第 18 卷第 3 期　　南京气象学院学报　　Vol. 18 No. 3
1995 年 9 月　　Journal of Nanjing Institute of Meteorology　　Sep. 1995

递归相似和线性多步法*

刘桂馥
（南京气象学院基础科学系，南京　210044）

摘要　在递归相似的基础上提出数值求解初值问题的一种新方案，与传统方法的区别，在于求解归结为演化系数的计算。这里给出常微（组）初值问题的算法；并指出新方法与传统方法的关系。

关键词　演化系数，递归相似，微分领域

分类号　O241.8

常微分方程初值问题有两类初等算法，即线性多步法和 Runge-Kutta 方法[1]。前者是从与初值问题等价的积分方程入手

$$y(t)=y^0+\int_0^t F(s,y(s))\mathrm{d}s$$

对未知的被积函数作精度较高的数值积分，借以估算解的近似值。

Runge-Kutta 法相对于多步法又称非线性方法，它是借函数在局部的 Taylor 公式，对解 $y(t)$ 作高阶逼近。在应用中，求解初值问题初等方法中确有一些成功的新形式，究其实质都不外乎这两套方法的变形。

本文借递归概念建立的递归方法[2]当然也不例外，然而仅仅是方法形式的不同，往往会产生新的意义，此处可指望把常微和偏微分方程数值方法统一起来。

1　递归方法及其意义

1.1　函数、微分邻域和态

光滑函数 $y(t)$ $(t>0)$ 及其导函数 $\dot{y},\ddot{y},\cdots$，构成向量

$$\boldsymbol{V}(t)=(y,\dot{y},\ddot{y},\cdots)^{\mathrm{T}} \tag{1}$$

此向量称为 y 在 t 时刻的递归态，由态向量构成的某个度量空间，称 $y(t)$ 的微分邻域。

态向量还可以定义成另一种形式

$$\boldsymbol{V}(t)=(t,y,\dot{y},\cdots)^{\mathrm{T}} \tag{2}$$

递归态的分量个数可以根据 y 的光滑性和问题需要，合适地确定。

把(1)或(2)视为空间 $(y,\dot{y})$ 和 $(t,y,\dot{y})$ 中的点，那么(1)、(2)分别在空间中各自描出一条曲线 L，称 L 为 y 的态曲线。态曲线仅仅是曲线 $y(t)$ 的派生物，理论上并不产生什么新的数学意义。然而借此来说明和理解递归概念却方便，直观。

---

* 中国气象局气象科技短平快资助项目

收稿日期：1994－07－14；改回日期：1995－01－05

1.2 递归方法

线性多步法实质是借 y 在多点的值实现了对近似解的高阶逼近。而 Runge-Kutta 法则仅借 $y(t)$ 在一点的值作出高阶逼近。由于这些算法的复杂形式，它们的这种内涵表露得并不清楚。

递归方法的形象是这样的，若对 $t, t_1, t_2, \cdots$ $(t > t_1 > t_2 > \cdots)$ 处的态已知，一般有

$$\boldsymbol{V}(t) = \lambda_1 \boldsymbol{V}(t_1) + \lambda_2 \boldsymbol{V}(t_2) + \cdots \tag{3}$$

其中 λ_i 为 t 的函数，称为系统在 t 时刻的演化系数(演化坐标，递归坐标)。

递归方法将归结为对演化系数 $\lambda(t)$ 的计算和估算。

1.3 偏微分方程的初值问题

大气科学中的数值模式是一个拟线性(双曲)偏微组的初值问题。目前，它所使用的数值方法仅有网格法和谱方法两种。由于问题规模过大，即多层次和大量的网格点，计算时效是个很大的问题，此外，计算质量很难从理论上得到论证。

