使用多种数值方法求e

数值分析与算法 课程项目

自54班 叶沁媛 2015011469

2017 年 12 月 25 日

摘要 # **TODO** wth? why do we even need to write abstract for a project report?

1 对e^x进行Taylor展开

1.1 算法描述

$f(x) = e^x$在$x = 0$处Taylor展开，得到：

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... + \frac{x^n}{n!} + ... \\ &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... + \frac{x^n}{n!} + e^{\xi} \frac{x^{n+1}}{(n+1)!}, \qquad \xi \in (0, x) \end{aligned} \tag{1}$$

本题中，求取$f(1) = e$，即$x = 1$，

$$e = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + ... + \frac{1}{n!} + ... \tag{2}$$

令$y_0 = 1$，使用如下的递推公式进行计算：

$$y_{n+1} = y_n + \frac{1}{n!} \tag{3}$$

1.2 误差分析

方法误差 方法误差为Taylor展开的余项

$$\begin{aligned} \Delta_n &= \frac{e^{\xi}}{(n+1)!}, \qquad \xi \in (0, 1) \\ &< \frac{e}{(n+1)!} \\ &< \frac{3}{(n+1)!} \end{aligned} \tag{4}$$

舍入误差 回顾式(2)中的迭代过程。在编写程序时，$n!$使用long long类型保存，无误差，因此只用考虑n次除法所产生的舍入误差，与$n-1$次加法所产生的舍入误差。故

$$\delta_n \leq (2n-1) \times \frac{1}{2} \times 10^{-m} \leq n \times 10^{-m} \tag{5}$$

总误差

$$r_n \le \Delta_n + \delta_n \le \frac{3}{(n+1)!} + n \times 10^{-m} \tag{6}$$

题目要求精度达到至少小数点后6位，即$r_n \le \frac{1}{2} \times 10^{-6}$。使用C++中的double类型，可以近似认为$m = 14$，此时取$n = 10$，$r_n \le 10^{-13} + \frac{3}{10!} \approx \frac{3}{10!} < \frac{1}{2} \times 10^{-6}$，符合要求。

1.3 计算代价

Code Snippet 1: Taylor展开求e，核心代码

```
for (int i=1;i<=n;i++){
    sum+=xpow/fac;
    xpow*=x;
    fac*=i+1;
}
```

其中xpow表示x的i次方，fac表示i的阶乘，在计算过程中使用了上一次循环的结果。每次循环中，加法1次，乘法2次，除法1次，总体运算量在$O(n)$量级。本题中，取$n = 10$。

2 求解常微分方程$y' = y$

2.1 常微分方程的设计

构造常微分方程

$$\begin{cases} y' = y \\ y(0) = 1 \end{cases} \tag{7}$$

该常微分方程的解为

$$y = e^x \tag{8}$$

$x = 1$时，$y(x) = e$，使用常微分方程数值解法求解$y(1)$即可得到e。

2.2 梯形公式

梯形公式是数值积分中常用的公式，一般写成如下的隐式形式：

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \tag{9}$$

本题中，由于$f(x_{n+1}, y_{n+1}) = y_{n+1}$，可以将梯形公式整理为显式形式：

$$y_{n+1} = \frac{2+h}{2-h} y_n \tag{10}$$

可以利用此式进行递推求解。

2.2.1 误差分析

方法累计误差

$$y(x_{n+1}) = y_n + hy'(x_n) + \frac{h^2}{2!}y^{(2)}(x_n) + \frac{h^3}{3!}y^{(3)}(x_n) + ... \tag{11}$$

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}[y'(x_n) + y'(x_{n+1})] \\ &= y_n + \frac{h}{2}[y'(x_n) + y'(x_n) + hy^{(2)}(x_n) + \frac{h^2}{2}y^{(3)}(x_n) + ...] \end{aligned} \tag{12}$$

$$\therefore y(x_{n+1}) - y_{n+1} = \frac{h^3}{12}y^{(3)}(x_n) = o(h^3) \tag{13}$$

