控制系统计算机辅助设计

吴重光　沈承林　编著

机 械 工 业 出 版 社

控制系统计算机辅助设计

吴重光　沈承林　编著

*

机械工业出版社出版（北京阜成门外百万庄南里一号）

（北京市书刊出版业营业许可证出字第 117 号）

重庆印制一厂印刷

新华书店北京发行所发行 · 新华书店经售

*

开本 787×1092 1/16 · 印张 16¾ · 字数 416 千字

1986年 8 月重庆第一版 · 1986年 8 月重庆第一次印刷

印数 0.001—5.000 · 定价 3.95 元

*

统一书号：15033·6270

前　言

控制系统计算机辅助设计在我国是新兴领域，国际上也仅有二十年历史。这门新技术为控制理论与实际应用搭起桥梁，使在设计中大量应用理论进行定量分析成为可能。电子计算机运算速度快、数值计算精确，能大大提高设计效率及设计质量。计算机辅助设计系统是一种人——机联合系统，除了发挥计算机高速度、高效率与多功能优势外，还可以发挥设计人员本身的经验和智能。这种系统使用简便，一般中级技术人员就能操作。软件的移植和维护都比较容易。这门新技术不但广泛应用于设计，还可以辅助科研，辅助培训设计人员以及控制理论教学。因此，控制系统计算机辅助设计技术越来越受到重视，发展十分迅速。

近年来，我国电子计算机推广应用出现了可喜局面，特别是微型电子计算机配置数量有较大增长。设计、科研及高等教学部门对控制系统计算机辅助设计软件和有关知识的需求日渐迫切。在这种形势下，1983年3月由中国机械工程学会全国高等学校工业自动化专业教育委员会委托，我们举办了控制系统计算机辅助设计学习班。学习班采用的讲义就是本书的前身。讲义内容是我们自1976年以来进行算法研究和软件开发的成果。学习班之后，我们的工作受到普遍重视，得到许多同志的热情鼓励与帮助。在此基础上我们对讲义作了较大修改，对计算程序也进行了改进与增补，写成本书。

控制系统计算机辅助设计包括的内容很广，限于作者的水平及篇幅，我们不可能面面俱到。本书着重论述连续系统数字仿真、线性系统频域分析和图论技术实用计算方法及程序实现，相当于把经典控制理论实现了计算机化，并且借助于图论技术将线性单变量算法程序推广到线性多变量系统。这些内容可以说是控制理论中最基础，应用最广泛的部分。从软件的移植情况看，能够使用这套程序的领域有航天、航空、武器、电子、电力、交通、冶金、石油化工、轻工、建材等。

本书所需要的基础知识有：调节原理、现代控制论、计算方法、算法语言及图论技术。

书中介绍的程序汇集在一起，组成了一套控制系统分析软件。命名为CSAP 软件系统。全书也可以说是这套软件系统的详尽说明书。

CSAP软件开发过程中，曾得到清华大学方崇智教授、北京轻工业学院夏德钤教授，化工部设计公司万学达高级工程师、北京化工学院麻德贤副教授等专家指导。还得到化工部设计公司计算站、兰化公司设计院计算站及北京化工学院计算站大力协助。

本书由清华大学郑维敏教授主审，给我们提出许多宝贵的意见和帮助。在此对他们致以衷心感谢。

由于作者水平有限，全书尚有许多不足之处，恳请读者批评指正。

沈承林　吴重光

1984年6月于北京化工学院

目　录

第一章 导 论

控制系统计算机辅助设计常简写成控制系统CAD。其中CAD是英文Computer Aided Design的缩写。广义理解，控制系统CAD可以描述为控制系统设计活动中对计算机的任何一种应用。J. J. Allan Ⅲ[1]将CAD系统归纳为四个部分组成。控制系统CAD也可以引用相同的概念，这四个部分是：

设计者——用户；

硬 件——计算机及其外部设备；

软 件——计算机系统软件和控制系统CAD应用软件；

问 题——所要解决的设计课题及其数学模型。

因此更确切地说，控制系统CAD是包含计算机硬件、软件及设计者的人-机联合系统。在这个系统中，设计者与计算机相互对话并共同有效地工作，完成预想的设计任务。

控制系统CAD应用软件是最基础的部分。这种软件能够完成数据处理、系统分析、系统综合、系统仿真、优化计算、结果处理等任务。其中系统分析及系统仿真软件又是使用最频繁、最核心的部分。利用电子计算机进行控制系统分析及系统仿真，必须将控制理论相应的内容计算机化。也就是把控制理论常用于设计的内容转换成有效的数值计算方法，并编写成应用程序。近来，这种软件[2][3]国外多有发表。