为此，我们认为考虑有效的粗算法极有价值。

偏微与常微初值问题的数值方法很不相同。然而按递归的概念可以形式地把这两类算法统一起来。

记函数 $u(t,x) = u(t, x_1, x_2, \cdots)$

$$u(t_i, x) = u_i(x) \quad i = 0,1,2,\cdots$$

称 $u_i(x)$ 为 $u(t,x)$ 在 t_i 时刻的态。取 $t > t_i$，记

$$Su(t,x) = \sum_{i=1}^{p} \lambda_i(t)\, u_i(x)$$

称上式为 $u(t,x)$ 的有限阶递归相似，$\lambda_i(t)$ 称演化系数(坐标)。当然，一般 $Su \neq u$。

$$u(t,x) = Su(t,x) + \bar{u}(t,x)$$

其中 $\bar{u}$ 为递归表示之后的余项，称递归补。

我们认为偏微初值问题设计的递归方法就是借方程估算 λ_i 和 u，实现对近似解的估算。

常微问题是有限维(介质)问题，可以作有限的递归表示，递归补恒为零。

2 常微分方程的递归方法

这里仅讨论一阶方程的初值问题就够了。因为高阶方程可以化成一阶方程组，而对方程组描述递归方法要比方程式更为简便，自然。

本文把递归与多步法并列，是因为多步法实则是一种递归过程。我们认为，这里构成的递归方法形象更清楚，形式更一般，仅仅在新方法的形式下才可能实现对偏微的推广。

在阐述方法之前，先讨论它的理论根据。

2.1 态的独立性

对函数 $y(t)$，考虑它的态为($t \geqslant 0$)

$$\boldsymbol{V}(t) = (y(t), \dot{y}(t), \ddot{y}(t))^{\mathrm{T}}$$

若总有 $\boldsymbol{V} = 0$ (向量)，此时 $y \equiv 0$。

若 $\boldsymbol{V}(t_1) \neq 0$，又 $\boldsymbol{V}(t_1)$、$\boldsymbol{V}(t)$ $(t \geqslant 0)$ 线性相关，自然有

$$\begin{vmatrix} y(t_1) & y(t) \\ \dot{y}(t_1) & \dot{y}(t) \end{vmatrix} \equiv 0$$

即

$$y(t_1)\,\dot{y}(t) - \dot{y}(t_1)\,y(t) = 0$$

表明 $y(t)$ 是一阶线性常系数齐次方程的解 $y(t) = ae^{kt}$ 。

若 $\boldsymbol{V}(t_1)$、$\boldsymbol{V}(t_2)$ 线性独立,但对任何 $\boldsymbol{V}(t)$ 它们相关,即

$$\det(\boldsymbol{V}(t_1),\boldsymbol{V}(t_2),\boldsymbol{V}(t)) = \begin{vmatrix} y(t_1) & y(t_2) & y(t) \\ \dot{y}(t_1) & \dot{y}(t_2) & \dot{y}(t) \\ \ddot{y}(t_1) & \ddot{y}(t_2) & \ddot{y}(t) \end{vmatrix} \equiv 0$$

易知,$y(t)$ 为二阶线性常系数齐次方程的解 $y(t) = ae^{kt} + be^{kt}$ 。

以上讨论可归纳为如下结论:函数 y 的态(取在不同点上)相关的充要条件是 $y(t)$ 为线性常系数齐次方程的解。

事实上,这种方程有极简单的表达式,无需作数值解研究。因此,递归方法求解的对象将不包括线性常系数问题。在这种前提下,仍需注意 $\boldsymbol{V}(t_1)$、$\boldsymbol{V}(t_2)$、$\boldsymbol{V}(t)$ 的行列式不恒等于零,但可能有零点。即 $t = \bar{t}$ 时

$$\det(\boldsymbol{V}_1,\boldsymbol{V}_2,\boldsymbol{V}(\bar{t})) = 0$$

不过,此时总可选到点 $t = t_3$ 使

$$\det(\boldsymbol{V}_1,\boldsymbol{V}_2,\boldsymbol{V}_3) \neq 0$$

以下所建立的方法,在 $0 \leqslant t < T_1$ 上总可选到三个点 t_1 、t_2 、t_3 ,它们的态行列式不为零。只要待解方程不是线性常系数方程。方阵(4)称演化矩阵,

$$(\boldsymbol{V}_1,\boldsymbol{V}_2,\boldsymbol{V}_3) \equiv \begin{pmatrix} y_1 & y_2 & y_3 \\ \dot{y}_1 & \dot{y}_2 & \dot{y}_3 \\ \ddot{y}_1 & \ddot{y}_2 & \ddot{y}_3 \end{pmatrix} \tag{4}$$

