数值分析 Numerical Analysis

Lecture 1-2：误差分析(Error Analysis)

骞微著

苏州大学，计算机科学与技术学院

邮箱：wzqian@suda.edu.cn

办公室：理工楼543

基本概念Basic Concepts

1. 数值模型（迭代是计算的灵魂；迭代 == 循环？）
2. 误差分析
    1) 算法误差、舍入误差
    2) 相对误差、绝对误差
    3) 减小误差的方法
3. 收敛性 convergence和收敛速度 convergence rate
4. 相容性和稳定性
5. 计算成本（内存消耗、时间消耗、能否并行）

拓展：机器学习中迭代法的其他例子：优化过程、损失函数的设计。

误差分析Error Analysis

1. 算法误差/截断误差 Truncation Error:

例：用Taylor 展开计算$\int_0^1 e^{-x^2}\,dx$：

$$\int_0^1 e^{-x^2}\,dx = \int_0^1 \left(x - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} \cdots\right) dx$$

$$= \int_0^1 \left(x - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!}\right) dx + R$$

R是由近似方法造成的误差，即精确解与近似解之间的误差。

误差分析Error Analysis

2. 舍入误差 Rounding Error:

由于计算机字长有限，在实际计算时需要对数据采用近似表示：

$$\frac{1}{3} = 0.3333333\cdots$$

$$\pi = 3.1415926\cdots$$

3. 有效数字：

最小的有效数字位开始往前 数, 直至第一个非零数字为止。

例：已知精确值 $\pi = 3.14159265\cdots$, 则近似值 $x1 = 3.14$ 有 3 位有效数字, 近似值 $x2 = 3.1416$ 有 5 位有效数字, 近似值 $x3 = 3.1415$ 有 4 位有效数字。

误差分析Error Analysis

4. 绝对误差： $\tilde{x}$为x的近似值，则

$$\boldsymbol{\epsilon_r \triangleq \tilde{x} - x}$$

$\exists \varepsilon_r > 0$，s.t. $|\epsilon_r| \le \varepsilon_r$，则$\epsilon_r$为绝对对误差限。

5. 相对误差： $\tilde{x}$为x的近似值，则

$$\boldsymbol{\epsilon_r \triangleq \frac{\tilde{x} - x}{x}}$$

$\exists \varepsilon_r > 0$，s.t. $|\epsilon_r| \le \varepsilon_r$，则$\epsilon_r$为相对误差限

减小误差危害的方法

1. 避免相近的数相减： 如果两个相近的数相减，则会损失有效数字。例如 0.12346 – 0.12345 = 0.00001，两个操作数都有 5 位有效数字，但计算结果却只有 1 位有效数字。

解决方法： 各种等价公式来计算。

2. 避免数量级相差很大的数相除： 可能会产生溢出，即超出计算机所能表示的数的范围。特别需要注意的是，尽量不要用很小的 数作为除数，否则为放大分子的误差。

解决方法： 议把绝对值小的数作为分子或者取对数。

减小误差危害的方法

3. 避免大数吃小数： 如 （$10^9+10^{-9}-10^9/10^{-9}$），直接计算的话，结果为 0。

解决方法： 在对一组数求和时，建议按照绝对值从小到大求和。

4. 简化计算： 尽量减少运算次数，从而减少误差的积累.。

计算格式的相容性

定义1.1 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**。

例子 1:

数学模型：$y' = 2x$

计算格式：$\frac{y(a+h)-y(a)}{h} = 2a$

原因：$h \to 0$时，计算公式还原为$y' = 2x$的特例$y'(a) = 2a$。

计算格式的相容性

定义1.1 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**。

例子 2:

