第3章 连续系统定量仿真算法

连续系统定量仿真算法，通常是指运用数字计算机求解 n 阶常微分方程或偏微分方程所描述的动态系统模型，包括非线性环节和纯滞后环节等，最终获取对应某一初始条件或边值条件下的数值解或解析解，即时域瞬态响应数据或解的函数表达式。解析解的获取比较困难，对于有些微分方程相当困难。与理论分析不同，工业应用主要关注的是时域瞬态响应数据，而且绝大多数的计算机数值算法所得到的结果都是时域瞬态响应离散化数值解，也就是对应不同时间点上的状态变量的数值，本章主要介绍此类实用方法。

本章主要内容包括连续系统定量仿真算法分类；高阶常微分方程数值解法和续贯模块法仿真算法。由于连续系统定量仿真算法可以找到较多的参考书和计算程序，因此本章内容以简明扼要为特点，更为详细的内容可查阅有关书籍。学习本章内容要求具备数值计算和运用 C++语言编程等基础知识。

3.1 连续系统定量仿真算法分类

计算机化 n 阶常微分方程所描述的动态连续系统仿真数值解法主要有三类，第一类是解析算法，目前比较成熟的只有数字拉氏反变换法。比较精确的一种是以数字方法仿照人工拉氏反变换的步骤计算，得到解析解表达式。解析解的优点是能够对动态模型的瞬态响应及模型解的全部规律进行精确数学描述，便于分析研究。缺点是程序烦琐，计算费时，不能求解非线性问题。

第二类是数值积分算法，这类方法应用最广，种类繁多，发展较快。数值积分算法基于状态方程，也可以看做一阶微分方程组。解算思路可归结为，将仿真时间离散为一系列时间间隔，已知前一时刻的状态向量值，估算下一时刻的状态向量值。最经典的方法是欧拉法、四阶龙格-库塔法，其中四阶龙格-库塔法应用很广。目前求解刚性方程较有效的方法是隐式吉尔法及其改进方法。

第三类方法称为离散时间状态方程解法，又称状态转移矩阵或离散相

似法，在系统仿真程序中应用较多。另外，还有许多实时、快速仿真算法，如时域矩阵法、增广矩阵法、替换法、根匹配法、可调参数积分法等。

本章首先介绍常微分方程的数值解法作为预备知识，然后重点讨论两个连续系统仿真方法程序实现问题。这两个仿真方法分别是：运用数值积分解法面向微分方程的数字仿真方法，又称为联立方程法；运用离散相似解法面向对象的数字仿真方法，又称为序贯模块法。

3.2 高阶常微分方程数值解法

3.2.1 数值积分原理

本节旨在理清数值积分算法实际应用中的若干基本概念及需要注意的问题，因此对有关理论问题不作深入的探讨。

1. 从欧拉法看数值积分解法的特点

欧拉法是最简单的一种数值积分法，虽然它的精度较差，但能说明基本概念，当选取的积分步长较小时，可以提高欧拉法精度。因此，对于不太复杂的微分方程或计算机的速度有足够的余地等场合还经常采用此种方法。

考虑一阶微分方程的表达通式

$$\dot{x}(t) = f(t,x) \tag{3-1}$$

已知初值 $t = t_0$ 时有 $x(t_0)=x_0$，因此可得

$$\dot{x}(t_0) = f(t_0,x_0) \tag{3-2}$$

设 $t_1=h$ 是足够小的时间间隔，称 h 为积分步长，用差商代替微商，则有近似关系

$$\frac{x(t_1)-x(t_0)}{h} \approx \dot{x}(t_0) = f(t_0,x_0) \tag{3-3}$$

整理上式则有

$$x(t_1) = x(t_0)+hf(t_0,x_0) \tag{3-4}$$

即，从 $x(t_0)$ 递推计算出第一个积分步长 h 时刻的近似值 $x(t_1)$。利用 $x(t_1)$ 及 $f(t_1,x_1)$ 又可以递推计算出两个积分步长 $2h$ 时刻的近似值 $x(t_2)=x(2h)$