这里记 $y(t_i) = y_i$, $\dot{y}(t_i) = \dot{y}_i, \cdots$ 。

从计算的角度,应使演化方阵条件数尽量小为宜(减少病态性),这在设计算法中要注意 t_i 的选择。

注:极易证明,线性常系数问题,即递归方法不予处理的问题,它们的态曲线在空间 $(y,\dot{y},\ddot{y})$ 中分别为直线或平面曲线。

2.2 递归方法原理、逼近的阶

无需涉及方程就可以看到该方法的原理和效果。简单计,只考虑 $y(t)$ 的一阶态 $\boldsymbol{V}(t) = (y,\dot{y})^{\mathrm{T}}$ 。

取 $t_3 > t_2 > t_1$,选择 t_1 、t_2 使之 $\det(\boldsymbol{V}_1,\boldsymbol{V}_2) \neq 0$ 。由下式定义 t_3 处的演化系数 $\lambda_i = \lambda_i(t_3)$

$$\begin{pmatrix} y_3 \\ \dot{y}_3 \end{pmatrix} = \begin{pmatrix} y_1 & y_2 \\ \dot{y}_1 & \dot{y}_2 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \tag{5}$$

由(4),上式又可记为

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = (\boldsymbol{V}_1,\boldsymbol{V}_2)^{-1} \begin{pmatrix} y_3 \\ \dot{y}_3 \end{pmatrix} \tag{6}$$

此时,$\lambda_i(t_3)$ 被唯一确定。

下面考虑一种简单的递归方法,其目的是估算 $y(t_3 + h)$ 的近似值。这里 $h > 0$ 却充分小。自然,假设 y 在 t_1, t_2, t_3 局部有 Taylor 公式

$$y(t_i + h) = y_i + \dot{y}_i h + \frac{1}{2!}\ddot{y}_i h^2 + O(h^3) \tag{7}$$

计算下式,$\lambda_i = \lambda_i(t_3)$

$$\lambda_1 y(t_1+h)+\lambda_2 y(t_2+h)=(\lambda_1 y_1+\lambda_2 y_2)+(\lambda_1\dot{y}_1+\lambda_2\dot{y}_2)h+O(h^2) \tag{8}$$

由 λ_i 的定义知

$$\lambda_1 y(t_1+h)+\lambda_2 y(t_2+h)=y_3+\dot{y}_3 h+O(h^2) \tag{9}$$

另一方面，

$$y(t_3+h)=y_3+\dot{y}_3 h+\frac{1}{2!}\ddot{y}_3 h^2+O(h^3) \tag{10}$$

比较(9)、(10)有

$$\lambda_1 y(t_1+h)+\lambda_2 y(t_2+h)-y(t_3+h)=O(h^2)$$

也就是(11)至少是二阶的，

$$y(t_3+h)=\sum_{i=1}^{2}\lambda_i y(t_i+h)+O(h^2) \tag{11}$$

2.3 方程的简单递归方法

上节2.2已说明了简单递归法原理，这里讨论方法对方程的应用。

如下初值问题有唯一的局部光滑解 $y(t)$

$$\begin{cases} \dot{y}=F(t,y) \\ y(0)=y_0 \end{cases} \tag{12}$$

(1)递归方法对问题的要求比(12)条件多些。如需已知 $0\leqslant t<T_1$ 上的 $y(t)$，方法给出 $t\geqslant T$ 上的估值。

对实际问题可以这样要求，已知 $y_i=y(t_i)$

$$t_i=ih \quad i=0,1,2,\cdots,n$$

方法给出 $y(t_k)$ 估值

$$t_k=kh \quad k>n$$

这种要求并不难实现。若递归态取二阶，$n\geqslant 2$ 即可。y_0、y_1、y_2 值可由其他精度高的方法事先算出。

(2)态的计算

在 $t=t_i$ 处的态

$$\begin{cases} \dot{y}_i=F(t_i,y_i)\equiv F^i \\ \ddot{y}_i=\dot{F}^i=(F_t+FF_y)^i \end{cases} \tag{13}$$

