数值分析

李治平

北京大学
数学科学学院

引言

- 计算已经成为与理论分析和实验并重的科学与工程领域研究与应用的重要手段。
- 广义的科学计算包含
  - 数学建模；
  - 算法设计、实现与分析（狭义的科学计算）；
  - 数值结果的分析与应用。
- 计算方法与计算机技术的发展对科学工程问题计算能力的提高具有同等重要的意义。
- 计算方法的评判标准—快、省、准（实践是检验真理的唯一标准）。

本课程的主要内容概要

- 函数的插值与逼近—对复杂函数做近似数值计算的基础。用来做插值和逼近的函数要简单，便于计算且有较好的逼近等分析性质。例如，多项式函数、有理函数、三角函数等。
- 数值微分与数值积分—微积分是最基本的数学运算，相应的算法是进行科学计算的基本工具。
- 非线性方程（组）的数值解法—科学与工程中的大量非线性问题在数值计算时都离不开非线性方程（组）的数值求解。
- 快速 Fourier 变换—Fourier 变换在三角函数的插值与逼近、微分和积分方程的数值求解等方面有广泛应用。
- 常微分方程数值方法—数值求解微分方程是科学工程计算中最重要的问题之一。
- Monte Carlo 方法—对自由度极高的问题，确定性方法有本质的困难，这时随机性算法起到了不可替代的重要作用。

模型误差与观测误差

- 模型误差：数学模型的真解与实际问题的真解之差。
- 广义的模型误差还应该考虑
  - 数学模型的适定性；
  - 数学模型的真解能否反映实际问题的真解的某些重要性质；
  - 数学模型是否可以有足够好的近似解。
- 观测误差：原始数据的误差。观测误差一般是无法避免的，往往可以控制在一定的范围内。

截断误差与舍入误差

- 截断误差：数学模型的真解通常无法通过有限次运算获得。为此，需要设计能够通过有限次运算得到结果的离散模型（算法）来获得近似解。由此产生的离散模型与连续模型间的误差称为截断误差。
- 舍入误差：计算机的字长有限，数值计算结果一般只能按某种指定的方式舍入。例如在一定有效位后的四舍五入。
- 对误差的理解、分析和控制是计算科学最基本的内容之一。

绝对误差与绝对误差限

设 x 为某量的精确值，$\tilde{x}$ 为其近似值。

- 绝对误差：$e(x) = |x - \tilde{x}|$.
- 绝对误差限 ε：若 $|x - \tilde{x}| \leq \varepsilon$.
- 记号 $x = \tilde{x} \pm \varepsilon \quad \Leftrightarrow \quad \tilde{x} - \varepsilon \leq x \leq \tilde{x} + \varepsilon$.

相对误差与相对误差限

设 x 为某非零量的精确值，$\tilde{x}$ 为其近似值。

- 相对误差：$e_r(x) = \dfrac{e(x)}{|x|} = \dfrac{|x-\tilde{x}|}{|x|}$.
- 相对误差限 ε_r : 若 $\dfrac{|x-\tilde{x}|}{|x|} \le \varepsilon_r$. $\quad (\varepsilon = \varepsilon_r|x|)$.
- 当 x 未知时，常用 $\dfrac{|x-\tilde{x}|}{|\tilde{x}|}$ 表示相对误差, 并用 $x = \tilde{x}(1 \pm \varepsilon_r)$ 表示 $\tilde{x}(1-\varepsilon_r) \le x \le \tilde{x}(1+\varepsilon_r)$.

有效数字

有效数字是另一种更直观地反映近似值精度的概念。

- 将 $\tilde{x}$ 写成规范形式 $\tilde{x} = 0.a_1a_2\cdots a_i\cdots \times 10^m$, 其中$m$ 为整数， $a_i \in \{0, 1, \cdots, 9\}$, $a_1 \neq 0$。
- 有效数字: 若 $|x - \tilde{x}| \leq \frac{1}{2} \times 10^{m-n}$, 则称 $\tilde{x}$ 有 n 位有效数字， a_1, a_2, $\cdots$, a_n, 为其(n 位) 有效数字。
- 由于 $0.1 \leq 0.a_1a_2\cdots a_i\cdots < 1$, 所以 $\tilde{x}$ 有 n 位有效数字相当于其相对误差限 $\varepsilon_r \in [0.5 \times 10^{-n}, 5 \times 10^{-n})$。
- 由精确值按四舍五入原则截取得到的近似值，其每一位数字都是有效数字。