$$x(t_2) = x(t_1)+hf(t_1,x_1) \tag{3-5}$$

推而广之，可以得到在任一时间点 $t_{n+1}=(n+1)h$ 处 $x(t_{n+1})$ 近似值的一般表达式

$$x_{n+1} = x_n + hf(t_n,x_n) \tag{3-6}$$

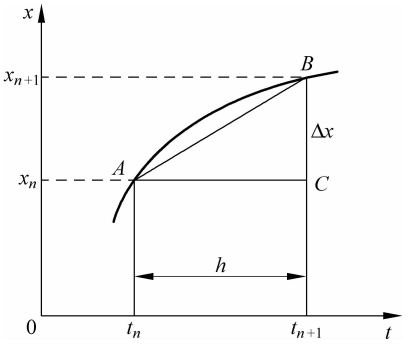

图 3-1 欧拉法的几何意义

欧拉法的几何意义可以用图 3-1 表示。图中线段 AB 的斜率即导数$\dot{x}_n$。

$$\dot{x}_n = f(t_n, x_n) \approx \frac{\Delta x}{h} \tag{3-7}$$

于是，$\Delta x = hf(t_n, x_n)$。

由 x_n 估计 x_{n+1}的值可以通过 x_n 加上一个校正量来近似，则

$$\begin{aligned} x_{n+1} &= x_n + \Delta x \\ &= x_n + hf(t_n, x_n) \end{aligned} \tag{3-8}$$

从欧拉法可以看出，数值积分都是在离散点上进行，即把时间变量划分为许多小段，每一段的时间间隔 h 称为时间步长。如果 $x(t)$称为状态，那么求解问题描述为，已知前一时刻的状态，设法递推估计出下一时刻的状态。直到把全部离散点上的状态求出，就可以最终得到瞬态响应的值表。递推估计有多种方法，也就导致了各种各样的数值积分方法出现。

2. 四阶龙格-库塔法的几何意义

四阶龙格-库塔法在递推估计算法上对欧拉法进行了有效改进，采用了平均值校正的方法，因此精度有了很大的提高。四阶龙格-库塔法的递推估计公式如下

$$x_{n+1} = x_n + (K_1 + 2K_2 + 2K_3 + K_4)/6 \tag{3-9}$$

式中

$$\begin{aligned} K_1 &= hf(t_n, x_n) \\ K_2 &= hf(t_{n+1/2}, x_n + K_1/2) \\ K_3 &= hf(t_{n+1/2}, x_n + K_2/2) \\ K_4 &= hf(t_{n+1}, x_n + K_3) \end{aligned}$$

为了说明几何意义将式(3-9)改写为

$$x_{n+1} = x_n + \Delta x \tag{3-10}$$

式中

$$\Delta x = (K_1 + 2K_2 + 2K_3 + K_4)/6$$

Δx 即平均值校正量，见图 3-2，仍然沿用前述欧拉法的几何意义，以 Δx 的含义分别解释校正量 K_1、K_2、K_3 及 K_4。图 3-2 中线段 AB 的斜率为①、AC 斜率为②、AD 斜率为③、AE 斜率为④。

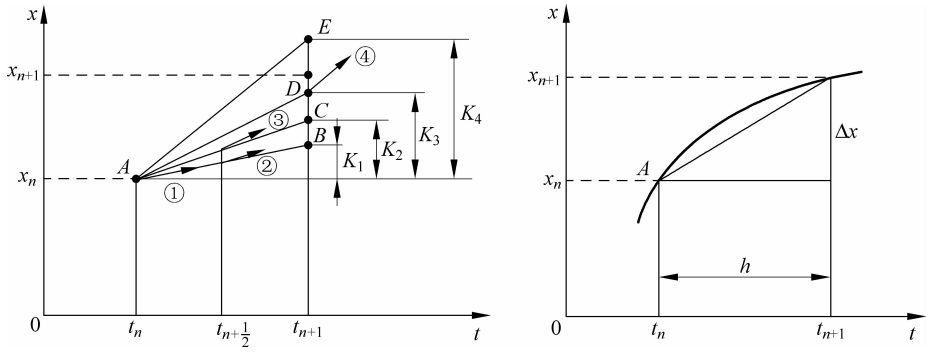

图 3-2　四阶龙格-库塔法的几何意义

3.2.2 欧拉法及四阶龙格-库塔法积分公式

为了说明原理，前文仅讨论了单个一阶微分方程的问题。高阶微分方程最终都可以转化为一阶微分方程组的形式，见式(3-11)。只要把一阶算法直接推广就可以得到高阶微分方程解法公式。下面详细列写欧拉法及四阶龙格-库塔法公式步骤。

