第二节 常微分方程初值问题的单步法 – 微分方程数值解

笔记本：	我的第一个笔记本
创建时间：	2017/5/9 15:11
URL：	https://iclass.bupt.edu.cn/webapps/blackboard/content/listContent.jsp?course_id=_11052_1&content_id=_52836_1

第二节 常微分方程初值问题的单步法

学习指导：B 初值问题 第二节

本节介绍常微分方程初值问题的单步法。

作业&思考：B 初值问题 第二节

1. 试分析Runge-Kutta法的Heun三阶格式的收敛性。
2. 试分析改进Euler格式关于初值的稳定性。

讲义：B 初值问题 第二节

3.3 单步法

在前一节，我们介绍了 $Euler$ 法. $Euler$法为单步法，用它计算第 n 个节点的近似值时只用到前一节点的值. $Euler$ 法的优点很显然，该方法编程实现容易；具有良好的稳定性；容易改变步长；且为自起步的算法. 但是，$Euler$ 法简单地取切线的端点作为下一步的起点进行计算，当步数增多时，误差会因积累而越来越大，因此$Euler$格式一般不用于实际计算.

一般而言，当舍入误差占主导地位时，可用缩小步长的方法提高求解精度. 如果还不能达到缩小误差的目的，应采用精度更高的方法. 所以我们需要寻找一种新方法. 适当增加计算量较快提高精度. 通常，局部截断误差的阶是衡量一个方法精度高低的主要依据，本节将介绍基于提高局部截断误差阶来提高方法的精确度的数值算法：$Runge-Kutta$法.

$Runge-Kutta$法的基本思想

单步递推法的基本思想是从(t_n, u_n)点出发，以某一斜率沿直线达到(t_{n+1}, u_{n+1}) 点. 为提高精度，在$Euler$格式的基础上进行改进的梯形格式和改进 $Euler$ 法的主要出发点是采用区间两端的函数值的平均值作为直线方程的斜率. 由此，可得改进 $Euler$ 法（2 阶格式）如下：

$$\begin{cases} u_{n+1} = u_n + \dfrac{h}{2}(K_1 + K_2) \\ K_1 = f(t_n, u_n) \\ K_2 = f(t_n + h, u_n + hK_1) \end{cases}$$

若K_1仍取函数$u(t)$在t_n点的斜率，但K_2取为在区间$(t_n, t_{n+1}]$上某一点的斜率，记这一点为

$$t_{n+p} = t_n + hp, 0 < p \leq 1.$$

记$u_{n+p} = u_n + hpK_1$,则有$K_1 = f(t_n, u_n)$, $K_2 = f(t_{n+p}, u_{n-p})$. 取$K_1$和$K_2$的加权平均值作为平均斜率$K^*$的近似，即$K^* = c_1K_1 + c_2K_2$，则有如下的计算公式

$$\begin{cases} u_{n+1} = u_n + h[c_1K_1 + c_2K_2] \\ K_1 = f(t_n, u_n) \\ K_2 = f(t_n + hp, u_n + hpK_1) \end{cases} \tag{3.3.1}$$

若参数c_1, c_2及p使得格式(3.3.1)有2阶精度，该格式称为2阶的$Runge-Kutta$格式. 使用$Taylor$ 级数

展开的方法可以确定3个待定系数c_1, c_2及p, 使公式具有2 阶精度.

设对于单步法(3.3.1),第n 个节点的数值求解是准确的, 即$u_n = u(t_n)$, 则$K_1 = f(t_n, u(t_n)) = u'(t_n)$. 由二元函数的Taylor展开式得

$$\begin{aligned}
K_2 &= f(t_n + ph, u_n + hpK_1) \\
&= f(t_n, u_n) + phf_t(t_n, u_n) + hpK_1 f_u(t_n, u_n) + O(h^2) \\
&= f(t_n, u(t_n)) + ph[f_t(t_n, u(t_n) + u'(t_n)f_u(t_n, (t_n))] + O(h^2) \\
&= u'(t_n) + phu''(t_n) + O(h^2).
\end{aligned}$$

代入式(3.3.1), 可得

$$\begin{aligned}
u_{n+1} &= u_n + h(c_1K_1 + c_2K_2) \\
&= u(t_n) + h[c_1u'(t_n) + c_2u'(t_n) + c_2phu''(t_n) + c_2O(h^2)] \\
&= u(t_n) + (c_1 + c_2)hu'(t_n) + c_2ph^2u''(t_n) + O(h^3).
\end{aligned}$$

