黄振侃◎编著

Shuzhi Jisuan

数值计算

——微分方程数值解

北京工业大学出版社

数值计算

——微分方程数值解

黄振侃　编著

北京工业大学出版社

内 容 提 要

本书从工科院校应用数学角度，在掌握一定的数值计算理论的基础上，着重于计算机计算和应用环节的学习和训练．本书的主要内容包括：微分方程数值解的基本概述；常用算法及其精度、稳定性、收敛性；在计算中应注意事项；在物理及工程中的应用等．

本书适合综合及工科院校应用数学专业本科生学习，也可作为其他专业研究生、教师和工程技术人员的自学和参考用书．

图书在版编目（CIP）数据

数值计算：微分方程数值解/黄振侃编著．—北京：北京工业大学出版社，2006.9

ISBN 7-5639-1693-8

Ⅰ．数…　Ⅱ．黄…　Ⅲ．数值计算-高等学校-教材　Ⅳ．O241

中国版本图书馆 CIP 数据核字（2006）第 114918 号

数 值 计 算

——微分方程数值解

黄振侃　编著

*

北京工业大学出版社出版发行

邮编：100022 电话：（010）67392308

各地新华书店经销

徐水宏远印刷厂印刷

*

2006 年 10 月第 1 版　2006 年 10 月第 1 次印刷

787 mm×1 092 mm　16 开本　12.75 印张　307 千字

印数：1～3 000 册

ISBN 7-5639-1693-8/G·842

定价：19.00 元

前　言

微分方程数值解有关的专著及教学用书已有很多，有的偏重于理论，有的偏重于工程，非本专业读者读起来均有一定困难．本书从工科院校应用数学角度，在掌握一定的数值计算理论的基础上，着重于计算机计算和应用环节的学习和训练，也就是以数值解为载体，培养读者的计算能力，解决实际问题的能力，提高应用综合素质．

本着上述宗旨，教材编写时立足于少而精，主要介绍：微分方程数值解的基本概念；常用算法及其精度、稳定性、收敛性；在计算中应注意的事项；在物理及工程中的应用。希望本教材起到抛砖引玉的作用，使读者体会到计算数学的规律性东西．

为加强读者的计算编程能力培养，编写了“上机实习”一章，尽管这些内容在很多参考书上或多或少能找到．读者还是应该具有这方面的基本功，才能应对将来的各种挑战．

为提高读者解决实际问题的能力，特意编写了“工程应用”一章，这是作者多年做实际课题用数值解解决工程问题的部分内容。本书应用的课题资料，有的是已经发表于相关期刊上，有的是第一次引用．

本书适合综合及工科院校应用数学专业本科生学习，也可作为其他专业研究生、教师和工程技术人员的自学和参考用书．

本书在编写过程中得到多位教师的帮助和支持．杨中华副教授为第 4 章内容提出建议，王术教授认真阅读了本书初稿并提出了修改意见，在此表示谢意．特别感谢各课题的合作者徐贺文教授、陆介平教授、王世洪教授、蒋大林教授、王学宝高工的支持，他们深厚的专业知识对作者的帮助很大．

在本书编写过程中，参阅了大量的他人著作和相关文献，已一一列入书后的“参考文献”中，在此对这些作者一并表示感谢．

编　者

2006.8

目 录

第 1 章 常微分方程差分算法 ······ 1
1.1 引言 ······ 1
1.2 算法简述 ······ 2
1.3 单步方法 ······ 6
1.4 线性多步方法 ······ 10
1.5 两点边值问题 ······ 14
1.6 方程组及高阶方程的数值解法 ······ 17
第 2 章 偏微分方程差分算法 ······ 20
2.1 引言 ······ 20
2.2 椭圆型方程差分方法 ······ 21
2.3 抛物型方程差分方法 ······ 25
2.4 双曲型方程差分方法 ······ 32
2.5 流体力学数值计算 ······ 37
第 3 章 近似解法 ······ 49
3.1 引言 ······ 49
3.2 Ritz 法与 Galerkin 法 ······ 49
3.3 加权余量法 ······ 59
3.4 边界元法 ······ 62
3.5 特解边界元法 ······ 65
3.6 应用特解边界元法解三维 Pennes 方程 ······ 66
第 4 章 椭圆型方程有限元法 ······ 71
4.1 引言 ······ 71
4.2 一维椭圆型方程的有限元法 ······ 71
4.3 二维问题的三角形元及插值函数 ······ 75
4.4 二维椭圆型偏微分方程有限元法 ······ 78
4.5 单元和插值函数 ······ 83
第 5 章 上机实习 ······ 89
5.1 引言 ······ 89
5.2 离散矩阵的计算 ······ 89
5.3 常微分方程差分算法实习 ······ 95
5.4 偏微分方程差分算法实习 ······ 96
5.5 椭圆型方程有限元法实习 ······ 98
5.6 计算机程序设计 ······ 102
5.7 微分方程数值解各算法程序清单 ······ 106
第 6 章 工程应用 ······ 163
6.1 引言 ······ 163