设一阶微分方程组的一般形式为

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_n\end{bmatrix}=\begin{bmatrix}f_1(t,x_1,x_2,\cdots,x_n)\\ f_2(t,x_1,x_2,\cdots,x_n)\\ \vdots\\ f_n(t,x_1,x_2,\cdots,x_n)\end{bmatrix} \tag{3-11}$$

时间步长取为 h，$t=t_0$ 时的初始值是

$$\begin{bmatrix}x_1(t_0)\\ x_2(t_0)\\ \vdots\\ x_n(t_0)\end{bmatrix}=\begin{bmatrix}x_1(0)\\ x_2(0)\\ \vdots\\ x_n(0)\end{bmatrix}$$

若其解表示成

$$\boldsymbol{x}=\begin{bmatrix}x_1(t)\\ x_2(t)\\ \vdots\\ x_n(t)\end{bmatrix}$$

则第 m 个时间步长(即 $t=t_m=t_0+mh$ 时刻)上的解为

$$\boldsymbol{x}(m)=\begin{bmatrix}x_1(m)\\ x_2(m)\\ \vdots\\ x_n(m)\end{bmatrix}$$

已知 t_m 时刻状态递推估计 $t_{m+1}=t_m+h$ 时刻状态的欧拉法公式为

$$\begin{bmatrix}x_1(m+1)\\ x_2(m+1)\\ \vdots\\ x_n(m+1)\end{bmatrix}=\begin{bmatrix}x_1(m)\\ x_2(m)\\ \vdots\\ x_n(m)\end{bmatrix}+h\begin{bmatrix}f_1(t_m,x_1(m),x_2(m),\cdots,x_n(m))\\ f_2(t_m,x_1(m),x_2(m),\cdots,x_n(m))\\ \vdots\\ f_n(t_m,x_1(m),x_2(m),\cdots,x_n(m))\end{bmatrix} \tag{3-12}$$

四阶龙格-库塔法公式为

$$\begin{bmatrix}x_1(m+1)\\ x_2(m+1)\\ \vdots\\ x_n(m+1)\end{bmatrix}=\begin{bmatrix}x_1(m)\\ x_2(m)\\ \vdots\\ x_n(m)\end{bmatrix}+\frac{1}{6}\left[\begin{bmatrix}k_{11}\\ k_{21}\\ \vdots\\ k_{n1}\end{bmatrix}+2\begin{bmatrix}k_{12}\\ k_{22}\\ \vdots\\ k_{n2}\end{bmatrix}+2\begin{bmatrix}k_{13}\\ k_{23}\\ \vdots\\ k_{n3}\end{bmatrix}+\begin{bmatrix}k_{14}\\ k_{24}\\ \vdots\\ k_{n4}\end{bmatrix}\right] \tag{3-13}$$

式中

$$\begin{pmatrix}k_{11}\\k_{21}\\\vdots\\k_{n1}\end{pmatrix}=h\begin{bmatrix}\begin{pmatrix}f_1(t_m,x_1(m),x_2(m),\cdots x_n(m))\\f_2(t_m,x_1(m),x_2(m),\cdots x_n(m))\\\vdots\\f_n(t_m,x_1(m),x_2(m),\cdots x_n(m))\end{pmatrix}\end{bmatrix}$$

$$\begin{pmatrix}k_{12}\\k_{22}\\\vdots\\k_{n2}\end{pmatrix}=h\begin{bmatrix}\begin{pmatrix}f_1\left(t_{m+1/2},x_1(m)+\frac{k_{11}}{2},x_2(m)+\frac{k_{21}}{2},\cdots,x_n(m)+\frac{k_{n1}}{2}\right)\\f_2\left(t_{m+1/2},x_1(m)+\frac{k_{11}}{2},x_2(m)+\frac{k_{21}}{2},\cdots,x_n(m)+\frac{k_{n1}}{2}\right)\\\vdots\\f_n\left(t_{m+1/2},x_1(m)+\frac{k_{11}}{2},x_2(m)+\frac{k_{21}}{2},\cdots,x_n(m)+\frac{k_{n1}}{2}\right)\end{pmatrix}\end{bmatrix}$$