因此，

$$\Delta_n \leq \frac{2+h}{2-h}\Delta_{n-1} + \frac{Th^3}{12} \tag{14}$$

其中，$y^{(3)}(x) \leq T$，$n = \frac{1}{h}$。令$a = \frac{2+h}{2-h}$、$b = \frac{Th^3}{12}$，从而

$$\Delta_{n+1} + \frac{b}{a-1} \leq a(\Delta_n + \frac{b}{a-1}) \leq ... \leq a^{n+1}(\Delta_0 + \frac{b}{a-1}) \tag{15}$$

$$\Delta_n \leq \frac{a^n - 1}{a-1}b = \frac{(\frac{2+h}{2-h})^n - 1}{\frac{2+h}{2-h} - 1} \times \frac{Th^3}{12} \tag{16}$$

其中，$\lim\limits_{h\to 0}(\frac{2+h}{2-h})^n = e^2$.由于$y^{(3)}(x) = e^x$，取$T = y^{(3)}(1) = e$。整理得到：

$$\Delta_n \leq \frac{(e^2-1)eh^2}{12} \tag{17}$$

由于求解时e的具体值未知，根据$e \leq 3$再度进行放大，得到

$$\Delta_n \leq 2h^2 \tag{18}$$

舍入累计误差

$$\delta_{n+1} \leq \frac{2+h}{2-h}\delta_n + \frac{1}{2} \times 10^{-m} \tag{19}$$

与方法累计误差的分析方法相类似，将不等式分解为等比数列，再根据$e \leq 3$进行放大，得到

$$\begin{aligned} \delta_n &\leq \frac{(\frac{2+h}{2-h})^n - 1}{\frac{2+h}{2-h} - 1} \times \frac{1}{2} \times 10^{-m} \\ &\leq \frac{10^{-m}}{4h} \end{aligned} \tag{20}$$

总误差

$$r_n \leq \Delta_n + \delta_n \leq 2h^2 + \frac{10^{-m}}{h} \tag{21}$$

题目要求精度达到至少小数点后6位。使用C++中的double类型，可以近似认为$m = 14$，此时取$h = 10^{-4}$，$r_n \leq 2 \times 10^{-8} + \frac{10^{-10}}{4} \approx 2 \times 10^{-8} \leq \frac{1}{2} \times 10^{-6}$，符合要求。

2.2.2 计算代价

Code Snippet 2: 梯形公式，核心代码

```
double co = (2+h)/(2-h);
for (int i=0;i<n;i++){
    y=co*y;
}
```

每次循环中使用一次乘法，计算代价为$O(n)$量级。本题中，由于取$h=10^{-4}$，故$n=10000$。

2.3 改进欧拉法

采用改进的欧拉法求解常微分方程，方法如下：

预测 $\bar{y}_{n+1}=y_n+hf(x_n,y_n)$

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

在本题中，$f(x_n,y_n)=y_n$，上式可以化简为

预测 $\bar{y}_{n+1}=y_n+hy_n$

校正 $y_{n+1}=y_n+\frac{h}{2}[y_n+\bar{y}_{n+1}]$

2.3.1 误差分析

预测部分采用**普通欧拉公式**：

$$y(x_{n+1})=y(x_n)+hy'(x_n)+\frac{h^2}{2!}y^{(2)}(\xi_n),\xi\in(x_n,x_{n+1}) \tag{22}$$

$$y_{n+1}=y_{x_n}+hy'(x_n) \tag{23}$$

$$\therefore y(x_{n+1})-y_{n+1}=\frac{h^2}{2!}y^{(2)}(\xi_n)=o(h^2) \tag{24}$$

校正部分采用**梯形公式**，推导方法与式(11)~(13)相同。

方法累计误差

$$\begin{cases}\bar{\Delta}_{n+1}\leq(1+h)\Delta_n+\frac{L}{2}h^2\\ \Delta_{n+1}\leq\Delta_n+\frac{h}{2}(\Delta_n+\bar{\Delta}_{n+1})+\frac{Th^3}{12}\end{cases} \tag{25}$$

