Rightarrow \frac{a}{x_k} \approx \sqrt{a}$

易知 x_k 和 $\frac{a}{x_k}$ 的算术平均值是更好的近似值；

实际上 $\sqrt{x_k \frac{a}{x_k}} = a$ 为 x_k 与 $\frac{a}{x_k}$ 的几何平均值；

开方的迭代公式可理解为两个近似值的算术平均值逐步逼近它的几何平均值 $\sqrt{a}$

3 校正技术的设计思想

以简驭繁---将复杂计算化归为一系列简单计算的重复；

迭代法---简单的重复生成复杂，它的设计思想为：删繁就简，逐步求精。

校正方程的基本要求：

（1）逼近性

（2）简单性

递推化，迭代“逐步求精”

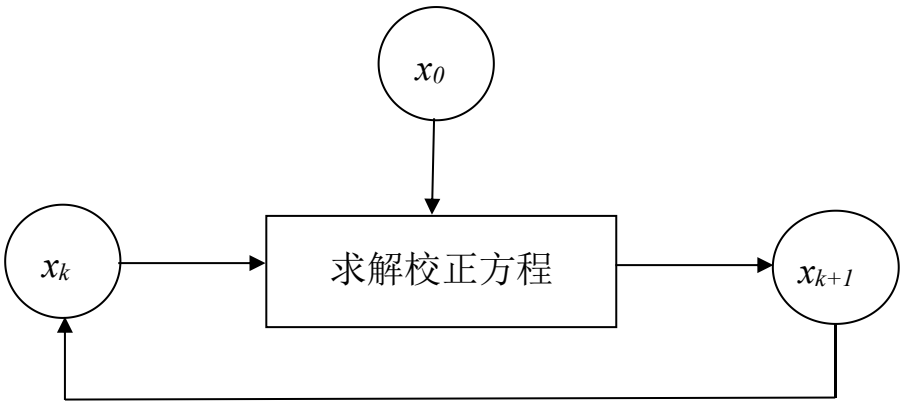


图 3-1　迭代法的逐步求精过程

§4　化粗为精的松弛技术

1 Zeno 算法的精华

考察 Zeno 算法$t_k=\frac{S+vt_{k-1}}{V}$，对于给定的预报值t_0，有$t_1=\frac{S+vt_0}{V}$

$\Rightarrow Vt_1-vt_0=S$（两端同除以$V-v$）

$\Rightarrow \frac{V}{V-v}t_1-\frac{v}{V-v}t_0=\frac{S}{V-v}=t^*$（精确解）

所以，$t^*=(1+\omega)t_1-\omega t_0$（为$t_0$和$t_1$的加权平均，其中$\omega=\frac{v}{V-v}$）

意义，将任意一对迭代值t_0和t_1，按上面固定方程式进行松弛，总可以获得所给方程的精确解t^*。

这种加工效果是奇妙的。

2 千古绝技“割圆术”

古希腊的阿基米德，公元前 3 世纪，利用内接与外切正 96 边形逼近圆周，得到公元前π的最好近似值 3.14；

魏晋大数学家刘徽，公元 3 世纪，提出“割圆术”，利用内接正 3072 边形逼近面积，得到当时π的最好近似值 3.1416。

刘徽不但提供了一种π的迭代算法，还给出一个加速逼近公式，此即为刘徽“割圆术”中最为精彩的部分。

加速逼近公式为：

$$\hat{S}=S_{192}+\frac{36}{105}(S_{192}-S_{96})=314\frac{4}{25}\approx S_{3072}$$

这里利用了$S_{96}=313\frac{584}{625}$与$S_{192}=314\frac{64}{625}$这两个粗糙的数据进行松弛，可以获得$\pi=3.1416$，从而实现$\pi$计算的一次革命性飞跃。

1700 多年后的 20 世纪，西方数学家才基于所谓余项展开式探讨逼近过程的加速问题。

3 求倒数的迭代算法

由预报值x_0，获得与之相伴随的另一个近似值$\bar{x}_0$，对x_0和$\bar{x}_0$适当加权平均，获得较高精度的改进值x_1：

$$x_1 = (1-\omega)x_0 + \omega\bar{x}_0$$

例：计算$x^* = \frac{1}{a}$

解：设$x_0 \approx \frac{1}{a} \Rightarrow ax_0 \approx 1$，则可得$\bar{x}_0$：

$$\bar{x}_0 = ax_0 \cdot x_0 = ax_0^2 \approx \frac{1}{a}$$

注意：$x_0 > x^*$时，$\bar{x}_0 > x_0$；$x_0 < x^*$时，$\bar{x}_0 < x_0$。即x_0和$\bar{x}_0$在x^*的同一侧。

由$ax_0 \approx 1$有

$$\bar{x}_0 - x_0 = ax_0^2 - x_0 = ax_0\left(x_0 - \frac{1}{a}\right) \approx x_0 - \frac{1}{a} = x_0 - x^*$$

$$\Rightarrow x^* \approx 2x_0 - \bar{x}_0$$

由此得到迭代公式：$x_{k+1} = 2x_k - ax_k^2$　　$k=0,1,2,...$

其中，x_k为优，ax_k^2为劣，　2、1 为系数，这种优劣互补的加工方式显著地改善了精度。

4 松弛技术的设计思想

$$\hat{F} = (1-\omega)F_0 + \omega F_1 = F_0 + \omega(F_1 - F_0)$$

选取系数来调整与松动校正量，称之为松弛技术。

$$\hat{F} = (1+\omega)F_1 + \omega F_0 \quad , \quad \omega > 0$$

亦即，张扬F_1，抑制F_0，这种设计策略称作为外推松弛技术，简称超松弛。

此法的设计机理为：优劣互补，化粗为精。

提高算法精度的效果，关键在于松弛因子的选取，它往往是困难的。

前面刘徽的割圆术已蕴涵着这种大智慧。