$$\begin{pmatrix}k_{13}\\k_{23}\\\vdots\\k_{n3}\end{pmatrix}=h\begin{bmatrix}\begin{pmatrix}f_1\left(t_{m+1/2},x_1(m)+\frac{k_{12}}{2},x_2(m)+\frac{k_{22}}{2},\cdots,x_n(m)+\frac{k_{n2}}{2}\right)\\f_2\left(t_{m+1/2},x_1(m)+\frac{k_{12}}{2},x_2(m)+\frac{k_{22}}{2},\cdots,x_n(m)+\frac{k_{n2}}{2}\right)\\\vdots\\f_n\left(t_{m+1/2},x_1(m)+\frac{k_{12}}{2},x_2(m)+\frac{k_{22}}{2},\cdots,x_n(m)+\frac{k_{n2}}{2}\right)\end{pmatrix}\end{bmatrix}$$

$$\begin{pmatrix}k_{14}\\k_{24}\\\vdots\\k_{n4}\end{pmatrix}=h\begin{bmatrix}\begin{pmatrix}f_1(t_{m+1},x_1(m)+k_{13},x_2(m)+k_{23},\cdots,x_n(m)+k_{n3})\\f_2(t_{m+1},x_1(m)+k_{13},x_2(m)+k_{23},\cdots,x_n(m)+k_{n3})\\\vdots\\f_n(t_{m+1},x_1(m)+k_{13},x_2(m)+k_{23},\cdots,x_n(m)+k_{n3})\end{pmatrix}\end{bmatrix}$$

如果把以上公式写成向量式，则形式和一阶方程完全一样。还可以触类旁通，用以上方法去理解其他数值积分公式的详细计算步骤。

3.2.3　几个基本概念

1. 非线性微分方程

非线性系统的特征是系统不满足叠加原理，系统的输入输出比取决于输入的幅值和形式。例如，一个非线性系统对不同幅值的阶跃输入可能有完全不同的响应。描述这类系统的微分方程称为非线性微分方程。从方程的特点上看变量之间的关系不是一次关系，可能出现自变量之间的其他关系，例如乘幂、各阶导数相互乘积等。例如，非线性力学中常讨论的杜芬(Duffing)方程

$$m\ddot{x}+f\dot{x}+kx+k'x^3=0 \tag{3-14}$$

万达波尔方程

$$m\ddot{x}-f(1-x^2)\dot{x}-kx=0 \tag{3-15}$$

都是非线性微分方程。或者在线性环节中插入典型的非线性环节，如饱和、失灵区、滞环、间隙、静摩擦、流体可压缩等固有非线性后，这样的系统也就成为非线性系统

了。图 3-3 是具有滞环的非线性系统。

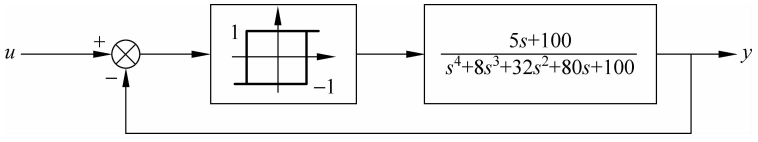

图 3-3 具有滞环的非线性系统

这类方程的解析解很难得到，但数值积分方法几乎都可以用来直接求解非线性微分方程。

2. 刚性微分方程

刚性微分方程又称为病态方程。这类微分方程概念是，对于一般常微分方程

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = f_i(t, x_1, x_2, \cdots, x_n) \quad (i = 1, 2, \cdots, n) \tag{3-16}$$

其物理特征通常由雅可比矩阵$(\alpha_{ij}) = \left(\dfrac{\partial f_i}{\partial x_j}\right)$的特征值$\lambda_i = \alpha_i + \mathrm{j}\beta_i$来表达。