从第二章开始，我们将用大量篇幅介绍一套控制系统分析和系统仿真基本程序。这套程序命名为CSAP软件系统。CSAP是英文Control System Analysis Program字头缩写，全称控制系统分析程序。程序采用FORTRAN语言，并且尽量不超出FORTRAN-Ⅳ的规定。因此备有FORTRAN编译系统的电子计算机都可以移植应用。为了讲解原理及便于移植，软件不设管理程序，而是将程序按功能分立成若干主程序。每个程序只要32kB容量就可以运行。用户可以利用这些程序或稍加改造后任意组合自己的控制系统CAD软件，以及计算机辅助教学软件。使用方式为批处理，很容易实现人-机会话。CSAP软件共有44个子程序，详见附录一。总规模为1800条FORTRAN语句，程序清单详见附录二。

CSAP软件主要具有以下内容和功能：

图论程序：用于简化复杂单输入/单输出系统及多输入/多输出系统信号流图。能够从信号流图直接获取规格化的传递函数或传递矩阵。

稳定判据程序：运用劳斯判据判定线性系统的稳定性。配合图论程序可以判定多输入/多输出系统稳定性。

频率响应程序：能够精确求得增益裕量、相位裕量及频率响应值表。打印伯德图、奈奎斯特图。配合图论程序还可以求多输入/多输出系统的伯德或奈奎斯特阵列。

根轨迹程序：以最优化算法在已知开环极、零点条件下，快速求出准确的根轨迹。此外还能准确求出分离点或汇合点、求多重极点系统的根轨迹、求具有纯滞后系统的主导根轨迹以及根轨迹与虚轴的交点。

时间响应程序：首先将传递函数转换成状态方程，然后调用定步长四阶龙格-库塔方法

求解。可以求得线性系统的阶跃、斜坡、加速度或正弦响应数值结果，并打印出响应曲线。

模型降阶程序：配合时间响应程序运用连分式降阶方法，可以进行线性系统有理降阶。

时域质量指标程序：配合时间响应程序能够求出阶跃响应曲线的积分指标(ITAE、IAE、ISE、ITSE)、过渡时间、上升时间、峰值时间、超调量及衰减比等。

连续系统仿真程序：采用离散相似原理，直接输入系统结构图进行仿真。能够仿真纯滞后系统、多输入/多输出系统。只要按§2-4提供的公式对程序稍作扩充，还可以仿真非线性系统。输出为响应值表及曲线。

