《数值分析》之

常微分方程数值方法

徐岩

中国科学技术大学数学系

yxu@ustc.edu.cn

https://faculty.ustc.edu.cn/yxu

方法的要点是$y(x)$的Taylor级数展开：

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2!}y''(x) + \frac{h^3}{3!}y'''(x) + \frac{h^4}{4!}y^{(4)}(x) + \cdots$$

因此对于固定的x和h, 为了计算出$y(x+h)$的值，我们只需要知道在x点$y(x)$的各阶导数值

$$\begin{aligned} y'(x) &= f(x,y) \\ y''(x) &= f_x(x,y) + f_y(x,y)y'(x) \\ y'''(x) &= \cdots \end{aligned}$$

所以，可以构造格式

$$\begin{aligned} y(x_{n+1}) = y(x_n) &+ hf(x_n, y(x_n)) \\ &+ \frac{h^2}{2}(f_x(x_n, y(x_n)) + f_y(x_n, y(x_n))f(x_n, y(x_n)) \end{aligned}$$

为了应用Taylor级数方法，我们需要假定f的各阶偏导数存在，例如

$$\begin{cases} y' = \cos x - \sin y + x^2 \\ y(-1) = 3 \end{cases}$$

- 这些导数值可以从给定的微分方程和初值条件中得到：

$$\begin{aligned} y' &= \cos x - \sin y + x^2 \qquad (\text{已知条件}) \\ y'' &= -\sin x - y' \cos y + 2x \\ y''' &= -\cos x - y'' \cos y + (y')^2 \sin y + 2 \\ y^{(4)} &= \sin x - y''' \cos y + 3y'y'' \sin y + (y')^3 \cos y \end{aligned}$$

  我们当然还可以继续下去。如果我们决定仅应用Taylor展开中到h^4之前的项，那么其它项共同构成方法的截断误差，所对应的方法称为四阶方法
- 注意求导中要应用$d \sin y(x)/dx$的链式法则
- 当然可以执行各种代换，使得右边不出现y的导数y', y'', y''', ...。但如果是按上面给出的次序应用这些公式的话，就不必进行这种代换
- 运行Mathematica程序ode_taylor.nb

局部截断误差的累加

- 在上面的算法的每一步中，因为不包含Taylor级数中涉及$h^5, h^6, \ldots$的项，所以局部截断误差是$\mathcal{O}(h^5)$
- 因此当$h \to 0$时，局部截断误差类似于Ch^5。但我们并不知道C是多大
- 不过此例中$h = 0.01$，因此$h^5 = 10^{-10}$，每一步中的误差粗略地具有10^{-10}的量级，因此几百步后这些小的误差累加起来，可能不太会损坏精度
- 另外，在每一步中，$y(x_k)$的估计值y_k中已包含误差，进一步地计算继续增加这些误差，因此在得到的数值解中，不要盲目地采用所有的数字

解的验证

- 因此我们需要给出一种方法，来确定最终解的有效数字到底是多少?
- 在此例中我们有$y_{200} = 6.42194$. 以这个值作为同样方程的初值，并且取$h = -0.01$, 重复前面的求解过程，得到$x = -1.0$时解为3.00000, 它与原来的初值几乎相同，因此我们可以认为原来的解具有六位精度

误差的数值估计

- 在n阶方法中，Taylor级数展开到h^n项，那么有如下的误差估计

$$E_n = \frac{1}{(n+1)!} h^{n+1} y^{(n+1)}(x+\theta h), \qquad 0<\theta<1$$

- 因此可以用简单的有限差分逼近估计这个误差。例如，对上前例，$n=4$, $h=0.01$, 那么

$$E_4 \approx \frac{1}{5!} h^5 \frac{y^{(4)}(x+h) - y^{(4)}(x)}{h} = \frac{h^4}{120}[y^{(4)}(x+h) - y^{(4)}(x)]$$