特征值实部 $\mathrm{Re}\,\lambda_i$ 表示增长($\alpha_i > 0$)或者衰减($\alpha_i < 0$)因子，虚部 $I_m\lambda_i = \beta_i$ 表示周期振荡频率。而$|\mathrm{Re}\,\lambda_i|$表示物理系统时间常数。定义刚性比为

$$\rho = \max_{1 \leqslant i \leqslant n} |\,\mathrm{Re}\,\lambda_i\,| \,/\, \min_{1 \leqslant i \leqslant n} |\,\mathrm{Re}\,\lambda_i\,|$$

当 $\rho \gg 1$ 时，系统就称为刚性的。例如有些控制系统、电子网络、精馏塔模型等系统中部分动态数学模型的 ρ 可能达到 10^4 以上。

数值处理这类方程的困难在于，最大时间常数 T 决定解题总时限，而最小时间常数 τ 决定积分步长，因此解题非常耗时，高阶问题更为严重。以龙格-库塔法为例，积分步长 h 必须满足 $h < 2.7\tau$，若 $T = 1$，则积分步数 $N > \dfrac{1}{2.7}10^6$。如果系统阶次很高，每积分一步计算量已经可观，完成如此多步计算所耗时间就更无法估量。目前求解刚性方程较好的方法是变阶变步长吉尔方法。

3. 混合方程

混合方程是指常微分方程与非线性代数方程的混合体，机理建模往往采用这种形式。如大规模集成电路的动态模型，其状态向量不仅含有相当于微分方程的变量，也包括相应于代数方程的变量。又如精馏塔机理建模，在微分方程中常插有静态工艺和物性计算的非线性代数方程。

混合方程可采用后面介绍的序贯模块仿真方法处理，或用隐式积分方法处理。

4. 显式和隐式积分法

显式积分算法特点是，下一时刻状态值由前一时刻已求出的状态所确定。而隐式方法不然，其状态变量是自身的函数。以最简单的欧拉法为例，显式欧拉积分公

式为

$$x_{m+1} = x_m + h\dot{x}_m \tag{3-17}$$

隐式欧拉法有多种形式，如梯形积分公式

$$x_{m+1} = x_m + \frac{h}{2}(\dot{x}_{m+1} + \dot{x}_m) \tag{3-18}$$

矩形积分公式

$$x_{m+1} = x_m + h\dot{x}_{m+1} \tag{3-19}$$

关于显式及隐式更一般的情况，可以通过下面的例子说明。

设微分方程通式

$$\dot{x} = f(t,x) \tag{3-20}$$

可以导出差分方程近似表达式

$$x_m = a_1 x_{m-1} + a_2 x_{m-2} + \cdots + h[b_0\dot{x}_m + b_1\dot{x}_{m-1} + b_2\dot{x}_{m-2} + \cdots] \tag{3-21}$$

式中系数 a_i 和 b_i 一般用 $x(t)$ 在某点附近的台劳展开式的各种匹配方法确定。如果 $b_0=0$，这个公式就是显式积分，因为公式右端所有数据都已由前面的积分运算得出。但是，如果 $b_0\neq 0$，$\dot{x}_m$ 出现在公式的右端，要得到 x_m 必须计算 $\dot{x}_m$。必然涉及求解联立 x_m 和 $\dot{x}_m$ 的方程组问题。这种方程组可能是非线性的，则要用迭代方法求解，因此，每积分一步都必须解一次代数方程组。但隐式方法稳定性好，步长可以大大超过最小时间常数，求解刚性方程特别有效。为了说明这个问题，仍以欧拉法为例。