记

$$\ddot{y}_i=\dot{F}^i \quad \dddot{y}_i=\ddot{F}^i\cdots$$

类似地计算 $y_i^{(k)}$ $k=3,4,\cdots$。

所以，只要给出 $y_i=y(t_i)$，即可求出 t_i 处的各阶演化态。

$$\boldsymbol{V}(t_i)=\boldsymbol{V}_i=(y_i,\dot{y}_i,\cdots)^{\mathrm{T}}$$

如果对解的未知值 $y(t_k)=y_k$ 已被估算出，那么，相应的态 $\boldsymbol{V}_k$ 即可得出。

(3)演化矩阵选取和解值估算

假设 y_k 和 $i<k$ 处的 y_i 均被确定，问题是估计 y_{k+1}。

简单计，仅取一阶态 $\boldsymbol{V}(t)=(y,\dot{y})^{\mathrm{T}}$。既然 y_i 已知，由此算出它们的态 $\{\boldsymbol{V}_i \quad i<k\}$，其中任两态均可构成一个演化矩阵。从中选 $\boldsymbol{V}_a,\boldsymbol{V}_b$，使 $(\boldsymbol{V}_a,\boldsymbol{V}_b)$ 条件数尽量地小，自然有 $\det(\boldsymbol{V}_a,\boldsymbol{V}_b)\neq 0$。

由(14)定义并计算 y_k 的演化系数

$$\boldsymbol{V}_k = (\boldsymbol{V}_a, \boldsymbol{V}_b)\begin{pmatrix}\lambda_1\\ \lambda_2\end{pmatrix} \tag{14}$$

进而得出 y_{k+1} 的估算公式

$$y_{k+1} = \lambda_1 y_{a+1} + \lambda_2 y_{b+1} \tag{15}$$

至此,完成了一步计算。重复以上算法估算 $y_{k+2}, y_{k+3}, \cdots$ 。

取一阶态来估算,至少可达在截断误差意义下的二阶精度。

2.4　演化系数的变率

在简单递归方法中,(14)式是精确的。y_k 的演化系数由(14)唯一地求出,而估算式(15)是近似的,因为 y'_k 与 y_{k+1} 的演化系数不同。

这里指出演化系数的变率估算方法,由此改善对(15)估算。记 $t_a = t - P_1, t_b = t - P_2$, P_i 为固定常数。注意如下恒等式

$$\begin{cases} y(t) = \lambda_1(t)y(t-P_1) + \lambda_2(t)y(t-P_2) \\ \dot{y}(t) = \lambda_1(t)\dot{y}(t-P_1) + \lambda_2(t)\dot{y}(t-P_2) \end{cases} \tag{16}$$

记向量 $(\lambda_1(t), \lambda_2(t))^{\mathrm{T}} = \boldsymbol{\Lambda}(t)$,(16)式的向量形式为

$$\boldsymbol{V}(t) = (\boldsymbol{V}(t-P_1), \boldsymbol{V}(t-P_2))\boldsymbol{\Lambda}(t) \tag{17}$$

求导上式,记 $\boldsymbol{V}_1 = \boldsymbol{V}(t-P_1)\cdots$, ∂_t 表示对矩阵元素求导,

$$\dot{\boldsymbol{V}}(t) = \partial_t(\boldsymbol{V}_1, \boldsymbol{V}_2)\boldsymbol{\Lambda}(t) + (\boldsymbol{V}_1, \boldsymbol{V}_2)\dot{\boldsymbol{\Lambda}}(t) \tag{18}$$

$$\dot{\boldsymbol{\Lambda}}(t) = (\boldsymbol{V}_1, \boldsymbol{V}_2)^{-1}[\dot{\boldsymbol{V}}(t) - \partial_t(\boldsymbol{V}_1, \boldsymbol{V}_2)\boldsymbol{\Lambda}(t)] \tag{19}$$

当取 $t = t_k, t - P_1 = t_a, t - P_2 = t_b$ 时,(19)的右端是可计算的,为一个确定的向量 $\boldsymbol{Q}(t_k)$,

$$\dot{\boldsymbol{\Lambda}}(t_k) = \boldsymbol{Q}(t_k) \tag{20}$$

有近似式

$$\boldsymbol{\Lambda}(t_{k+1}) = \boldsymbol{\Lambda}(t_k) + h\boldsymbol{Q}(t_k) \tag{21}$$

由此改换估算式(15),得到新估计,

$$y_{k+1} = \lambda_1(t_{k+1})\, y_{a+1} + \lambda_2(t_{k+1}) y_{b+1} \tag{22}$$

其中 $\lambda_i(t_{k+1})$ 即(21)式的分量。这种改善效果如何,尚待细致分析。不过,若 h 充分小,估计将优于(14)。应该指明,尽管演化系数估计(21)是一阶的,但(22)远非是一阶的。