6.2 表冷器凝水问题的研究 …… 164
6.3 卤钨灯结构的研究 …… 167
6.4 肿瘤热疗疗效的研究 …… 171
6.5 复合材料残余应力的研究 …… 180
6.6 大型电炉对流换热的研究 …… 185
6.7 表冷器风速均匀化的试验与数值计算的研究 …… 189
参考文献 …… 194

第 1 章　常微分方程差分算法

1.1　引言

当前，数学模型日益受到人们的重视．首先是计算技术的不断发展，其次是交叉学科的发展，对事物定量的描述需求所致．微分方程数学模型在数学模型中占有重要地位，而常微分方程模型是常用的模型之一．如在传统的力学和运动学以及控制论中，描述系统的动态演变过程，以及其他学科中的初值问题．这些方程的一种特性是：如果物理系统的某一个指定初始时刻 $t = t_0$，当 $t \geqslant t_0$ 时，该问题的解存在，且由相应的初始条件、边界条件和其他附加条件唯一确定．

建立模型，首先要熟悉所在的学科专业知识，找出变量之间的定量关系，对它进行检验需作工程试验，将模型的状态与试验数据进行比较，其次是对模型的运动规律进行研究，满足人们对事物的了解和期望．

常微分方程能直接进行积分的是少数，而多数是借助于计算机来求常微分方程的近似解，是其中重要的算法．

有限差分法是常微分方程中数值解法中通常有效的方法．差分方法也称离散变量法，是一种递推算法．

建立差分算法有两个基本步骤：

第一步，建立差分格式．

主要工作是：①对解的存在域剖分；②采用不同的算法可得不同的逼近误差，即截断误差；③数值解对真解的精度；④数值解收敛于真解的速度；⑤差分算法的稳定性．

第二步，差分格式的求解．

将积分方程通过差分方程转化为代数方程求解，一般常用递推算法．

设初值问题

$$\begin{cases} u' = f(t,u) \qquad (t_0 \leqslant t \leqslant T) & (1.1) \\ u|_{t=t_0} = u_0 & (1.2) \end{cases}$$

式（1.1）、式（1.2）有解 $u(t)$，需满足存在唯一性、适定性．

定理 1.1　若 $f(t,u)$ 在 $t_0 \leqslant t \leqslant T$ 中连续，且 $u(t)$ 满足 Lipschitz 条件，存在 L（常数），使

$$|f(t,u) - f(t,u^*)| \leqslant L|u - u^*| \tag{1.3}$$

则式（1.1）、式（1.2）存在唯一的连续可微解 $u(t)$．

例如，扰动问题：

$$\begin{cases}\dfrac{\mathrm{d}v}{\mathrm{d}t} = f(t,u) + \delta(t) = \tilde{f}(t,v) & (1.4)\\ v(t_0) = u_0 + \varepsilon_0 = \tilde{u}_0 & (1.5)\end{cases}$$

式中，$\delta(t)$ 和 ε_0 都是很小的扰动.

定义 1.1 如果存在正常数 k 、η，使对任何 $\varepsilon \leqslant \eta$，当 $|\varepsilon_0| < \varepsilon$，$t_0 \leqslant t \leqslant T, |u| < \infty$ 时，有 $|\delta(t)| < \varepsilon$. 扰动问题式（1.4）及式（1.5）满足

$$|u(t) - v(t)| \leqslant k\varepsilon \tag{1.6}$$

称常微分方程（1.1）对初始条件（1.2）是适定的.

定理 1.2 如果 $f(t,u)$ 满足 Lipschitz 条件，则（1.1）对任何初值条件都是适定的.
证明令 ε 是由 $\max[|\varepsilon_0|, \max\limits_{t_0\leqslant t\leqslant T}|\delta(t)|]$ 定义的模 $\|\varepsilon_0,\delta\|$.
如果 $\varepsilon(t) = u(t) - v(t)$，则

$$\varepsilon'(t) = v'(t) - u'(t) = f(t,v) - f(t,u) + \delta(t)$$
$$\varepsilon(t_0) = |v(t_0) - u(t_0)| = |\varepsilon_0| \leqslant \varepsilon$$

于是得

$$|\varepsilon'(t)| \leqslant |f(t,v) - f(t,u)| + |\delta(t)| \leqslant L|\varepsilon(t)| + \varepsilon$$

一阶线性方程解

$$\begin{aligned}|\varepsilon(t)| &\leqslant \mathrm{e}^{\int_{t_0}^{t} L\mathrm{d}t}\left(\int_{t_0}^{t} \varepsilon \mathrm{e}^{-\int_{t_0}^{t} L\mathrm{d}t}\mathrm{d}t + C\right)\\ &= \mathrm{e}^{L(t-t_0)}\left(-\frac{\varepsilon}{L}\mathrm{e}^{-L(t-t_0)} + C\right)\end{aligned}$$