将$u(t_{n+1})$在$t = t_n$处进行$Taylor$展开得

$$u(t_{n+1}) = u(t_n) + hu'(t_n) + \frac{h^2}{2}u''(t_n) + O(h^3),$$

由$u(t_{n+1}) - u(t_n) = O(h^3)$, 可知待定系数$c_1$, c_2及p满足方程组

$$\begin{cases} c_1 + c_2 = 1; \\ c_2p = 1/2. \end{cases} \tag{3.3.2}$$

两个方程构成的方程组(3.3.2)有三个待定参数, 故2阶的$Runge-Kutta$ 格式为一系列的差分格式, 在三个参数中自定一个, 再用上式解出其余二个, 则就是一个具体的$Runge-Kutta$法的2阶公式. 如取$c_1 = 1/2, c_2 = 1/2, p = 1$则得到改进的$Euler$格式(3.2.9), 又如取$c_1 = 0, c_2 = 1, p = 1/2$则得到2阶的中点格式

$$\begin{cases} u_{n+1} = u_n + hK_2 \\ K_1 = f(t_n, u_n) \\ K_2 = f(t_n + \frac{1}{2}h, u_n + \frac{1}{2}hK_1) \end{cases} \tag{3.3.3}$$

二阶$Runge-Kutta$格式用多算一次f函数值的办法避开了二阶$Taylor$级数法所要计算的f的导数. 在这种意义上, 可以说$Runge-Kutta$方法实质上是$Taylor$级数法的变形. 由此启发, 我们得到提高精度的一种基本途径.

考察方程(3.1.1), 由$Lagrange$ 微分中值定理知, 存在一点$\theta \in [0, 1]$使得

$$\frac{u(t_{n+1}) - u(t_n)}{h} = u'(t_n + \theta h) = f(t_n + \theta h, u(t_n + \theta h)) = K^*,$$

可得

$$u(t_{n+1}) = u(t_n) + hK^*, \tag{3.3.4}$$

其中K^*称为$u(t)$在区间$[t_n, t_{n+1}]$上的平均斜率. 对于单步递推格式(3.3.4),只要知道平均斜率, 就可数值求解. 因此只要对平均斜率提供一种近似算法, 则由(3.3.4)式可导出一种相应的求解公式.

现在的问题已经转化为, 如何对K^*进行数值计算. 从理论上讲, 只要函数$u(t)$在区间$[0, T]$上充分光滑, 那么它的各阶导数值$u^k(t_n)$与函数$u(t)$在区间$[0, T]$上某些点的值就相互有联系, 就是说, 函数值可用各阶导数值近似地表示出来, 反之, 各阶导数值也可用函数在一些点上值的线性组合近似地表示出来.

如同机械的数值积分公式的推导, 可取$u(t)$在$[t_n, t_{n-1}]$上若干个点的斜率值, 或预报斜率值$K_1, K_2, \cdots K_m$的加权平均

$$\sum_{r=1}^{m} c_r K_r (\sum_{r=1}^{m} c_r = 1)$$

作为 K^*的近似值. 由此可得, 若设计$[t_n, t_{n+1}]$上若干个点的斜率值, 或预报斜率值K_1,K_2,$\cdots$, K_m, 以及加权系数c_1,c_2,$\cdots$ c_m ($\sum_{r=1}^{m} c_r = 1$), 使得差分格式

$$u_{n+1} = u_n + h \sum_{r=1}^{m} c_r K_r \quad (3.3.5)$$

达到期望阶的精度, 我们就建立了m级的单步递推格式, 式(3.3.5)称为 m 级的$Runge-Kutta$ 格式.