已知一阶常微分方程

$$\tau\dot{x} + x = u \tag{3-22}$$

式中　τ 是常数，$x=x(t)$，$u=u(t)$ 为外作用函数，设 h 为积分步长，由式(3-22)可变换得

$$\dot{x}_m = \frac{1}{\tau}(u_m - x_m) \tag{3-23}$$

代入式(3-17)

$$x_{m+1} = x_m + \frac{h}{\tau}(u_m - x_m) \tag{3-24}$$

合并同类项化简后得到显式解法通式。

$$x_{m+1} = \left(1 - \frac{h}{\tau}\right)x_m + \frac{h}{\tau}u_m \tag{3-25}$$

再看隐式矩阵公式的情况，将式(3-23)改写为

$$\dot{x}_{m+1} = \frac{1}{\tau}(u_{m+1} - x_{m+1})$$

代入式(3-19)得

$$x_{m+1} = x_m + \frac{h}{\tau}(u_{m+1} - x_{m+1})$$

移项

$$x_{m+1} + \frac{h}{\tau}x_{m+1} = x_m + \frac{h}{\tau}u_{m+1}$$

合并

$$\left(1 + \frac{h}{\tau}\right)x_{m+1} = x_m + \frac{h}{\tau}u_{m+1}$$

得到隐式解法通式

$$x_{m+1} = \left(\frac{1}{1+h/\tau}\right)x_m + \left(\frac{h/\tau}{1+h/\tau}\right)u_{m+1} \tag{3-26}$$

可以证明式(3-25)及式(3-26)的数值稳定性完全取决于等式右端第一项系数，若超过±1，解法变为不稳定。结论表达为

$$\left|\frac{1}{1+h/\tau}\right| \leqslant 1 \qquad |1-h/\tau| \leqslant 1$$

隐式　　显式

显式稳定步长范围　$0 \leqslant h/\tau \leqslant 2$

隐式稳定步长范围　$0 \leqslant h/\tau$

显然，隐式方法对于这个一阶方程已取消了对步长 h 的限制，即 h 取值较大也可能保证解法的稳定性。

5. 截断误差、舍入误差及稳定性

数值解法是一种近似的解法，近似解的误差首先是由于差商代替微商，且 $\Delta t = h$ 又不能无限小所致，如欧拉法就是用一阶差商代替微商的方法，这种近似替代产生的误差，称为截断误差。计算机采用的字长有限，在计算机中由数值舍入引起的误差称为舍入误差。必须考虑最初产生的误差在以后计算中是否会被无限扩大，这种问题称为数值解法的稳定性。

一般稳定性与步长有密切关系，可用下面的简单例子说明。已知一阶线性微分方程

$$\frac{\mathrm{d}x}{\mathrm{d}t} = 1 - x$$

初始条件 $t=0$，$x=0$，精确解是

$$x(t) = 1 - \mathrm{e}^{-t}$$

选择不同的时间步长，分别用欧拉法和四阶龙格-库塔法求解，将结果描绘在图 3-4 及图 3-5 中。

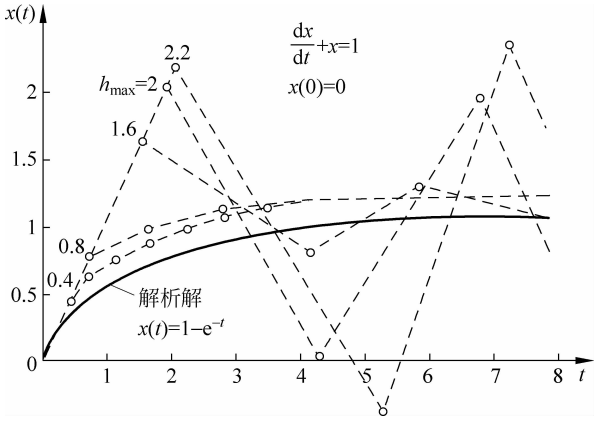

图 3-4　欧拉法的数值稳定性

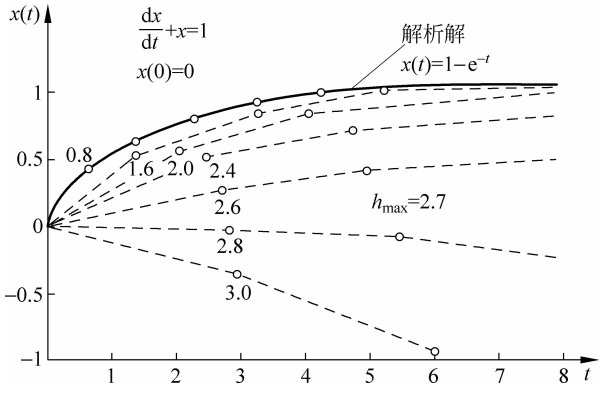

图 3-5　四阶龙格-库塔法的数值稳定性

可以明显地看出随着步长加大数值解法越来越不稳定。实际计算中出现不稳定情况的特征是结果发散，甚至很快造成计算机溢出。欧拉法积分出现不稳定的临界步长$(h)_{\max}=2$。四阶龙格-库塔法积分出现不稳定的临界步长$(h)_{\max}=2.7$。