将 $C = \varepsilon + \dfrac{\varepsilon}{L}$，代入上式得

$$|\varepsilon(t)| \leqslant \frac{\varepsilon}{L}[(L+1)\mathrm{e}^{L(t-t_0)} - 1]$$

扰动问题解中最大改变量以 $k\varepsilon$ 为界：

$$\begin{aligned}\max_{t_0\leqslant t\leqslant T}|\varepsilon(t)| &\leqslant \|\varepsilon\| \frac{1}{L}[(L+1)\mathrm{e}^{L(t-t_0)} - 1]\\ &= \|\varepsilon_0,\delta\| \cdot \frac{1}{L}[(L+1)\mathrm{e}^{L(T-t_0)} - 1] = k\varepsilon\end{aligned}$$

式中，k 与 ε 无关.

1.2 算法简述

一般算法的过程必须能够在有限步后停止并有结果，说明中定义了实施措施；计算方法具备了算法的所有特征，但不能保证在有限步后终止，对实施措施一般只进行抽象描述.

1. 差分方法

差分算法的优点是通用性强，适用性广，方法简单易于掌握．其原理是，函数在某点处的值可通过 Taylor 公式表示，是一个无限项问题，需取有限项计算，因此产生截断误差问题，以及引起相关的其他问题，这正是差分方法所需研究的．

设 $u = u(t), t \in [t_0, T]$，节点 $t_i = t_0 + ih$，$(i=1,2,\cdots,n)$即 $t_{i+1} = t_i + h$ $(i = 0, 1,2,\cdots,n-1)$．其中步长 $h = \dfrac{T - t_0}{n}$．边界节点 $t_0, t_n = T$，函数 $u(t)$ 在节点 t_i 处，由 Taylor 公式可得

$$u(t+h) = u(t) + u'(t)h + \frac{1}{2!}u''(t)h^2 + \frac{1}{3!}u'''(t)h^3 + \cdots + \frac{1}{n!}u^{(n)}(t)h^n + o(h^{n+1})$$

为用有限项表示导数，常见导数关于截断误差 h 的阶数 k，视 $o(h^k)$ 表示如下：

$$\begin{cases} u'(t_i) = \dfrac{u(t_{i+1}) - u(t_i)}{h} + o(h) \\ u'(t_i) = \dfrac{u(t_{i-1}) - u(t_i)}{h} + o(h) \\ u'(t_i) = \dfrac{u(t_{i+1}) - u(t_{i-1})}{2h} + o(h^2) \\ u'(t_i) = \dfrac{3u(t_i) - 4u(t_{i-1}) + u(t_{i-2})}{2h} + o(h^2) \\ u''(t_i) = \dfrac{u(t_{i+2}) - 2u(t_{i+1}) + u(t_i)}{h^2} + o(h) \\ u''(t_i) = \dfrac{u(t_{i+1}) - 2u(t_i) + u(t_{i-1})}{h^2} + o(h^2) \\ u''(t_i) = \dfrac{-u(t_{i+3}) + 4u(t_{i+2}) - 5u(t_{i+1}) + 2u(t_i)}{h^2} + o(h^2) \\ u''(t_i) = \dfrac{-u(t_{i+2}) + 16u(t_{i+1}) - 30u(t_i) + 16u(t_{i-1}) - u(t_{i-2})}{12h^2} + o(h^4) \\ u'''(t_i) = \dfrac{u(t_{i+3}) - 3u(t_{i+2}) - 3u(t_{i+1}) - u(t_i)}{h^3} + o(h) \\ u'''(t_i) = \dfrac{u(t_{i+2}) - 2u(t_{i+1}) + 2u(t_{i-1}) - u(t_{i-2})}{2h^3} + o(h^2) \end{cases} \tag{1.7}$$

如从式(1.7)中舍去 $o(h^k)$项，可得不同差分格式．函数 $u(t)$ 在 t_i 点处计算值记作 u_i，设 $u_i \approx u(t_i)$，在式（1.7）中第一式舍去 $o(h)$，方程（1.1）可写为差分方程