数学模型：$S_n = \sum_{k=0}^{n} \frac{1}{k!} x^k$

计算格式：$e^x = \sum_{k=0}^{\infty} \frac{1}{k!} x^k$

原因：$\lim\limits_{n \to \infty} S_n = \sum_{k=0}^{\infty} \frac{1}{k!} x^k$。

计算格式的相容性与稳定性

定义1.1 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**。

思考问题：

微分和积分哪一个相对来说好计算？

计算格式的稳定性

定义1.2 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**（简称为稳定），反之则为不稳定。

稳定性分析通常基于对初始误差的传播状况的讨论。

稳定性分析Stability Analysis

例： 试建立$I_n = \int_0^1 x^n e^{x-1}\,dx$的稳定计算格式。(教材pp. 4-5)

使用分部积分integration by parts:

$$I_n = x^n e^{x-1}|_0^1 - n\int_0^1 x^{n-1} e^{x-1}\,dx = 1 - nI_{n-1}$$

两种计算格式:

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

格式(B)：$I_{n-1} = \frac{1}{n}(1 - I_n), \mathrm{n} = 1, 2, \ldots$

推导。

稳定性分析Stability Analysis

注意到被积函数 $x^n e^{x-1}$ 在区间(0,1)内恒大于零，故得

性质 1 对任何 n，$I_n>0$

又由于 $I_n < \int_0^1 x^n e^{1-1} dx = \frac{1}{n+1}$

$$I_{n-1} = \frac{1-I_n}{n} > \frac{1}{n}\left(1-\frac{1}{n+1}\right) = \frac{1}{n+1}$$

$$I_n > \int_0^1 x^n e^{0-1} dx = \frac{e^{-1}}{n+1}$$

故得

性质 2 $\frac{e^{-1}}{n+1} < I_n < \frac{1}{n+1} < I_{n-1}$

进一步，由性质 2 易得

性质 3 $I_n \downarrow 0$（即 I_n 单调递减趋于零）

关于计算格式的初始值，可采用如下方法确定。对于格式(A)，我们可根据 $I_0 = \int_0^1 e^{x-1} dx = 1-e^{-1}$ 通过对 e^{-1} 的近似取值得到格式(A)的充分精确的近似值 $\tilde{I}_0$。

对于格式(B)，则可根据性质 2，近似取 $\tilde{I}_N = \frac{1}{2}\left(\frac{1}{N+1}+\frac{e^{-1}}{N+1}\right)$ 作为初始值 I_N 的近似值。这时初始误差 $|I_N-\tilde{I}_N| < \frac{1-e^{-1}}{2(N+1)}$。