s域多项式处理程序：包括传递函数多因式相乘规格化程序及多项式求根程序。

图1-1表示使用CSAP软件系统时各程序的相互关联情况。图1-1也直观地表达了全书各章内容的相互联系。

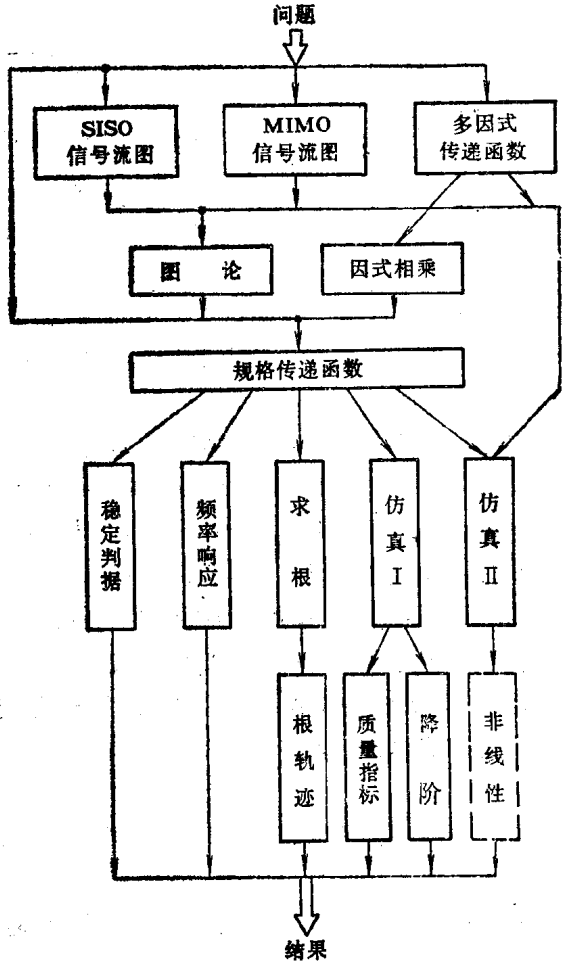

图1-1 CSAP软件系统各程序关联图

为了便于读者理解，各章节划分本着从易到难、由浅入深的原则。第二、三、四章按控制系统瞬态响应分析、频域分析与根域分析三方面，介绍经典控制理论数字计算方法。第五章引入图论技术，解决图论实用程序问题。第六章介绍基本程序的灵活应用及稳定判据、多因式相乘、多项式求根等程序。在这一章中，通过实例说明如何将基本程序相互联合、稍加修改或补充某些程序后，构造出多种功能的控制系统计算机辅助设计软件。第四章和第五章的计算原理比较复杂，如果认为难懂又急于使用程序，可以不必搞通原理，只要明白程序功能和使用说明就能上机计算。

每一种程序都给出较丰富的计算例题和计算机输出结果。利用这些例题，读者不仅可以学会程序使用方法，还可以直接用来考核移植的程序。

从第二章到第五章，各章后面附有一些练习题，主要目的是使读者熟悉这套程序的使用方法。

第二章　连续系统数字仿真方法

§2-1　引　　言

仿真技术在控制系统计算机辅助设计中主要用途是，对所有设计方法得出的结果或其中各阶段进行质量检验，它本身也是一种计算机辅助设计手段。

连续系统数字仿真方法，实质就是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，包括非线性环节和纯滞后环节等，最终获取瞬态响应。

连续系统仿真数值解法主要有三类：第一类是解析算法。只能求解线性常微分方程描述的系统，即数字拉氏反变换法。比较精确的一种[4]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。优点，解析解是瞬态响应全部规律的精确描述，便于分析研究。缺点，程序繁琐，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，也可以看作一阶微分方程组。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-库塔法应用甚广。目前求解刚性方程较有效的方法是隐式吉尔法[5][6]及其改进方法。

第三类方法称为离散时间状态方程解法。又称状态转移矩阵法或离散相似法。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[7][8]。

本章首先介绍常微分方程的数值解法作为预备知识。然后通过剖析两个连续系统仿真程序深入讨论程序实现问题。这两个程序分别是，运用数值积分解法面向微分方程的数字仿真程序Ⅰ，以及运用离散相似解法面向系统方框图的数字仿真程序Ⅱ。此外，还介绍时域分析中求阶跃响应的质量指标、打印瞬态响应曲线的程序。连分式降阶程序包括在数字仿真程序Ⅰ中，原理见§6-5。