整理得到：

$$\Delta_{n+1}\leq(\frac{h^2}{2}+h+1)\Delta_n+(\frac{L}{4}+\frac{T}{12})h^3 \tag{26}$$

其中$y^{(2)}(x)\leq M$、$y^{(3)}(x)\leq T$。令$a=\frac{h^2}{2}+h+1$、$b=\frac{L}{4}+\frac{T}{12}$，则式(26)可改写为：

$$\Delta_{n+1}+\frac{b}{a-1}\leq a(\Delta_n+\frac{b}{a-1})\leq ...\leq a^{n+1}(\Delta_0+\frac{b}{a-1}) \tag{27}$$

$$\Delta_n\leq\frac{a^n-1}{a-1}b=\frac{(1+h+\frac{h^2}{2})^n-1}{h+\frac{h^2}{2}}(\frac{L}{4}+\frac{T}{12})h^3 \tag{28}$$

其中，$\lim\limits_{h\to 0}(1+h+\frac{h^2}{2})^n=e$。由于$y^{(2)}(x)=e^x$，取$L=y^{(2)}(1)=e$，同理取$T=y^{(3)}(1)=e$。最终整理得到：

$$\Delta_n \le \frac{1}{3}e(e-1)h^2 \tag{29}$$

由于求解时e的具体值未知，根据$e\le 3$再度进行放大，得到

$$\Delta_n \le 2h^2 \tag{30}$$

舍入累计误差

$$\begin{cases} \bar{\delta}_{n+1} \le (1+h)\delta_n + \frac{1}{2}\times 10^{-m} \\ \delta_{n+1} \le \delta_n + \frac{h}{2}\delta_n + \frac{h}{2}\bar{\delta}_{n+1} + \frac{1}{2}\times 10^{-m} \end{cases} \tag{31}$$

整理得到：

$$\delta_{n+1} \le (1+h+\frac{h^2}{2})\delta_n + (\frac{h}{2}+1)\times\frac{1}{2}\times 10^{-m} \tag{32}$$

与方法累计误差的分析方法相类似，将不等式分解为等比数列，再根据$e\le 3$进行放大，得到：

$$\begin{aligned} \delta_{n+1} &\le \frac{(1+h+\frac{h^2}{2})^n-1}{h+\frac{h^2}{2}}(\frac{1}{2}+h)\times\frac{1}{2}\times 10^{-m} \\ &\le \frac{10^{-m}}{h} \end{aligned} \tag{33}$$

总误差

$$r_n \le \Delta_n + \delta_n \le 2h^2 + \frac{10^{-m}}{h} \tag{34}$$

题目要求精度达到至少小数点后6位。取$m=14$、$h=10^{-4}$，$r_n\le 2\times 10^{-8}+10^{-10}\approx 2\times 10^{-8}\le \frac{1}{2}\times 10^{-6}$，符合要求。

2.3.2 计算代价

Code Snippet 3: 改进欧拉法，核心代码

```
for (int i=0;i<n;i++){
    y_est=y+h*f(x,y);
    y=y+(h/2)*(f(x,y)+f(x+h,y_est));
    x=x+h;
}
```

所设计的常微分方程为$y'=y$，故本题中的$f(x,y)$函数无需额外的运算，直接返回y。在每次循环的过程中，共使用4次加法，2次乘法，1次除法。在循环n次的情况下，加减法的运算量为$4n$，乘除法的运算量为$3n$，总体的运算量在$O(n)$量级。本题中，由于取$h=10^{-4}$，故$n=10000$。