由数据误差估计运算结果的误差

设 y 由数据 $x_1, \cdots, x_n$ 通过光滑运算 $\varphi(x_1, \cdots, x_n)$ 得到。

- 设 $x_1, \cdots, x_n$ 的近似值为 $\tilde{x}_1, \cdots, \tilde{x}_n$.
- 则运算结果的绝对误差和相对误差分别为:
  $|y - \tilde{y}| = |\varphi(x_1, \cdots, x_n) - \varphi(\tilde{x}_1, \cdots, \tilde{x}_n)|,$
  $$\frac{|y - \tilde{y}|}{|y|} = \frac{|\varphi(x_1, \cdots, x_n) - \varphi(\tilde{x}_1, \cdots, \tilde{x}_n)|}{|\varphi(x_1, \cdots, x_n)|}.$$
- 由此得运算结果的误差估计: $|y - \tilde{y}| \leq \sum_{i=1}^{n} \left|\frac{\partial \varphi}{\partial x_i}\right| |x_i - \tilde{x}_i|,$
  $$\frac{|y - \tilde{y}|}{|y|} \leq \sum_{i=1}^{n} \left|\frac{x_i}{y}\frac{\partial \varphi}{\partial x_i}\right| \frac{|x_i - \tilde{x}_i|}{|x_i|} \text{ (设 } y, x_i, 1 \leq i \leq n \text{ 均非零)}.$$

数据误差在运算结果误差中的放大（缩小）系数

设 y 由数据 $x_1, \cdots, x_n$ 通过光滑运算 $\varphi(x_1, \cdots, x_n)$ 得到，则

- 数据的绝对误差在运算结果的绝对误差中分别被放大（缩小） $\left|\dfrac{\partial \varphi}{\partial x_i}\right|$ 倍.
- 数据的相对误差在运算结果的相对误差中分别被放大（缩小） $\left|\dfrac{x_i}{y}\dfrac{\partial \varphi}{\partial x_i}\right|$ 倍.

四则运算结果的误差限与数据误差限的关系

两个数 a 和 b 的四则运算是最简单的二元光滑运算。由以上误差分析得：

- 运算结果的绝对误差：$e(a \pm b) \leq e(a) + e(b)$,
  $e(ab) \leq |b|e(a) + |a|e(b)$, $e(a/b) \leq e(a)/|b| + |a/b^2|e(b)$.
- 运算结果的相对误差:

$$e_r(a \pm b) \leq \frac{e(a) + e(b)}{|a \pm b|} \leq \frac{|a|e_r(a) + |b|e_r(b)}{|a \pm b|},$$

$$e_r(ab) \leq e_r(a) + e_r(b),$$

$$e_r(a/b) \leq e(a/b)/|a/b| \leq e_r(a) + e_r(b).$$

二进制浮点数系统

1. 二进制浮点数系统 $\mathbb{F}(2,t,L,U)$ 定义为

$$\mathbb{F}=\left\{\pm 0.d_1d_2\cdots d_t\times 2^m : d_1=1, d_j\in\{0,1\}, 2\le j\le t, L\le m\le U\right\}\cup\{0\},$$

其中 t 为一给定自然数，称为字长；L, U 为给定的整数，m 为介于 L, U 之间的任意整数。

2. 在每一台计算机上，实数系 $\mathbb{R}$ 都是用一定的浮点数系统来近似的。例如，对 $x=\pm 0.1d_2\cdots d_td_{t+1}\cdots\times 2^m$, 可通过截断或舍入将其在 $\mathbb{F}(2,t,L,U)$ 中表示为 $fl(x)$.