§2-2　常微分方程的数值解法

本书目的不是专门研究理论，而是解决应用问题。因此对有关理论问题不作深入探讨。本节旨在把数值积分算法中的若干基本概念及应用中应当注意的主要问题搞清。

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。有些场合还常常用到此算法。

考虑一阶微分方程：

$$\dot{x}(t)=f(t,x) \tag{2-1}$$

已知初值，$t_0=0$时有$x(t_0)=x_0$，因此可得：

$$\dot{x}(t_0)=f(t_0,x_0)$$

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

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

整理上式则有

$$x(t_1)=x(t_0)+hf(t_0,x_0)$$

即，从$x(t_0)$递推出$x(t_1)$的近似值。利用$x(t_1)$及$f(t_1,x_1)$又可以递推出$x(t_2)=x(2h)$的近似值：

$$x(t_2)=x(t_1)+hf(t_1,x_1)$$

我们可以得到在任一点$t_{n+1}=(n+1)h$处$x(t_{n+1})$近似值的一般表达式。

$$x_{n+1}=x_n+hf(t_n,x_n) \tag{2-2}$$

欧拉法的几何意义可以用图2-1表示。图中线段$\overline{AB}$的斜率即导数$\dot{x}_n$：

$$\dot{x}_n=f(t_n,x_n)=\frac{\Delta x}{h}$$

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Δx来近似，则：

$$x_{n+1}=x_n+\Delta x=x_n+hf(t_n,x_n)$$

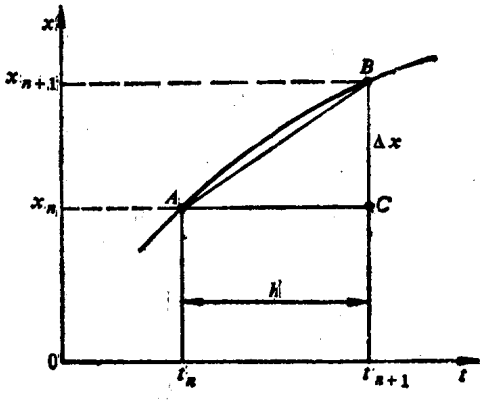

图2-1　欧拉法的几何意义

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，因而精度有了很大提高。

四阶龙格-库塔法的递推估计公式如下：

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

式中　$K_1=hf(t_n,x_n)$

$K_2=hf(t_{n+\frac{1}{2}},\ x_n+K_1/2)$

$K_3=hf(t_{n+\frac{1}{2}},\ x_n+K_2/2)$

$K_4=hf(t_{n+1},\ x_n+K_3)$

为了说明几何意义将式(2-3)改写为：

$$x_{n+1}=x_n+\overline{\Delta x}$$

式中　$\overline{\Delta x}=(K_1+2K_2+2K_3+K_4)/6$ (2-4)

$\overline{\Delta x}$即平均值校正量。见图 2-2a，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2-2a中线段$\overline{AB}$的斜率为①、$\overline{AC}$斜率为②、$\overline{AD}$斜率为③、$\overline{AE}$ 斜率为④.

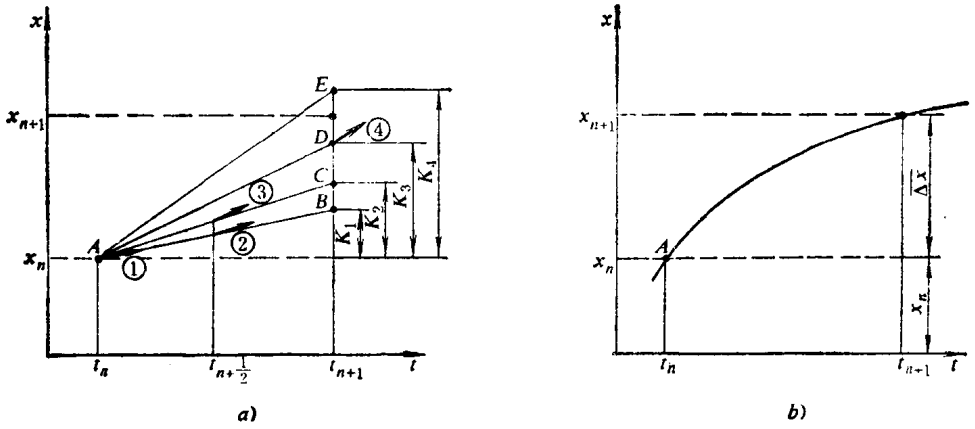

图2-2　四阶龙格-库塔法的几何意义

3. 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归结为一阶微分方程组的形式(2-5)。只要把一阶算法直接推广就可以得到解法公式。下面详细列写欧拉法及四阶龙格-库塔法公式步骤，以便搞清方程的计算过程。

设一阶微分方程组的一般形式为：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} f_1(t, x_1, x_2, \cdots x_n) \\ f_2(t, x_1, x_2, \cdots x_n) \\ \vdots \\ f_n(t, x_1, x_2, \cdots x_n) \end{bmatrix} \tag{2-5}$$

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：

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

则第m个时间步长上的解，即$t=t_m=t_0+mh$时刻的解是：

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

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：

$$\begin{bmatrix} x_1(m+1) \\ x_2(m+1) \\ \vdots \\ x_n(m+1) \end{bmatrix} = \begin{bmatrix} x_1(m) \\ x_2(m) \\ \vdots \\ x_n(m) \end{bmatrix} + h \begin{bmatrix} f_1(t_m, x_1(m), x_2(m), \cdots x_n(m)) \\ f_2(t_m, x_1(m), x_2(m), \cdots x_n(m)) \\ \vdots \\ f_n(t_m, x_1(m), x_2(m), \cdots x_n(m)) \end{bmatrix} \tag{2-6}$$