3 求解方程$\int_1^x \frac{1}{t}dt - 1 = 0$

3.1 算法简介

复化梯形公式求积分 $f(x) = \frac{1}{x}$。将区间$[1, x]$分为N个小区间，$h_N = \frac{x-1}{N}$，在这些小区间上做梯形公式积分。

$$\begin{aligned} I &= \sum_{k=0}^{N-1} \frac{h_N}{2}[f(x_k) + f(x_{k+1})] \\ &= \sum_{k=0}^{N-1} \frac{x-1}{2n}[\frac{1}{1+kh_N} + \frac{1}{1+(k+1)h_N}] \end{aligned} \tag{35}$$

牛顿迭代法求方程 $f(x) = \int_1^x \frac{1}{t}dt - 1$、$f'(x) = \frac{1}{x}$，使用下式进行迭代：

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad x_{n+1} = x_n(2 - \int_1^x \frac{1}{t}dt) \tag{36}$$

初值选择 选取初值区间$[2, 3]$。在此区间上，满足：

- $f(2)f(3) < 0$
- $f^{(2)}(x) = -\frac{1}{x^2}$在$[2, 3]$不变号
- $\forall x \in [2, 3] \qquad f'(x) = \frac{1}{x} \neq 0$
- $\frac{|f(2)|}{3-2} \leq f'(2)$

因此，$\forall x \in [2, 3]$，牛顿迭代法收敛。取$x_0 = 2$。

3.2 误差分析

3.2.1 复化梯形公式

假设将区间$[1, x]$分为N个小区间，则$h_N = \frac{x-1}{N}$。在第k个小区间上，使用梯形公式积分。

方法误差

$$R_k[f] = -\frac{h^3}{12}f^{(2)}(\eta_k), \qquad \eta_k \in [x_k, x_{k+1}] \tag{37}$$

$$R[f] = \sum_{k=0}^{n-1} R_k(f) \in [\min f^{(2)}(\eta), \max f^{(2)}(\eta)] \tag{38}$$

$$\because f^{(2)}(x) = -\frac{2}{x^3} \in C[2, 3] \tag{39}$$

$$\therefore \exists \eta \in [2, 3] \qquad s.t. \sum_{k=0}^{N-1} f^{(2)}(\eta_k) = nf^{(2)}(\eta) \tag{40}$$

$$R[f] = -\frac{(b-a)h_N^2}{12}f^{(2)}(\eta) \tag{41}$$

$$\because \left|f^{(2)}(\eta)\right| \leq \left|f^{(2)}(2)\right| = \frac{1}{4} \qquad\qquad \therefore |R[f]| \leq \frac{h_N^2}{48} \tag{42}$$

舍入误差 在N个小区间上分别做梯形公式积分，整体可以看做$2N$项相加而成。可以粗略地认为这$2N$项在计算时各产生了$\frac{1}{2}\times 10^{-m}$的舍入误差，故舍入误差$\delta \le N\times 10^{-m}$。

3.2.2 牛顿迭代法

分为方法误差和舍入误差两部分进行分析，

累计方法误差 回顾式(36)，假设x_n无误差，复化梯形积分产生了误差：

$$\Delta_{n+1}^{(1)} \le \frac{h_N^2}{48}\times x_n < \frac{h_N^2}{16} \tag{43}$$

假设复化梯形积分无误差，x_n有误差Δ_n：

$$\Delta_{n+1}^{(2)} = \phi'(\xi)\Delta_n = (1-\ln\xi)\Delta_n < (1-\ln 2)\Delta_n < 0.4\Delta_n \tag{44}$$

综合以上二式，总体方法误差

$$\Delta_{n+1} \le \Delta_n^{(1)} + \Delta_n^{(2)} \le \frac{h_N^2}{16} + 0.4\Delta_n \qquad \therefore \Delta_n \le \frac{1-0.4^n}{1-0.4}\frac{h_N^2}{16} \tag{45}$$

累计舍入误差 假设x_n无误差，复化梯形积分产生了舍入误差：

$$\delta_{n+1}^{(1)} \le x_n \times N\times 10^{-m} \le 3N\times 10^{-m} \tag{46}$$