$Runge-Kutta$格式

一般而言, $Runge-Kutta$需要构造差分格式

$$u_{n+1} = u_n + h \sum_{r=1}^{m} c_r K_r \tag{3.3.6}$$

其中

$$\begin{cases} K_1 = f(t_n, u_n) \\ K_2 = f(t_n + a_2 h, u_n + b_{21} h K_1) \\ K_3 = f(t_n + a_3 h, u_n + b_{31} h K_1 + b_{32} h K_2) \\ \cdots \\ K_m = f(t_n + a_m h, u_n + b_{m1} h K_1 + b_{m2} h K_2 + \cdots + +b_{m,m-1} h K_{m-1})) \end{cases} \tag{3.3.7}$$

且$\sum_{r=1}^{m} c_r = 1, \sum_{k=1}^{r-1} b_{rk} = a_r, r = 2, 3, \cdots, m.$

格式(3.3.7)中的$\{a_r\}$,$\{b_{rk}\}$和$\{c_r\}$均为待定系数, 确定系数的原则就是使得局部截断误差的阶尽可能高即可. 系数$\{a_r\}$,$\{b_{rk}\}$和$\{c_r\}$一般按如下的原则确定: 将K_r 在(t_n, u_n)点进行$Taylor$展开, 代入(3.3.6) 式, 使l 次幂$h^l(l = 0, 1, \cdots, p)$ 的系数和$u(t_{n+1})$在$t = t_n$处的$Taylor$ 展开式的同次幂的系数相等. 如此得到的算法(3.3.6)$-$ (3.3.7) 称之为m级p 阶的$Runge-Kutta$格式, 若$p = m$, 则就简单地称(3.3.6)$-$ (3.3.7) 为m阶的$Runge-Kutta$ 格式.

考虑$m = 3$的情形, 对于差分格式

$$\begin{cases} u_{n+1} = u_n + h(c_1 K_1 + c_2 K_2 + c_3 K_3) \\ K_1 = f(t_n, u_n) \\ K_2 = f(t_n + a_2 h, u_n + b_{21} h K_1) \\ K_3 = f(t_n + a_3 h, u_n + b_{31} h K_1 + b_{32} h K_2) \end{cases} \tag{3.3.8}$$

类似二阶$Runge-Kutta$格式的推导, 利用$Taylor$展开式, 可得常见的三阶$Runge-Kutta$ 格式差分格式有

(1) $Heun$三阶格式

$$
\begin{cases}
u_{n+1} = u_n + \dfrac{h}{4}(K_1 + 3K_3) \\
K_1 = f(t_n, u_n) \\
K_2 = f(t_n + \dfrac{1}{3}h, u_n + \dfrac{1}{3}hK_1) \\
K_3 = f(t_n + \dfrac{2}{3}h, u_n + \dfrac{2}{3}hK_2)
\end{cases}
\tag{3.3.9}
$$

(2) $Kutta$三阶格式

$$
\begin{cases}
u_{n+1} = u_n + \dfrac{h}{6}(K_1 + 4K_2 + K_3) \\
K_1 = f(t_n, u_n) \\
K_2 = f(t_n + \dfrac{1}{2}h, u_n + \dfrac{1}{2}hK_1) \\
K_3 = f(t_n + h, u_n - hK_1 + 2hK_2)
\end{cases}
\tag{3.3.10}
$$

上述3阶$Runge-Kutta$ 格式的截断误差阶都是$O(h^4)$.

通常人们所说的 $Runge-Kutta$ 法是指四阶而言的, 同样可仿照二阶的情形推导出如下常用的四阶 $Runge-Kutta$格式:

(1) 四阶经典$Runge-Kutta$格式

$$
\begin{cases}
u_{n+1} = u_n + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\
K_1 = f(t_n, u_n) \\
K_2 = f(t_n + \dfrac{h}{2}, u_n + \dfrac{h}{2}K_1) \\
K_3 = f(t_n + \dfrac{h}{2}, u_n + \dfrac{h}{2}K_2) \\
K_4 = f(t_n + h, u_n + hK_3)
\end{cases}
\tag{3.3.11}
$$

(2) 四阶常用$Runge-Kutta$格式

$$
\begin{cases}
u_{n+1} = u_n + \dfrac{h}{8}(K_1 + 3K_2 + 3K_3 + K_4) \\
K_1 = f(t_n, u_n) \\
K_2 = f(t_n + \dfrac{h}{3}, u_n + \dfrac{h}{3}K_1) \\
K_3 = f(t_n + \dfrac{2h}{3}, u_n - \dfrac{h}{3}K_1 + hK_2) \\
K_4 = f(t_n + h, u_n + hK_1 - hK_2 + hK_3)
\end{cases}
\tag{3.3.12}
$$