四阶龙格-库塔法公式为：

$$\begin{bmatrix} x_1(m+1) \\ x_2(m+1) \\ \vdots \\ x_n(m+1) \end{bmatrix} = \begin{bmatrix} x_1(m) \\ x_2(m) \\ \vdots \\ x_n(m) \end{bmatrix} + \frac{1}{6}\left(\begin{bmatrix} k_{11} \\ k_{21} \\ \vdots \\ k_{n1} \end{bmatrix} + 2\begin{bmatrix} k_{12} \\ k_{22} \\ \vdots \\ k_{n2} \end{bmatrix} + 2\begin{bmatrix} k_{13} \\ k_{23} \\ \vdots \\ k_{n3} \end{bmatrix} + \begin{bmatrix} k_{14} \\ k_{24} \\ \vdots \\ k_{n4} \end{bmatrix} \right) \tag{2-7}$$

式中

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

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

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

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

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

4. 几个基本概念

（1） 非线性微分方程

非线性系统的特征是，系统不满足迭加原理，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。举例来说，非线性力学中常讨论的杜芬 (Duffing) 方程：

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

万达波尔方程：

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

都是非线性微分方程。或者在线性环节中插入了典型的非线性环节，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2-3是具有滞环的非线性系统。

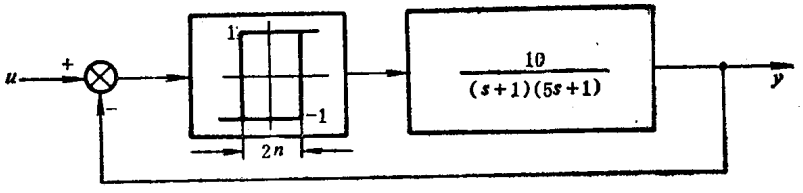

图2-3 具有滞环的非线性系统

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求解非线性方程。

(2) 刚性微分方程

刚性微分方程又称为病态方程。这类方程概念是，对于一般常微分方程：

$$\frac{dx_i}{dt}=f_i(t,x_1,x_2\cdots x_n)\qquad(i=1,2,\cdots n)$$

其物理特征通常由雅可比矩阵

$$(a_{ij})=\left(\frac{\partial f_i}{\partial x_j}\right)\text{的特征值}\lambda_i=\alpha_i+j\beta_i$$

来表达。特征值实部$\mathrm{Re}\lambda_i=\alpha_i$表示增长($\alpha_i>0$)或衰减($\alpha_i<0$)因子，虚部$I_m\lambda_i=\beta_i$表示周期振动频率。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$\rho\gg1$时，系统就称为刚性的。如某些控制系统、电子网络、精馏塔模型等系统中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决定积分步距。因此解题时间将拉得很长，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$\tau=10^{-6}$，$T=1$，则积分步数$N>\frac{1}{2.7}10^6$。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机时就更无法估量。目前求解刚性方程较好的方法是变阶变步长吉尔方法。我国数学工作者还提出了小参数法[9]。

(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。机理模型往往采取这种形式。如大规模集成电路的动态模型，其状态向量不仅含有相应于微分方程的变量，电容树枝电压和电感弦电流，也包括相应于代数方程的变量，电导树枝电压和电阻弦电流。又如精馏塔机理模型，在微分方程中常穿插有静态工艺计算的非线性代数方程。

混合方程可采取分别计算的方法处理。或用隐式积分方法直接处理。

（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：

$$x_{m+1}=x_m+h\dot{x}_m \tag{2-8}$$

隐式欧拉法有多种形式，如梯形积分公式：

$$x_{m+1}=x_m+\frac{h}{2}(\dot{x}_{m+1}+\dot{x}_m) \tag{2-9}$$

矩形积分公式：

$$x_{m+1}=x_m+h\dot{x}_{m+1} \tag{2-10}$$

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：$\dot{x}=f(t,x)$ (2-11)

可以导出差分方程的近似表达式：

$$x_m=a_1x_{m-1}+a_2x_{m-2}\cdots+h[b_0\dot{x}_m+b_1\dot{x}_{m-1}+b_2\dot{x}_{m-2}\cdots] \tag{2-12}$$