3. 此时有 $\left|\dfrac{fl(x)-x}{x}\right|\le 2^{-t}$. 通常将 β^{-t} 称为 β 进制、字长为 t 的计算机的机器精度, 常记为 ε_{mach}。

β 进制浮点数系统的性质

- 它是只含实数系 $\mathbb{R}$ 中 $2\beta^{t-1}(U-L+1)+1$ 个数的有限集;
- 其中非零的数关于原点对称地非均匀地分布于 $[UFL, OFL]$ 和 $[-OFL, -UFL]$ 中$(UFL = 0.1\beta^L,\ OFL = (1-\beta^{-t})\beta^U)$;
- m 越小，数系分布越密，m 越大，数系分布越稀疏。当 $m = L$ 时， 浮点数间隔为 β^{L-t}, 而当 $m = U$ 时， 浮点数间隔为 β^{U-t}.
- 浮点数系中四则运算不满足实数系中的法则（如结合律）.
- 数学上等价的公式在浮点数系中的计算结果可能是不同的.

计算复杂度—运算量与问题规模的关系

- 通常可将算法大致分为两类：
  - 直接法：在没有误差的情况下，可在有限步计算出问题精确解的算法；
  - 迭代法：采取逐次逼近的方法来逼近问题的精确解，一般在任意有限步内都不能得到精确解的算法。
- 算法的计算复杂度是指其总的运算量，既所有加、减、乘、除运算的总次数(通常只精确到问题规模 n 的最高次幂)。
- 例如，对于一个有 n 个变量的问题，一个直接法需要通过 $3n^3 + 20n^2 + 50$ 次加、减、乘、除运算得到结果，则称该算法的运算量为 $3n^3$.
- 迭代法的总运算量不仅依赖于其每步的运算量，还依赖于其收敛速度和对计算结果精度的要求.

迭代法的收敛速度

- 设某迭代法产生的序列 $\{x_k\}$ 收敛于 x, 且存在正数 $r \in [1, \infty)$ 和正数 $c_r \in (0, \infty)$ 使得

$$\lim_{k\to\infty} \frac{\|x_k - x\|}{\|x_{k-1} - x\|^r} = c_r,$$

  则称该算法是 r 阶收敛的。

- 特别地，当 $r = 1$, 且 $0 < c_r < 1$时, 称相应算法为线性收敛的.

- 注意, 当 $r = 1$ 时，若 $c_r > 1$, 则序列显然不收敛于 x; 但不能除外 $c_r = 1$. 例如: 若 $\|x_k - x\| = (1 - \frac{1}{k})\|x_{k-1} - x\|$, 则有 $\lim_{k\to\infty} \|x_k - x\| = \lim_{k\to\infty} \frac{1}{k}\|x_1 - x\| = 0$, 序列收敛; 但若 $\|x_k - x\| = (1 - \frac{1}{k^2})\|x_{k-1} - x\|$, 则序列一般不收敛, 因为 $\lim_{k\to\infty} \|x_k - x\| = \lim_{k\to\infty} \frac{k+1}{2k}\|x_1 - x\| = \frac{1}{2}\|x_1 - x\|$.

迭代法的超线性收敛

- 当 $r = 2$ 时，也称该算法为平方收敛的，或二次收敛的。
- 当 $r = 3$ 时，也称该算法为立方收敛的，或三次收敛的。
- 又若 $\lim\limits_{k\to\infty} \dfrac{\|x_k - x\|}{\|x_{k-1} - x\|} = 0$, 则称该算法为超线性收敛的。显然，当 $r > 1$ 时， r 阶收敛的算法是超线性收敛的。

思考题：为什么只对 $r \geq 1$ 定义收敛阶？

敏度与敏度分析

由于数据一般总会有误差，因此，即便所有的运算都是精确的，问题的解也会有误差。敏度是指精确运算得到的解对微小的数据误差的敏感程度。敏度分析通常用计算问题的条件数来实现。

1. 函数 $f(x)$ 的条件数：设 $x \neq 0, f(x) \neq 0$, 如果可以找到 (尽可能小的) 正数 $c(x)$ 使得
$$\frac{|f(x+\delta x)-f(x)|}{|f(x)|} \leq c(x)\frac{|\delta x|}{|x|}, \quad \forall \frac{|\delta x|}{|x|} \ll 1,$$
则称 $c(x)$ 为 $f(x)$ 在 x 点的条件数。

2. $c(x)$ 很大时称 $f(x)$ 在 x 点是病态的；$c(x)$ 较小时称 $f(x)$ 在 x 点是良态的。

敏度与敏度分析

1. 敏度是问题的属性。例如，某物理现象（气象）本身对初始数据十分敏感，则相应的数学模型问题一般是病态的。
2. 对于一般的问题，要分析其敏度，或估计条件数，通常是相当困难的。
3. 但当 $f(x)$ 在 x 点可微时，有 $c(x) \approx \dfrac{|f'(x)||x|}{|f(x)|}$.