$$u_{i+1} = u_i + hf(t_i, u_i)$$

建立何种差分格式，需从引言中建立差分格式的五个方面来考虑．

2.Euler 法（折线法）

在常微分方程差分法中最简单的方法是 Euler 方法，尽管在计算中不会使用，但从中可领悟到建立差分格式的技术路线．

(1) 算法

式（1.1）等价于积分方程

$$u(t+h)=u(t)+\int_t^{t+h} f(\tau,u(\tau))\mathrm{d}\tau \tag{1.8}$$

令 $t=t_i$，利用矩形积分公式

$$\int_{t_i}^{t_i+h} f(\tau,u(\tau))\mathrm{d}\tau = hf(t_i,u(t_i))+R_i$$

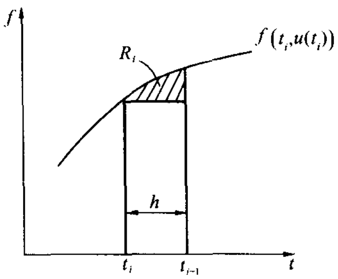

图 1.1　矩形积分示意图

如图 1.1 所示，R_i 是第 i 个小曲边形面积．

由式（1.8）得

$$u(t_{i+1})=u(t_i)+hf(t_i,u(t_i))+R_i \tag{1.9}$$

称 R_i 为 Euler 法局部截断误差，$\varepsilon_i=u(t_i)-u_i$ 为整体截断误差．

舍去式（1.9）中 R_i，令 $u_i\approx u(t_i)$，可得到 $t_{i+1}=t_i+h$ 处 Euler 公式为

$$u_{i+1}=u_i+hf(t_i,u_i) \tag{1.10}$$

式（1.10）随着步数增多，由于每步舍入误差都很大，误差积累的增加，可能把真解掩盖掉．

(2) 误差估计

由式（1.9）得

$$R_i=\int_{t_i}^{t_{i+1}} f(t,u(t))\mathrm{d}t-hf(t_i,u(t_i))=\int_{t_i}^{t_{i+1}}[f(t,u(t))-f(t_i,u(t_i))]\mathrm{d}t$$

误差界

$$\begin{aligned}|R_i| &\leqslant \int_{t_i}^{t_{i+1}}|f(t_i,u(t))-f(t_i,u(t))|\mathrm{d}t+\int_{t_i}^{t_{i+1}}|f(t_i,u(t))-f(t_i,u(t_i))|\mathrm{d}t\\ &\leqslant K\int_{t_i}^{t_{i+1}}|t-t_i|\mathrm{d}t+L\int_{t_i}^{t_{i+1}}|u(t)-u(t_i)|\mathrm{d}t\\ &=\frac{Kh^2}{2}+L\int_{t_i}^{t_{i+1}}|u'(t_i+Q(t-t_i))|(t-t_i)\mathrm{d}t\end{aligned}$$

式中，$0<Q<1$，取 $M=\max\limits_{t_0\leqslant t\leqslant T}|u'(t)|=\max\limits_{t_0\leqslant t\leqslant T}|f(t,u)|$，$K=\max\limits_{t_0\leqslant t\leqslant T}|f'_t|$，$L=\max\limits_{u\in D}|f'_u|$

记 $R=\dfrac{h^2}{2}(K+LM)$，可得

$$|R_i|\leqslant R=\frac{Kh^2}{2}+\frac{LMh^2}{2}=\frac{h^2}{2}(K+LM) \tag{1.11}$$

(3) 收敛性

定理 1.3　设 $f(t,u)$ 关于 t,u 满足 Lipschitz 条件，K,L 为相应的常数，且当 $h\to 0$ 时，$u_0\to u(t_0)$，则 Euler 法收敛，且误差界为

$$|\varepsilon_i|\leqslant \mathrm{e}^{L(T-t_0)}|\varepsilon_0|+\frac{h}{2}\left(M+\frac{K}{L}\right)(\mathrm{e}^{L(T-t_0)}-1) \tag{1.12}$$

如果 $u_0=u(t_0)$，由式（1.12），得整体截断误差 $|\varepsilon|=o(h)$，由式（1.11）得局部截断误

差 $|R_i| = o(h^2)$.

例 1.1 用 Euler 法解下式的初值问题，并估计它的离散误差.

$$\begin{cases} \dfrac{\mathrm{d}u}{\mathrm{d}t} + u^2 = 0 \qquad (0 \leqslant t \leqslant 1) \\ u(0) = 1 \end{cases}$$

解 解析解 $u = \dfrac{1}{1+t}$,

$$M = \max_{0\leqslant t\leqslant 1} |u'(t)| = \max_{0\leqslant t\leqslant 1} \left| \frac{-1}{(1+t)^2} \right| = 1$$

$$K = \max_{0\leqslant t\leqslant 1} |f'_t| = \max_{0\leqslant t\leqslant 1} \left| \frac{2}{(1+t)^3} \right| = 1$$

$$L = \max_{0\leqslant t\leqslant 1} |f'_u| = \max_{0\leqslant t\leqslant 1} |-2u| = 2$$

设 $\varepsilon_0 = 0, h = 0.1$，误差界为

$$|\varepsilon_i| \leqslant \frac{0.1}{2}\left(1 + \frac{1}{2}\right)(\mathrm{e}^2 - 1) = 0.487$$

(4) 稳定性

定义 1.2 设 u_i 及 v_i 是以任意初值 u_0 及 v_0 的 Euler 方法求出的精确解（没有舍入误差），如果存在常数 c 及 h_0，对任何满足条件 $0 < h < h_0, h < T$ 的 i 和 h 都有

$$|u_i - v_i| \leqslant c|u_0 - v_0| \tag{1.13}$$

则称 Euler 方法是稳定的.