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算出。但是，如果$b_0\neq 0$，$\dot{x}_m$出现在公式右端，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：

$$\tau\dot{x}+x=u \tag{2-13}$$

式中，τ是常数，$x=x(t)$，$u=u(t)$外作用函数。设h为积分步长。由式(2-13)可变换得：

$$\dot{x}_m=\frac{1}{\tau}(u_m-x_m) \tag{2-14}$$

代入式(2-8)：

$$x_{m+1}=x_m+\frac{h}{\tau}(u_m-x_m) \tag{2-15}$$

合并同类项化简后得到显式解法通式：

$$x_{m+1}=\left(1-\frac{h}{\tau}\right)x_m+\frac{h}{\tau}u_m \tag{2-16}$$

再看隐式矩形公式的情况，将式(2-14)改写为：

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

代入式(2-10)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：

$$x_{m+1}=\left(\frac{1}{1+h/\tau}\right)x_m+\left(\frac{h/\tau}{1+h/\tau}\right)u_{m+1} \tag{2-17}$$

可以证明式(2-16)及(2-17)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。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：

$$\frac{dx}{dt}=1-x$$

初始条件$t=0$，$x=0$。精确解是：

$$x=1-e^{-t}$$

我们选择不同的时间步长，分别用欧拉法四阶龙格-库塔法求解，将结果描绘在图2-4及图 2-5 中。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、甚至很快造成溢出。欧拉法解法出现不稳定的临界步长 $(h)_{max}=2$。四阶龙格-库塔法解法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、输出量及输出量对时间的各阶导数也都定为 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，便于比较系统的瞬态响应。

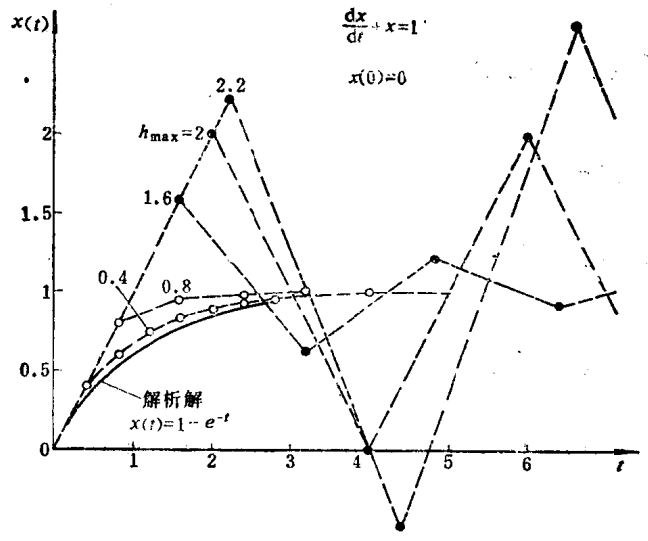

图2-4　欧拉法的数值稳定性

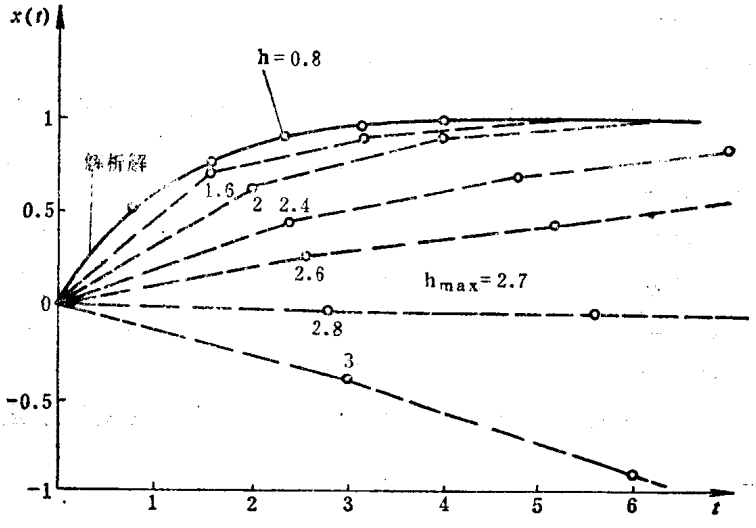

图2-5　四阶龙格-库塔法的数值稳定性

（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验。因为大部分问题在计算前对于系统特性、过渡时间、各时间常数确切值不一定了解。通常选用尽可能小的步长计算。但阶次较高的系统，为了节省机时应当探索确保稳定性及必要精度的尽可能大的步长。