向后误差分析

由于运算一般总会有舍入误差，因此，即便所有数据都是精确的，运算的结果也会有误差。舍入误差对计算结果的影响程度是衡量一个算法优劣的重要指标之一。

1. 设用某种算法计算得到的函数 f 在 x 点值为 $\hat{y}$. 一般地说 $\hat{y} \neq f(x)$, 但往往存在 $\delta x \neq 0$ 使得 $\hat{y} = f(x + \delta x)$.

2. 向后误差分析: 分析给出一个与计算机精度和算法有关的正数 ε, 使得存在 $\delta x \neq 0$ 满足

$$\hat{y} = f(x + \delta x), \quad \text{且} \quad \frac{|\delta x|}{|x|} \leq \varepsilon.$$

数值稳定性

我们可以认为，ε 越小，舍入误差对算法结果的影响越小。

1. 当 ε 较小时，称算法是数值稳定的；当 ε 很大时，称算法是数值不稳定的。
2. 数值稳定性是算法的固有属性，它与问题的敏度是两个完全不同的概念。
3. 由问题的敏度和算法的数值稳定性可导出计算结果相对误差上界的估计：

$$\frac{|\hat{y} - f(x)|}{|f(x)|} = \frac{|f(x + \delta x) - f(x)|}{|f(x)|} \leq c(x)\frac{|\delta x|}{|x|} \leq c(x)\,\varepsilon.$$

误差、稳定性与计算可靠性

- 误差: 模型、数据、方法（离散）、计算（舍入）。
- 稳定性:
  1. 原始问题及数学模型的稳定性（敏度分析，病态、良态）；
  2. 算法稳定性（精确计算时，小数据误差 $\Rightarrow$ 小结果误差）；
  3. 数值稳定性（舍入误差对结果的影响，向后误差分析）。
- 由此可知，要得到可靠的计算结果，我们需要有相对
  1. 比较良态的原始问题及数学模型；
  2. 有较好的稳定性和数值稳定性的算法；
  3. 较小的数据误差和方法（离散）误差。

算法设计与软件使用

- 本课程旨在针对一些基本的问题分析构造有较小的方法（离散）误差和较好的稳定性的算法。对算法的数值稳定性也做了一定的讨论。值得指出的是, 避免大数吃小数等基本原则和尽量减少运算量对相应算法的数值稳定性有重要作用。
- 已有大量的成熟的数学软件，其中许多算法在设计和实现过程中很好地考虑了数值稳定性。

函数的多项式逼近的需要

- 在应用中，许多函数是待求的，我们只知道其某些信息，但却需要由此出发获取（近似地获取）其它更多的信息。
- 例如，已知一光滑函数在 x_0 点的函数值及其直至 k 阶的导数值，则在 x_0 的充分小邻域中有

$$P_k(x) = f(x_0) + \sum_{i=1}^{k} \frac{1}{i!} f^{(i)}(x_0)(x - x_0)^i \approx f(x).$$

- 可见多项式可以在一定条件下用来逼近充分光滑的函数。
- 多项式是有许多很好的分析和计算性质的函数。例如其运算简单运算量小，特别是其导函数和原函数仍然是多项式，且在不考虑舍入误差时可在计算机上精确表达和运算。

函数的多项式逼近的可行性

- 多项式逼近在理论上说是可行的。事实上，我们有 Weierstrass 定理(证明将在以后给出)。

定理： 设 $f(x)$ 是闭区间 $[a, b]$ 上的连续函数。则对任给的 $\varepsilon > 0$, 存在 $N(\varepsilon) > 0$, 只要 $n \geq N(\varepsilon) > 0$, 就可以找到 n 次多项式 $P_n(x)$ 使得

$$\|f - P_n\|_\infty = \|f - P_n\|_{\infty,[a,b]} \triangleq \max_{a \leq x \leq b} |f(x) - P_n(x)| < \varepsilon.$$