从定义可知，稳定性是对于方法而言的.

定理 1.4 如果 $f(t,u)$ 关于 u 满足 Lipschitz 条件，则 Euler 方法是稳定的.

讨论稳定性概念，当 $h \to 0$ 时讨论，称之为古典稳定性. 在每次计算过程中，实际上是取固定步长 h 进行计算. 因此，最重要的是在计算过程中所产生的扰动（误差）对以后的计算结果的影响，不会步步增大，称之为绝对稳定性.

(5) 改进 Euler 方法

为提高局部截断误差关于 h 的阶数，对式（1.8）改用梯形积分公式，可得

$$u(t_{i+1}) = u(t_i) + \frac{h}{2}[f(t_i, u(t_i)) + f(t_{i+1}, u(t_{i+1}))] + R_i^{(1)}$$

求 $R_i^{(1)}$ 阶的估计式.

根据 Newton 前插值公式，在 $t_i \leqslant t \leqslant t_{i+1}$ 时，因为

$$\int_{t_i}^{t_{i+1}} u'(t)\mathrm{d}t = \int_0^1 \{u'(t_i) + \tau[u'(t_{i+1}) - u'(t_i)]\} h\mathrm{d}\tau + \frac{h^3}{2}\int_0^1 \tau(\tau-1)u'''(t_i + \xi h)\mathrm{d}\tau$$

$$= \frac{h}{2}[u'(t_{i+1}) + u'(t_i)] - \frac{h^3}{12}u'''(t_i + \xi h) \qquad (0 \leqslant \xi \leqslant 1)$$

所以

$$R_i^{(1)} = \int_{t_i}^{t_{i+1}} u'(t)\mathrm{d}t - \frac{h}{2}[u'(t_i) + u'(t_{i+1})] = -\frac{h^3}{12}u'''(t_i + \xi h)$$

$$R^{(1)} = \sup R_i^{(1)} \leqslant \frac{h^3}{12}M, \quad M = \max_{0\leqslant t\leqslant T}|u'''(t)|$$

可得

$$|R_i^{(1)}| = o(h^3)$$

略去 $R_i^{(1)}$ 得到改进的 Euler 方法：

$$u_{i+1} = u_i + \frac{h}{2}[f(t_i, u_i) + f(t_{i+1}, u_{i+1})] \tag{1.14}$$

这是一个隐格式，采用下面预估－校正算法是简易方便的.

$$\begin{cases} \text{预估} & u_{i+1}^{[0]} = u_i + hf(t_i, u_i) \\ \text{校正} & u_{i+1}^{[k+1]} = u_i + \dfrac{h}{2}[f(t_i, u_i) + f(t_{i+1}, u_{i+1}^{[k]})] \qquad (k = 0,1,2,\cdots) \end{cases} \tag{1.15}$$

式中，k 为迭代次数；$u_{i+1}^{[0]}$ 用显式 Euler 方法求得.

由式（1.15）求出迭代序列 $u_{i+1}^{[0]}$，$u_{i+1}^{[1]}$，$u_{i+1}^{[2]}$，$\cdots$ $\qquad (i = 1,2,\cdots,n-1)$

当 h 充分小时，序列收敛于改进的 Euler 方法的解 u_{i+1}，设 $f(t,u)$ 在 $t \in I = [a,b]$ 上满足 Lipschitz 条件（L 为常数），有

$$|u_{i+1}^{[k+1]} - u_{i+1}^{[k]}| = \frac{1}{2}h\,|f(t_{i+1}, u_{i+1}^{[k]}) - f(t_{i+1}, u_{i+1}^{[k-1]})|$$

$$\leqslant \frac{1}{2}Lh\,|u_{i+1}^{[k]} - u_{i+1}^{[k-1]}|$$

只要使 $\frac{1}{2}Lh < 1$ 及足够好的初始近似 $u_{i+1}^{[0]}$，解一定收敛于 u_{i+1}.

实际数值计算时，只需满足允许误差 ε，

$$|u_{i+1}^{[k]} - u_{i+1}^{[k-1]}| < \varepsilon \qquad (k = 0,1,2,\cdots)$$

$u_{i+1}^{[k-1]}$ 即为所求.

当舍入误差占主导地位时，一般再用缩小步长的方法，如果还不可能达到缩小误差的目的，应采用精度更高的方法.

1.3 单步方法

从 1.2 节中看出，Euler 方法是一种单步方法，即只需已知上一节点处解的近似值，便可求下一节点处的解的近似值.