确实还有改善演化系数估计(21)的可能性。积分(19)(两边皆为向量式)

$$\boldsymbol{\Lambda}(t) - \boldsymbol{\Lambda}(t_k) = \int_{t_k}^{t} \boldsymbol{Q}(s)\mathrm{d}s \tag{23}$$

其中被积向量

$$\boldsymbol{Q}(t) = (\boldsymbol{V}_1, \boldsymbol{V}_2)^{-1}[\dot{\boldsymbol{V}}(t) - \partial_t(\boldsymbol{V}_1, \boldsymbol{V}_2)\boldsymbol{\Lambda}(t)] \tag{24}$$

(23)是一个很特别的积分方程,其中除含明显的未知向量 $\boldsymbol{\Lambda}(t)$ 外,向量 $\boldsymbol{V}_1, \boldsymbol{V}_2, \dot{\boldsymbol{V}}$ 中还包含未知函数 $y(t)$ 。

从计算的角度看,若对 (t_k, t) 上的 $y(t)$ 有充分多的估值,可以借(23)改善对 $\boldsymbol{\Lambda}(t)$ 的估计效果。

然而,我们更大的兴趣在于方程(23)给出 $y(t)$ 及演化系数 $\lambda(t)$ 的一个重要关系。若把 $y(t)$ 视为一个可测量的动力系统半流,方程(23)对重建动力系统会有意义的。

2.5　一个有潜在意义的问题

递归方法的新意在于它更换了初值问题求解的对象。从针对"解函数"的传统观点转化成求解"解的演化系数 $\lambda(t)$ "。演化系数(或称递归坐标)的概念更接近于方程的算子和动力系统

的演化半群。

递归方法有两个明显的特点：

第1，在计算过程中，可以随时增大或减小演化态或递归态 $\boldsymbol{V}(t)$ 的维数，使求解在不同时刻粗、精程度不同。

第2，计算 t_k 时刻 y_k 的演化系数，可以选择 $0 \leqslant t < t_k$ 上两个时刻 t_a、t_b，由 $\boldsymbol{V}_a$、$\boldsymbol{V}_b$ 求 $\boldsymbol{V}_k$ 的演化系数。

当然，必须要 $\det(\boldsymbol{V}_a, \boldsymbol{V}_b) \neq 0$。事实上满足这个条件的两时刻 t_a、t_b 有多种取法。这就可能在同一时刻 t_k 处有多种演化系数组（它们都是精确的）。

由简单递归法估算 y_{k+1} 将有多个近似值。记 $\boldsymbol{V}_a, \boldsymbol{V}_b, \boldsymbol{V}_k \to \lambda \to y_{k+1}(a,b) \quad (t_a, t_b < t_k)$。这里，估算值 y_{k+1} 因 a、b 不同而不同。注意到重要的“外推算法”，该算法是一种“算子插值”过程。取某一组权值 $\{\rho_{a,b}\}$，令最终的估算值 $\bar{y}_{k+1}$

$$\bar{y}_{k+1} = \sum_{a,b} \rho_{a,b} y_{k+1}(a,b) \tag{25}$$

显然，$\bar{y}_{k+1}$ 的精度不低于每一个 $y_{k+1}(a,b)$ 的精度，但恰当的权重会使 $\bar{y}$ 精度大为提高。

如何选定权函数有待研究。猜想，既便(25)作简单平均运算，算法会达到计算“绝对稳定”。

3 小 结

本工作的兴趣开始于对动力系统的观测描述和诊断研究。递归概念得到了与多步法或Runge-Kutta法等效的结果。演化系数(坐标)是关键概念，由它们把“系统演化”和“解值”联系起来，演化系数应该看成动力系统的一个度量指标。

参 考 文 献

1 李荣华，冯果忱．微分方程数值解法．北京：人民教育出版社，1983．1～80

2 刘桂馥．递归相似．南京气象学院学报，1995，18(2)：166～171

RECURSIVE SIMILARITY AND LINEAR MULTI-STEP SCHEME

Liu Guifu

(Department of Basic Courses, Meteorology, NIM, Nanjing 210044)

Abstract Based on recursive similarity, a scheme is proposed for numerically solving the problem of initial values, differing from the traditional method in that the solution can come down to calculating the evolution coefficient. Presented therein is the algorithm of the initial value for an ordinary differential equation or the system. Also, the relation of the proposed scheme to the traditional one is illustrated.

Keywords evolution coefficient, recursive similarity, differential neighborhood