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。提高了效率，减少了人工试探的盲目性。

5．几种常用数值积分解法及选择

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提供选择。几种常用数值积分解法及比较见表2-1。从表2-1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。精度高、稳定性好的方法计算量都较大，程序也较复杂。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。据报导[10]采用

表2-1 几种常用数值积分解法及比较

方法名称	优点	缺点	适用范围
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	同上	同上	同上
定步长四阶龙格-库塔法	程序较简单、精度较高．小步长可解刚性方程	高阶较费时	非刚性、不太光滑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	精度高	计算量大、高阶费时不能解刚性方程	高精度计算，[illegible]用
默森单步法	精度高	计算量大、高阶费时不能解刚性方程	同上
外插法	精度高	稳定性不高	同上
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

改进隐式吉尔方法，配合大型稀疏矩阵技术可以解决这类问题。

表2-1仅仅列出了几种常用方法，在文献[11]中大部分可以得到FORTRAN程序。此外，如龙格-库塔的各种改进方法，隐式单步法，改进默森法等等，在文献中均可查到这里就不一一列举了。

§2-3 连续系统数字仿真程序 I

本程序面向 n 阶传递函数(2-18)。能够自动将(2-18)式转换成状态方程，然后运用定步长四阶龙格-库塔法求出阶跃、斜坡、加速度、正弦干扰信号的瞬态响应值表及曲线。本程序配合第五章图论程序可以解决复杂SISO与MIMO网络的仿真。本程序还具有连分式降阶、求阶跃响应的各项质量指标功能。

一、计算方法

设 n 阶传递函数的通式为：

$$G(s)=\frac{B_0s^m+B_1s^{m-1}+\cdots+B_{m-1}s+B_m}{A_0s^n+A_1s^{n-1}+\cdots+A_{n-1}s+A_n}\cdot e^{-\tau s}\qquad (n\geqslant m)\qquad (2\text{-}18)$$

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(2-18)转换成微分方程的推导如下：

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

$$=\frac{y(s)}{u(s)} \qquad (n\geqslant m)$$

$$\therefore \quad (A_0s^n+A_1s^{n-1}+\cdots+A_{n-1}s+A_n)y(s)$$
$$=(B_0s^m+B_1s^{m-1}+\cdots+B_{m-1}s+B_m)e^{-\tau s}u(s)$$
$$\therefore \quad A_0y^{(n)}(t)+A_1y^{(n-1)}(t)+\cdots+A_{n-1}y^{\cdot}(t)+A_ny(t)$$
$$=B_0u^{(m)}(t-\tau)+B_1u^{(m-1)}(t-\tau)+\cdots+B_{m-1}u^{\cdot}(t-\tau)+B_mu(t-\tau) \qquad (2\text{-}19)$$

为了应用龙格-库塔法求解n阶微分方程(2-19)，必须对式(2-19)作适当形式上的改变以适应于龙格-库塔法的数字程序。因此，首先考察四阶龙格-库塔法的求解步骤：

对于已知初始条件的一阶微分方程组

$$\frac{dy_i}{dt}=f_i(t,y_1,y_2,\cdots y_n)$$

$$y_i(t_0)=y_{i0} \qquad (i=1,2,\cdots n) \qquad (2\text{-}20)$$

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}之四阶龙格-库塔法公式如下：

$$k_{i1}=f_i(t_k,y_{1k},\cdots,y_{nk}),$$
$$k_{i2}=f_i(t_k+h/2,\ y_{1k}+hk_{11}/2,\cdots,y_{nk}+hk_{n1}/2),$$
$$k_{i3}=f_i(t_k+h/2,\ y_{1k}+hk_{12}/2,\cdots,y_{nk}+hk_{n2}/2),$$
$$k_{i4}=f_i(t_k+h,\ y_{1k}+hk_{13},\cdots,y_{nk}+hk_{n3}),$$
$$y_{ik+1}=y_{ik}+h/6(k_{i1}+2k_{i2}+2k_{i3}+k_{i4}) \qquad (i=1,2,\cdots,n) \qquad (2\text{-}21)$$

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，重复一次以上递推公式。同时以前一次求出的$y_{ik}(i=1,2,\cdots n)$作为求下一时刻y_{ik+1}的初始值，这样一直计算到要求的终止时间为止。这就是全部解题步骤。