Euler 方法的优点：①编制程序容易；②具有良好的稳定性；③改变步长容易；④是自起步的，也是最重要优点. 缺点：①近似解的精确度太低；②与其他精度相同的近似方法（数值解方法）需较长计算时间；③估计局部误差值难于求得. Euler 方法优点虽然很多，但缺点是主要的. 需寻找一种新方法，适当增加计算量，又较快提高精度.

单步方法中，Runge－Kutta 方法是高精度的单步方法，不需附加初值，因此可随意改变步长，不增加任何附加计算量，也是线性多步方法用来计算附加初值的常用方法之一. 其要点在于计算从 t_i 起适当增加不同节点的 f 值的线性组合，可提高截断误差的阶数，提高精度.

1. 单步方法

单步方法一般形式为

$$u_{i+1} = u_i + h\psi(t_i, u_i, h) \tag{1.16}$$

将 $\psi(t)$ 在 t_i 处展成 Taylor 级数，可得

$$\psi(t_i, u_i, h) = \sum_{j=1}^{r} \frac{h^{j-1}}{j!} \frac{\mathrm{d}^{(j-1)}}{\mathrm{d}t^{(j-1)}} f(t_i, u_i) + o(h^{r+1}) \tag{1.17}$$

式中，

$$u_i' = f(t_i, u_i)$$

$$u_i'' = f_{t_i} + f_{u_i} f_i$$

$$u_i''' = f_{t_i t_i} + 2f_i f_{t_i u_i} + f_i^2 f_{u_i u_i} + f_{u_i}(f_{t_i} + f_i f_{u_i})$$

$$\vdots$$

且 r 是使式（1.17）成立的最大整数，称式（1.16）为 r 阶单步方法.

2. Runge - Kutta 方法

Runge - Kutta 方法规定：

$$\begin{cases} \psi(t_i, u_i, h) = \sum\limits_{n=1}^{s} c_n k_n \\ k_1 = f(t_i, u_i) \\ k_n = f(t_i + ha_n, u_i + h\sum\limits_{l=1}^{n-1} b_{nl} k_l) \quad (n = 2,3,\cdots,s) \\ a_n = \sum\limits_{l=1}^{n-1} b_{nl} \\ u_{i+1} = u_i + h\psi(t_i, u_i, h) \end{cases} \tag{1.18}$$

式中，c_n，a_n，b_{nl} 为待定系数. 称算法（1.18）为 r 阶 s 级 Runge - Kutta 方法.

特别当 $r = 1$ 时，为 Euler 折线法.

$r = 2$ 阶，$s = 2$ 级情况，算法（1.18）取为

$$\begin{cases} \psi(t_i, u_i, h) = c_1 k_1 + c_2 k_2 \\ k_1 = f(t_i, u_i) \\ k_2 = f(t_i + ha_1, u_i + ha_2 k_1) \\ u_{i+1} = u_i + h\psi(t_i, u_i, h) \end{cases}$$

确定 c_1、c_2、a_2 值，由式（1.17）得

$$\psi(t_i, u_i, h) = \sum_{j=1}^{2} \frac{h^{j-1}}{j!} \frac{\mathrm{d}^{(j-1)}}{\mathrm{d}t^{(j-1)}} f(t_i, u_i) = f_i + \frac{h}{2}(f_{t_i} + f_u f_i) \tag{1.19}$$

k_2 在 (t_i, u_i) 处展开成 Taylor 级数，$k_2 = f_i + ha_2(f_{t_i} + k_1 f_{u_i})$. 将 k_1，k_2 代入

$$\psi(t_i, u_i, h) = c_1 k_1 + c_2 k_2$$

得

$$\psi(t_i, u_i, h) = (c_1 + c_2)f_i + hc_1a_2(f_{t_i} + f_u f_i) \tag{1.20}$$

比较式（1.19）和式（1.20），得

$$\begin{cases} c_1 + c_2 = 1 \\ c_1 a_2 = \dfrac{1}{2} \end{cases}$$

上式为三个未知数两个方程的不定方程组，需对（c_1, c_2, a_2）中设定一个值，其他两值根据上式计算.

设 c_1 分别为 0、$\frac{1}{4}$、$\frac{1}{2}$时，计算出三组值 $\left(0,1,\frac{1}{2}\right),\left(\frac{1}{4},\frac{3}{4},\frac{2}{3}\right),\left(\frac{1}{2},\frac{1}{2},1\right)$，得到三个二阶二级截断误差为 $o(h^3)$ 的方法：

（1）中点法（修正 Euler 方法）

$$\begin{cases} u_{i+1} = u_i + hk_2 \\ k_1 = f_i \\ k_2 = f(t_i + \dfrac{1}{2}h, u_i + \dfrac{h}{2}k_1) \end{cases} \tag{1.21}$$