6. 标准初始条件

微分方程的通解由曲线族组成，必须给定初始条件，才能得到曲线族中的一条特定曲线。仿真计算绝大多数情况都选用标准初始条件。所谓标准初始条件，即时间从 0 开始，$t_0=0$，全部状态变量、输出量及输入量对时间的各种导数也都定为 0。因为 $t_0=0$ 时系统处于静止状态，这样选定是符合物理概念的，同电模拟机积分器初值设定完全一致。采用统一的标准初始条件便于比较不同动态系统的瞬态响应。

7. 步长选取

采用定步长数值积分算法，合理选取步长很重要，同时也是比较复杂的问题。步长选取与计算方法、解题稳定性、精度要求、耗机时间、是否需要实时仿真等因素都有关，选择时应权衡利弊综合考虑。

如果采用定步长数值积分方法，实际计算时可能要进行必要的试算，或采用高精度方法作一定的平行检测。因为大部分问题在计算前对于系统特征、过渡时间、各时间常数确定值不一定了解，通常选用尽可能小的步长计算。但阶次较高的复杂系统，为了节省机时应当探索确保稳定性及必要精度的前提条件下尽可能大的步长。

采用变步长算法，能根据要求的精度在计算中自动改变步长，能够提高效率、减少人工试探的盲目性。

3.2.4　常用数值积分方法及选择

到目前为止还没有找到一种理想的、可以适用各种要求的数值积分算法，所以，许多仿真软件备有许多平行算法供选择。几种常用数值积分解法及比较见表 3-1。

从表 3-1 中所列的常用解法可以看出，速度快的、简单的算法精度及稳定性较差；精度及稳定性好的方法计算量都较大，程序也比较复杂。选用时应当因问题而论。大量的一般性问题多采用定步长四阶龙格-库塔法，因为这种方法简单、精度较高、稳定性较好，步长足够小时也可以适应刚性方程。如果精度要求不高、方程阶次很高，又要快速仿真，也还有用欧拉法和二阶龙格-库塔法的。非线性刚性方程用吉尔法较好。大规模集成电路瞬态分析，几乎包括了对数值积分解法的各种特殊要求，即刚性、非线性、超高阶、高精度、混合方程等。据报道，采用改进隐式吉尔方法，配合大型稀疏阵技术可以解决这类问题。

表 3-1 几种常用数值积分解法及比较

方法名称	优点	缺点	使用范围
欧拉法	程序简单、快速	精度低，稳定性差	高阶、低精度、实时仿真
二阶龙格-库塔法	程序简单、快速	精度低，稳定性差	高阶、低精度、实时仿真
定步长四阶龙格-库塔法	程序简单，精度较高，小步长可解刚性方程	高阶较费时	非刚性、弱非线性、中等精度的普通方程
变步长四阶龙格-库塔法	可以保证满足指定精度	高阶较费时	同上，仿真的实时性可能好一些
拉姆伯特五阶方法	高精度	计算量大，高阶费时，不能解刚性方程	高精度计算校验用
默森单步法	精度高	计算量大，高阶费时，不能解刚性方程	高精度计算校验用
外插法	精度高	稳定性不高	高精度计算校验用
阿当姆斯预报-校正法	可以估计误差，稳定性较好	不能解刚性方程	高阶光滑非刚性方程
小参数法	简单，变步长，速度快，可解刚性方程	精度中等	高阶刚性方程实时仿真
特雷纳法	可解刚性方程	只适于特殊情况	特殊刚性方程
吉尔法	变阶，变步长，适用于非线性刚性方程	占用容量大，高阶问题困难	非线性刚性方程
改进吉尔法	变阶，变步长，适于非线性刚性及混合方程	程序复杂，须解非线性代数方程组	高阶非线性刚性混合方程