以上步骤看出，为了应用龙格-库塔法求解n阶微分方程，必须设法把其变换成n元一阶微分方程组。应用状态方程表示法，可以将式(2-19)表示的n阶微分方程化为n元一阶微分方程组，具体方法是：

首先将式(2-19)规格化。即将式(2-19)的左端及右端同除以A_0使得$y^{(n)}$系数成为1。式(2-19)变成（当$n=m$时）：

$$y^{(n)}(t)+a_1y^{(n-1)}(t)+\cdots+a_{n-1}\dot{y}(t)+a_ny(t)$$
$$=b_0u^{(n)}(t-\tau)+b_1u^{(n-1)}(t-\tau)+\cdots+b_{n-1}\dot{u}(t-\tau)+b_nu(t-\tau) \qquad (2\text{-}22)$$

式中

$$a_i=A_i/A_0,\ b_i=B_i/A_0$$

然后，引入状态变量$x_1(t)$，$x_2(t)$，$\cdots x_n(t)$，当$n=m$时，

$$\begin{cases} x_1(t)=y(t)-\beta_0 u(t-\tau) \\ x_2(t)=\dot{y}(t)-\beta_0\dot{u}(t-\tau)-\beta_1 u(t-\tau) \\ \quad\quad =\dot{x}_1(t)-\beta_1 u(t-\tau) \\ x_3(t)=\ddot{y}(t)-\beta_0\ddot{u}(t-\tau)-\beta_1\dot{u}(t-\tau)-\beta_2 u(t-\tau) \\ \quad\quad =\dot{x}_2(t)-\beta_2 u(t-\tau) \\ \cdots\cdots \\ x_n(t)=y^{(n-1)}(t)-\beta_0 u^{(n-1)}(t-\tau)-\beta_1 u^{(n-2)}(t-\tau)-\cdots \\ \quad\quad -\beta_{n-2}\dot{u}(t-\tau)-\beta_{n-1}u(t-\tau) \\ \quad\quad =\dot{x}_{n-1}(t)-\beta_{n-1}u(t-\tau) \end{cases} \tag{2-23}$$

式中β_i由如下递推公式求出：

$$\begin{cases} \beta_0=b_0 \\ \beta_1=b_1-a_1\beta_0 \\ \beta_2=b_2-a_1\beta_1-a_2\beta_0 \\ \cdots\cdots \\ \beta_n=b_n-a_1\beta_{n-1}-\cdots-a_{n-1}\beta_1-a_n\beta_0 \end{cases} \tag{2-24}$$

得到的n元一阶微分方程组为：

$$\begin{cases} \dot{x}_1(t)=x_2(t)+\beta_1 u(t-\tau) \\ \dot{x}_2(t)=x_3(t)+\beta_2 u(t-\tau) \\ \dot{x}_3(t)=x_4(t)+\beta_3 u(t-\tau) \\ \cdots\cdots \\ \dot{x}_n(t)=-a_n x_1(t)-a_{n-1}x_2(t)-a_{n-2}x_3(t)- \\ \quad\quad \cdots-a_2 x_{n-1}(t)-a_1 x_n(t)+\beta_n u(t-\tau) \end{cases} \tag{2-25}$$

所求的输出变量$y(t)$可以从式(2-23)导出：

$$y(t)=x_1(t)+\beta_0 u(t-\tau) \tag{2-26}$$

一般情况下输入端阶次是低于输出端的，设为m阶，若$m<n$，则以上式中只要令$b_0=b_1=\cdots=b_{n-m+1}=0$即可，于是式(2-23)，(2-25)中的$\beta_0=\beta_1=\beta_2=\cdots=\beta_{n-m+1}=0$且式(2-26)成为：

$$y(t)=x_1(t) \tag{2-27}$$

式(2-25)常常表达成状态方程：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots\cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots\cdots & 0 \\ \cdots\cdots & & & & & \\ 0 & \cdots\cdots & & & 0 & 1 \\ -a_n & -a_{n-1} & \cdots & -a_2 & & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \vdots \\ \beta_n \end{bmatrix} u(t-\tau) \tag{2-28}$$

即：$\dot{\mathbf{x}}=\mathbf{A}\mathbf{x}+\mathbf{B}u(t-\tau)$ (2-29)

以上状态方程表达式用龙格-库塔法求解就非常方便了。

二、程序概述

连续系统仿真程序Ⅰ结构比较简单，是专门配合图论程序而编制的。可以接受n阶规格