（2）Houn 法

$$\begin{cases} u_{i+1} = u_i + \dfrac{h}{4}(k_1 + 3k_2) \\ k_1 = f_i \\ k_2 = f\left(t_i + \dfrac{2}{3}h, u_i + \dfrac{2}{3}hk_1\right) \end{cases} \tag{1.22}$$

（3）改进 Euler 法

$$\begin{cases} u_{i+1} = u_i + \dfrac{h}{2}(k_1 + k_2) \\ k_1 = f_i \\ k_2 = f(t_i + h, u_i + hk_1) \end{cases} \tag{1.23}$$

四阶 Runge - Kutta 方法是常用的算法，优点：精度高，程序简单，计算过程稳定，易于调整步长. 缺点：要求 $f(t,u)$ 具有较高的光滑性. 如果 f 光滑性差，可能不如 Euler 方法. 相对计算工作量大，费机时，每次需计算四次 f 值.

取 $r = 4, s = 4$，并将 Taylor 级数展开至 h^3 项，比较系数可得含有 11 个方程 13 个未知数的不定方程组，采用 $r = 2, s = 2$ 的处理方式，得到三种四阶 Runge - Kutta 形式.

（1）古典形式

$$\begin{cases} u_{i+1} = u_i + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(t_i, u_i) \\ k_2 = f(t_i + \dfrac{1}{2}h, u_i + \dfrac{h}{2}k_1) \\ k_3 = f(t_i + \dfrac{1}{2}h, u_i + \dfrac{h}{2}k_2) \\ k_4 = f(t_i + h, u_i + hk_3) \end{cases} \tag{1.24}$$

（2）Kutta 形式

$$\begin{cases} u_{i+1} = u_i + \dfrac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4) \\ k_1 = f(t_i, u_i) \\ k_2 = f(t_i + \dfrac{1}{3}h, u_i + \dfrac{h}{3}k_1) \\ k_3 = f(t_i + \dfrac{2}{3}h, u_i - \dfrac{h}{3}k_1 + hk_2) \\ k_4 = f(t_i + h, u_i + hk_1 - hk_2 + hk_3) \end{cases} \tag{1.25}$$

（3）Gill 形式

$$\begin{cases} u_{i+1} = u_i + \dfrac{h}{6}\left(k_1 + (2 - \sqrt{2})k_2 + (2 + \sqrt{2})k_3 + k_4\right) \\ k_1 = f(t_i, u_i) \\ k_2 = f\left(t_i + \dfrac{1}{2}h, u_i + \dfrac{h}{2}k_1\right) \\ k_3 = f\left(t_i + \dfrac{1}{2}h, u_i + \dfrac{\sqrt{2}-1}{2}k_1 + \left(1 - \dfrac{\sqrt{2}}{2}\right)k_2\right) \\ k_4 = f\left(t_i + h, u_i - \dfrac{\sqrt{2}}{2}k_2 + \left(1 + \dfrac{\sqrt{2}}{2}\right)k_3\right) \end{cases} \tag{1.26}$$

本书使用的是古典形式.

例 1.2 求初值问题：

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

解 解析解 $y = \sqrt{1 + 2x}$.

① Euler 法：

$$y_{n+1} = y_n + h\left(y_n - \frac{2x_n}{y_n}\right) \quad (h = 0.1,\ n = 0,1,2,\cdots,10)$$

② 改进 Euler 法：

$$y_{n+1} = y_n + \frac{h}{2}\left[\left(y_n - \frac{2x_n}{y_n}\right) + \left(y_{n+1} - \frac{2x_{n+1}}{y_{n+1}}\right)\right] \quad (h = 0.1,\ n = 0,1,2,\cdots,10)$$

③ 四阶 Runge – Kutta 方法：

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 = y_n - \dfrac{2x_n}{y_n} \\
k_2 = y_n + \dfrac{h}{2}k_1 - \dfrac{2\left(x_n + \dfrac{h}{2}\right)}{y_n + \dfrac{1}{2}hk_1} & (h = 0.2,\ n = 0,1,2,3,4,5) \\
k_3 = y_n + \dfrac{h}{2}k_2 - \dfrac{2\left(x_n + \dfrac{h}{2}\right)}{y_n + \dfrac{1}{2}hk_2} \\
k_4 = y_n + hk_3 - \dfrac{2(x_n + h)}{y_n + hk_3}
\end{cases}
$$

从计算结果中（见表 1.1）看到. 虽然四阶 Runge - Kutta 法计算量（每一步四次计算 f 值）比改进 Euler 法（二阶 Runge - Kutta 法每次只计算两个 f 值）大一倍，但由于放大步长 $h = 0.2$，总计算量相等，但是精度要高.