- 这说明我们可以用多项式在任给的精度内逼近连续函数。

函数的多项式逼近问题的提法

- 在应用中常会遇到这样的问题：已知一连续函数 $f(x)$ 在某些点(有限个或整个定义域)上的值，如何构造一多项式使其在一定意义下较好的逼近函数 $f(x)$.
- 一般地说，我们可以考虑以下问题：
  - 插值逼近：找一 k 次多项式使其取 f 在 $k+1$ 个点上的值.
  - 最小二乘逼近：找一 k 次多项式使其与 f 在 $k+m+1$ 个点上的 l^2-范数下的误差达到最小 (其中$m \geq 1$).
  - 最佳逼近：找一 k 次多项式使其在指定的范数下与 f 的误差达到最小（常用的范数包括 L^∞, L^2, 加权的 L^2 等）.

函数的插值多项式的定义

- 设已知区间 $[a, b]$ 上定义的函数 $y = f(x)$ 在该区间$n+1$ 个不同点 $x_0, x_1, \cdots, x_n$ 处的函数值

$$y_i = f(x_i), \quad i = 0, 1, 2, \cdots, n.$$

- 求一个定义在 $[a, b]$ 上的 n 次多项式 $P(x)$, 使其满足

$$P(x_i) = y_i, \quad i = 0, 1, 2, \cdots, n.$$

函数的多项式插值问题的提法

定义: 满足以上条件的多项式 $P(x)$ 称为是 $f(x)$ 的插值多项式，其中 $x_0, x_1, \cdots, x_n$ 称为插值节点， $f(x)$ 称为被插函数，$P(x_i) = y_i, \quad i = 0, 1, 2, \cdots, n$ 称为插值条件.

- 问题的适定性：满足以上定义的多项式是否存在唯一？问题是否良态？
- 能否分析给出算法的截断误差？
- 能否构造出有较好稳定性和数值稳定性的算法？

插值多项式的存在唯一定理

定理: 当 $n+1$ 个插值节点互不相同时, 必存在唯一的次数不超过 n 的多项式 $P(x)$ 满足插值条件 $P(x_i) = y_i,\ i = 0, 1, 2, \cdots, n.$

证明: 记 $P(x) = a_n x^n + \cdots + a_1 x + a_0$. 则有

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix}.$$

当 $n+1$ 个插值节点互不相同时, 该线性方程组的系数矩阵（Vandermonde 矩阵）是非奇异的，因此对任意给定的右端项存在唯一解。 ■

插值多项式问题是良态问题吗?

1. 对 $n = 1$, 由高斯消去法可得

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0}, \quad a_0 = y_0 - \frac{y_1 - y_0}{x_1 - x_0} x_0 = \frac{x_1 y_0 - x_0 y_1}{x_1 - x_0}.$$

2. 于是所得的多项式为

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

3. 当 $x \in [x_0, x_1]$ 时，由 $\left|\frac{\partial P}{\partial y_i}\right| \leq 1$, $\left|\frac{\partial P}{\partial x_i}\right| \leq \left|\frac{y_1 - y_0}{x_1 - x_0}\right|$, 知一次多项式插值问题是良态的.

插值多项式问题是良态问题吗?

4. 当 n 很小时，例如 $n = 1, 2, 3, 4, 5$，多项式插值问题一般是良态的.
5. 当 $n \gg 1$ 时, Vandermonde 矩阵常常是病态的。其病态程度强烈地依赖于 $\{x_i\}_{i=0}^n$ 的分布。
6. 因此，当 $n \gg 1$ 时， n 次多项式插值问题往往是非常病态的. 只有在适当的插值节点分布下问题才是不太病态的。
7. $n \gg 1$ 时， n 次多项式插值问题首先归结为寻找适当插值节点的问题。

如何构造数值稳定的插值多项式?

通过求解线性方程组

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix},$$

构造插值多项式一般并不是一个好办法. 除了运算量大之外，还有以下原因：

- 当换一组 $\{x_i\}_{i=0}^n$ 或 $\{y_i\}_{i=0}^n$ 后，需要重新求解方程组；
- 当增加一组数据 x_{n+1}, y_{n+1} 后，需要重新求解方程组。

总之，我们需要建立构造性的算法，这些算法应该具有较好的数值稳定性，并能部分地解决以上问题。

习题一：5，7，9　上机习题一: 1，　2

Thank You!