假设复化梯形积分无误差，x_n有舍入误差δ_n：

$$\delta_{n+1}^{(2)} = \phi'(\xi)\delta_n = (1-\ln\xi)\delta_n < (1-\ln 2)\delta_n < 0.4\delta_n \tag{47}$$

综合以上二式，总体舍入误差

$$\begin{aligned}\delta_{n+1} &\le \delta_{n+1}^{(1)} + \delta_{n+1}^{(2)} + \frac{1}{2}\times 10^{-m} \\ &\le 0.4\delta_n + (3N+\frac{1}{2})\times 10^{-m}\end{aligned} \qquad \therefore \delta_n \le \frac{1-0.4^n}{1-0.4}(3N+\frac{1}{2})\times 10^{-m} \tag{48}$$

总误差

$$r_n \le \Delta_n + \delta_n \le \frac{1-0.4^n}{1-0.4}\left[\frac{h_N^2}{16} + (3N+\frac{1}{2})\times 10^{-m}\right] \tag{49}$$

根据下节的分析，如果不存在误差，则牛顿迭代法迭代4次就可以到符合精度要求的结果。取$n = 4, h_N = 10^{-4}(N = 10^4)$。于是

$$r_4 \le \frac{1-0.4^4}{1-0.4}\left[\frac{10^{-8}}{16} + (3\times 10^4 + \frac{1}{2})\times 10^{-14}\right] \approx 1.50\times 10^{-9} \tag{50}$$

配合$n = 4$时得到的

$$e_4 \le 4\times (0.2)^{2^4} \le 2.62\times 10^{-11} \tag{51}$$

两者之和小于$\frac{1}{2}\times 10^{-6}$，符合精确到小数点后6位的条件。

3.3 收敛速度

对于牛顿迭代法

$$e_{n+1} = \frac{1}{2}\phi^{(2)}(\xi_n)e_n^2 \leq \frac{M}{2}e_n^2 \tag{52}$$

$$e_n \leq \frac{2}{M}[\frac{M}{2}e_0]^{2^n}, \qquad M = \max\left|\phi^{(2)}(x)\right| \tag{53}$$

本题中，取初始值$x_0 = 2$，故$e_0 = e - 2 < 0.8$。同时，$\phi^{(2)}(x) = \frac{1}{x}$，故取$M = \frac{1}{2}$。为使$e_n < \frac{1}{2} \times 10^{-6}$，求得$n \geq 4$。
在编写程序时，使用误差分析的事后估计，即当$\frac{1}{1-L}\left|x_{k+1} - x_k\right| < \frac{1}{2} \times 10^{-6}$时，认为$x_k$为满足精度要求的解。取$L = 0.4 > \phi'(2)$。

3.4 计算代价

分析复化梯形积分的计算代价，将区间$[a, b]$分割成为n个小区间。

Code Snippet 4: 复化梯形积分，核心代码

```
for (int i=0;i<n;i++){
    double x=a+i*h;
    sum+=(h/2)*(f(x)+f(x+h));
}
```

每次调用$f(x) = \frac{1}{x}$时，使用了一次除法。每次循环中，计算x使用了一次加法，一次乘法；每次循环中，计算sum的增量使用了2次加法，1次乘法，3次除法。故总计使用$3n$次加法，$2n$次乘法，$3n$次除法。总计算量在$O(n)$量级。根据误差分析的估计，本题中取$n = 10000$。

Code Snippet 5: 牛顿迭代法，核心代码

```
while (true){
    x_before = x;
    x = x - f(x) / f_derivative(x);
    if (5.0/3.0*abs(x-x_before)<eps) break;
}
```

运行程序时，发现牛顿迭代法经过5次迭代后收敛。每次调用$f(x)$（梯形积分）需要8000次运算，调用$f_{derivative}(x)$（即$\frac{1}{x}$)的相应运算次数可忽略不计。总运算次数约为40000次。