表 3-1 仅列出了几种常用方法，在文献中大部分可以得到源程序。此外，如龙格-库塔的各种改进方法，隐式单步法、改进默森法等，在文献中均可查到，这里就不一一列举了。

3.3 联立方程法仿真算法

本方法从具有纯滞后的 n 阶传递函数出发，即式(3-27)。先由程序自动将式(3-27)转换成状态方程，然后运用定步长四阶龙格-库塔法求出阶跃、斜坡、加速度或正弦干扰信号的瞬态响应值表及曲线。

3.3.1　计算方法

设 n 阶传递函数的通式

$$G(s)=\frac{B_0s^m+B_1s^{m-1}+\cdots+B_{m-1}s+B_m}{A_0s^n+A_1s^{n-1}+\cdots+A_{n-1}s+A_n}\cdot e^{-\tau s}\quad(n\geqslant m)\tag{3-27}$$

式中　s——拉氏算子；

A_i——s 域多项式分母系数；

B_i——s 域多项式分子系数；

τ——纯滞后时间。

将式(3-27)转换成微分方程的推导如下。由于

$$G(s)=\frac{B_0s^m+B_1s^{m-1}+\cdots+B_{m-1}s+B_m}{A_0s^n+A_1s^{n-1}+\cdots+A_{n-1}s+A_n}\cdot e^{-\tau s}=\frac{y(s)}{u(s)}\quad(n\geqslant m)$$

则

$$\begin{aligned}&(A_0s^n+A_1s^{n-1}+\cdots+A_{n-1}s+A_n)y(s)\\&=(B_0s^m+B_1s^{m-1}+\cdots+B_{m-1}s+B_m)\cdot e^{-\tau s}\cdot u(s)\end{aligned}$$

于是有

$$\begin{aligned}&A_0y^n(t)+A_1y^{n-1}(t)+\cdots+A_{n-1}\dot{y}(t)+A_ny(t)\\&=B_0u^m(t-\tau)+B_1u^{m-1}(t-\tau)+\cdots+B_{m-1}\dot{u}(t-\tau)+B_mu(t-\tau)\end{aligned}\tag{3-28}$$

为了应用龙格-库塔法求解 n 阶微分方程式(3-28)，必须对式(3-28)作适当形式上的改变以适应于龙格-库塔法的数值计算程序。首先考虑四阶龙格-库塔法的求解步骤。

对于已知初始条件的一阶微分方程组

$$\begin{aligned}&\frac{dy_i}{dt}=f_i(t,y_1,y_2,\cdots,y_n)\\&y_i(t_0)=y_{i0}\quad(i=1,2,3,\cdots,n)\end{aligned}\tag{3-29}$$

已知前一时刻 t_k 及 $y_{ik}(i=1,2,\cdots,n)$以及步长 h 即 Δt，求下一时刻 t_k+h 的 y_{ik+1}四阶龙格-库塔法公式如下

$$\begin{aligned}k_{i1}&=f_i(t_k,y_{1k},\cdots,y_{nk})\\k_{i2}&=f_i(t_k+h/2,y_{1k}+hk_{11}/2,\cdots,y_{nk}+k_{n1}h/2)\\k_{i3}&=f_i(t_k+h/2,y_{1k}+hk_{12}/2,\cdots,y_{nk}+k_{n2}h/2)\\k_{i4}&=f_i(t_k+h,y_{1k}+hk_{13},\cdots,y_{nk}+hk_{n3})\\y_{ik+1}&=y_{ik}+h/6(k_{i1}+2k_{i2}+2k_{i3}+k_{i4})\quad(i=1,2,3,\cdots,n)\end{aligned}\tag{3-30}$$

以上递推公式是从初始时刻 t_0 开始，以步长 h 为增量，每增加一个 h，重复一次以上的递推公式，同时用前一次求出的 $y_{ik}(i=1,2,3,\cdots,n)$作为下一时刻 y_{ik+1}的初始值，这样一直计算到要求的终止时间为止。

从以上步骤可以看出，为了应用龙格-库塔法求解 n 阶微分方程，必须设法把其