表 1.1　数值计算结果

x_n	y_n			$y(t_i)$
	①	②	③	
0.1	1.100 0	1.095 9	—	1.095 4
0.2	1.191 8	1.181 4	1.183 2	1.183 2
0.3	1.277 4	1.266 2	—	1.264 9
0.4	1.358 2	1.343 4	1.341 7	1.341 6
0.5	1.435 1	1.416 4	—	1.414 2
0.6	1.509 0	1.484 0	1.483 3	1.483 2
0.7	1.580 3	1.552 5	—	1.549 2
0.8	1.649 8	1.615 3	1.612 6	1.612 5
0.9	1.717 8	1.678 2	—	1.673 3
1.0	1.737 9	1.737 9	1.732 1	1.732 1

1.4　线性多步方法

计算过程中，增加多个节点的已知数据，减少由于计算方面而产生的误差是多步方法的思想. 在单步方法中，计算 u_{i+1} 时只用到前面 u_i 的值，现如果充分利用已知的 $u_0, u_1, \cdots, u_i$ 诸值，也就是利用这些点构造一个高精度的插值函数，来计算下一个节点值，以提高 u_{i+1}

的精度．以 Euler 两步法为例，采用改进 Euler 方法（1.14），隐式中 u_{i+1}采用中心差商计算，即

$$u_{i+1} = u_{i-1} + 2hf(t_i, u_i)$$

计算 u_{i+1} 需已知 u_{i-1} 和 u_i 两步数据，故称两步法．即

$$\begin{cases} \tilde{u}_{i+1} = u_{i-1} + 2hf(t_i, u_i) \\ u_{i+1} = u_i + \dfrac{1}{2}h[f(t_i, u_i) + f(t_{i+1}, \tilde{u}_{i+1})] \end{cases}$$

式中，$\tilde{u}_{i+1}$为中间变量．

如果要求关于因变量多步数据，计算 $t = t_{i+1}$ 处的值，称为多步法．一般表示为

$$u_{i+1} = \alpha_0 u_i + \alpha_1 u_{i-1} + \cdots + \alpha_k u_{i-k} + h(\beta_{-1} u'_{i+1} + \beta_0 u'_i + \cdots + \beta_k u'_{i-k})$$

式中，$u'_i = f_i = f(t_i, u_i), \alpha_i, \beta_i$ 是常数．需已知 $u_i, u_{i-1}, \cdots, u_{i-k}$ 的 $k+1$ 个数据，称“$k+1$”步法．当 $k = 0$ 时为单步法，$k > 0$ 时为多步法．又因为关于 u_i 和 u'_i 是线性的，故称“线性多步法”．当 $\beta_{-1} = 0$ 时为显式多步法，$\beta_{-1} \neq 0$ 为隐式多步法．

1. 显式多步法（Adams 外插法）

已知：$u_{i-k}, \cdots, u_{i-1}, u_i$，求 $u(t)$ 在 $t_{i+1} = t_0 + (i+1)h$ 处的近似值 u_{i+1}，式（1.1）的等价形式为

$$u(t_{i+1}) = u(t_i) + \int_{t_i}^{t_{i+1}} u'(t)\mathrm{d}t = u(t_i) + \int_{t_i}^{t_{i+1}} f(t, u(t))\mathrm{d}t \tag{1.27}$$

通过 $k+1$ 个数据点 $(t_{i-k}, u_{i-k}), \cdots, (t_{i-1}, u_{i-1}), (t_i, u_i)$ 作 $f(t, u(t))$ 的 Lagrange 型插值多项式 $L_{i,k}(t)$，插值余项 $r_{i,k}(t)$，即

$$f(t, u(t)) = L_{i,k}(t) + r_{i,k}(t) \tag{1.28}$$

将式（1.28）代入式（1.27），得

$$u(t_{i+1}) = u(t_i) + \int_{t_i}^{t_{i+1}} L_{i,k}(t)\mathrm{d}t + \int_{t_i}^{t_{i+1}} r_{i,k}(t)\mathrm{d}t$$

舍去 $R_{i,k} = \int_{t_i}^{t_{i+1}} r_{i,k}(t)\mathrm{d}t$，令 $u_i = u(t_i)$，

$$u_{i+1} = u_i + \int_{t_i}^{t_{i+1}} L_{i,k}(t)\mathrm{d}t \tag{1.29}$$

式中，$t_i \leqslant t \leqslant t_{i+1}$ 不在 $[t_{i-k}, t_i]$ 内，称式（1.29）为外插公式．

利用 Newton 后插公式

$$L_{i,k}(t + \tau h) = u'_i + \frac{\tau}{1!}\nabla u'_i + \frac{\tau(\tau+1)}{2!}\nabla^2 u'_i + \cdots + \frac{\tau(\tau+1)\cdots(\tau+k-1)}{k!}\nabla^k u'_i$$

式中，$\nabla u'_i = u'_i - u'_{i-1}$．

引入广义二次项系数

$$\binom{s}{j} = \frac{s(s-1)\cdots(s-j+1)}{j!} \quad 和 \quad \binom{s}{0} = 1$$

则式（1.29）可写作