稳定性分析Stability Analysis

将误差记为：$e_n = I_n - \tilde{I}_n$，$\tilde{I}_n$为初始值有误差时计算得到的近似值。

对于格式(A)：

$$\tilde{I}_n = 1 - n\tilde{I}_{n-1}$$

$$e_n = -ne_{n-1}$$

$$|e_n| = n!\,|e_0|$$

分析结果：误差随n迅速增长，计算不稳定。

稳定性分析Stability Analysis

对于格式(B)：

$$\tilde{I}_{n-1}=\frac{1}{n}(1-\tilde{I}_n)$$

$$e_{n-1}=-\frac{1}{n}e_n$$

$$|e_n|=\frac{1}{n+1}\frac{1}{n+2}\cdots\frac{1}{N}|e_N|, n<N$$

分析结果：误差随计算逐渐减小，计算稳定。

一般来说，如果一个计算格式满足如下误差关系

$$\left|e_{后}\right|\le \mathrm{C}\left|e_{初}\right|, \mathrm{C}为常数$$

则认为该计算格式数值稳定。

作业Homework 1

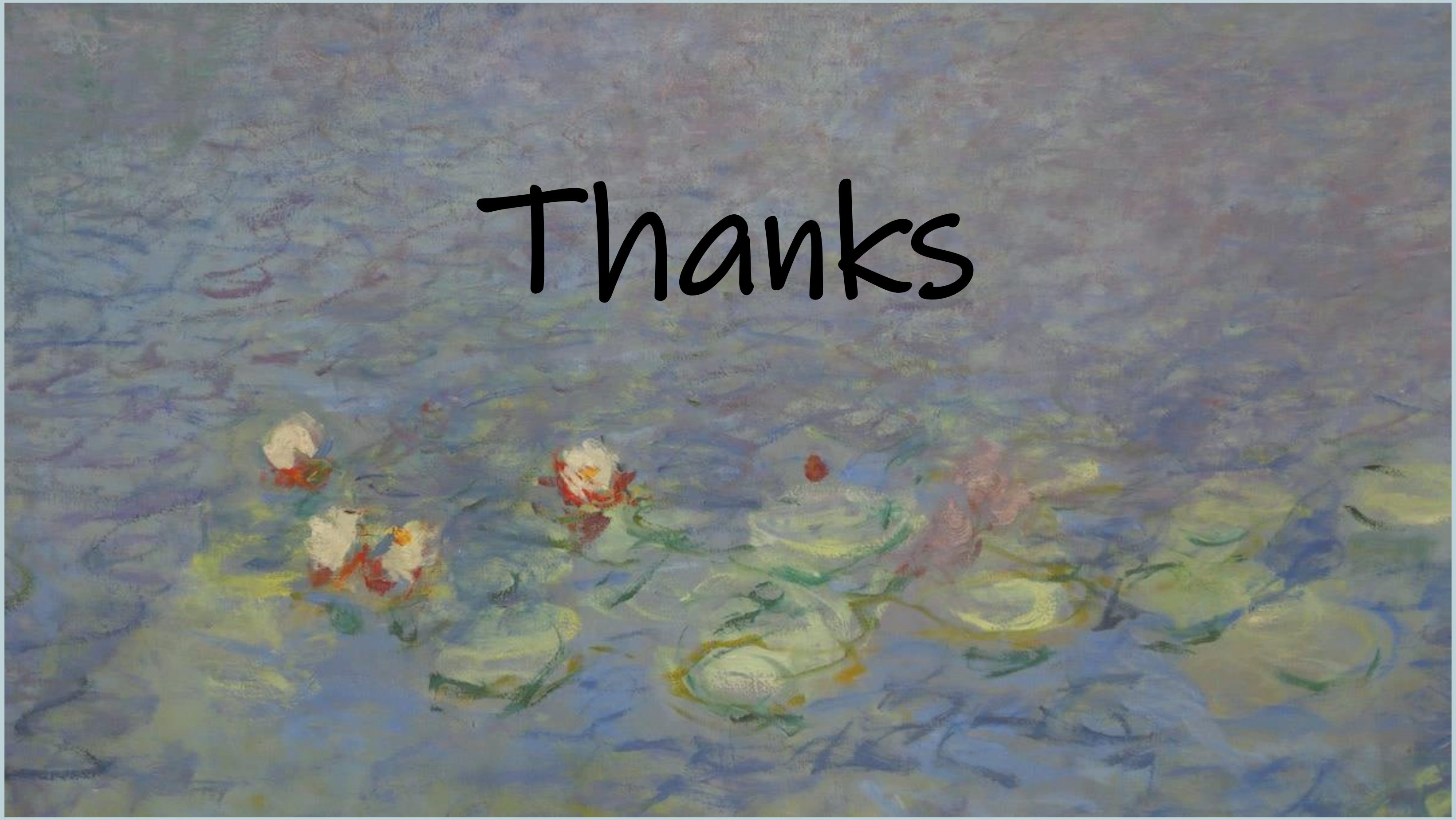
1. 给定计算格式 $y_{n+1}=(1-20h)y_n$ （$h=x_{n+1}-x_n$，y_n 为 $y(x_n)$ 的近似值）

(1)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。

作业以word格式将推导，代码，（可视化）运行结果等（控制文件大小，不要粘贴代码图片， 文件名：姓名_学号_作业1）发到助教（小沈）邮箱： 20245227049@stu.suda.edu.cn。

注意：在作业中自己编写函数，不要直接使用相关库中的现成函数。


Thanks