优点

- 方法概念简单，并且具有高精度的潜力。如果能很容易地得到$y(x)$的20阶导数，则没有什么能阻止我们使用20阶的方法。应用这样高的阶，同样的精度情形下可以采用较大的步长，如$h = 0.2$. 穿过给定区间需要的步数变少，从而有可能减小计算量
- 可以应用符号计算系统执行非数值类型的计算，从而把相当复杂的表达式的微分和积分转换到这些系统中进行。这些系统还可以把计算表达式转化为所需要的代码

缺点

- 依赖于给定的微分方程的反复求导，因此在解曲线经过的$x-y$平面的区域内函数$f(x,y)$必须具有所需要的偏导数。而这样的条件对于解的存在性是不必要的
- 需要对问题进行初步的分析工作。从而在这个步骤中造成的误差可能被忽略而且始终不被发现
- 对于各阶求导必须单独编程，增加了编程的复杂性以及编程错误出现的可能性，代码的可读性下降

延迟型微分方程

- 在一些实际问题中有一类特殊类型的微分方程，称为延迟型微分方程(delay differential equation)或具有延迟变量的微分方程(differential equation with retarded argument)
- 人口模型以及混合问题通常具有这种特征，即$y'(x)$的值与y在x的前面值上的函数值有关
- 例如：

$$y'(x) = f(y(x-1))$$

若知道y在$x-1$上值，微分方程就能够计算$y'(x)$的值。为了从$x=0$开始积分微分方程，我们需要在$x=-1$开始的$y(x)$的变化情况。因此必须提供$y(x)$在区间$[-1,0]$上的值作为初值：

$$\begin{cases} y'(x) = y(x-1) & x \geqslant 0 \\ y(x) = x^2 & -1 \leqslant x \leqslant 0 \end{cases}$$

- 上例中第二个等式给出所需要的$y(x)$的值。若x限定在区间$[0,1]$中，则$x-1$在$[-1,0]$中，因此

$$\begin{cases} y'(x) = y(x-1) = (x-1)^2 & 0 \leqslant x \leqslant 1 \\ y(0) = 0 \end{cases}$$

这是一个通常的ODE，通过积分可以得到解为

$$y(x) = \frac{1}{3}(x-1)^3 + \frac{1}{3}, \qquad 0 \leqslant x \leqslant 1$$

- 如果解被延拓到下一个区间$[1,2]$上，则可以类似处理。此时，对于$x \in [1,2]$, 我们有

$$\begin{cases} y'(x) = y(x-1) = \frac{1}{3}(x-2)^3 + \frac{1}{3} & 1 \leqslant x \leqslant 2 \\ y(1) = \frac{1}{3} \end{cases}$$

也可以得到显式解。类似计算可以一直持续下去

复杂情形

- 对于复杂的方程，如

$$y'(x) = \sin[y(x-1)^3] + \log[y(x) + x^5]$$

我们需要借助于数值方法在每一个区间上求解：Taylor级数方法

- 例如，考虑

$$\begin{cases} y'(x) = 2y(x-1) + y(x) & x > 0 \\ y(x) = x^3 & -1 \leqslant x \leqslant 0 \end{cases}$$

- 为了在区间[0, 1]中求解，采用如下截断的Taylor展开：

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{6}y'''(x)$$

以步长h向前进行求解

- 我们需要提供下述导数表达式：

$$\begin{aligned}
y'(x) &= 2y(x-1) + y(x) = 2(x-1)^2 + y(x) \\
y''(x) &= 2y'(x-1) + y'(x) = 4(x-2)^2 + 2(x-1)^2 + y'(x) \\
y'''(x) &= 2y''(x-1) + y''(x) = 8(x-3)^2 + 8(x-2)^2 \\
&\qquad + 2(x-1)^2 + y''(x)
\end{aligned}$$

- 基于上述信息，可以得到[0, 1]中的离散点上$y(x)$的值。同时为了在下一区间内使用，需要存放在这些离散点上的$y'(x)$, $y''(x)$和$y'''(x)$的值。若不改变h的值，那么可以在每个区间上应用适当的存储值类似处理