

实时仿真算法研究¹⁾

谢道奎

(北京自动化工程研究所)

摘 要

本文针对实时仿真的特点,提出了处理 STIFF 系统的分块混合算法。这个算法可以加大计算步长、减少实时计算工作量、降低半实物仿真对数字机速度的要求。

在控制系统半实物数字仿真中,数字机是闭环系统的一个组成部分,用来模拟控制系统中的某些环节。例如在实验室难以实现的控制对象、尚未生产出来的某些控制部件等。处理这些环节的时间要求是十分严格的,只有数字机的计算速度 $V_d \geq n_n/h$ (n_n 为步计算工作量, h 是满足精度要求的计算步长),步计算执行时间小于步长,才能做到实时。计算方法十分重要,它对计算量和步长两者都有影响。

一、问 题

一个复杂的控制系统通常由许多功能和特性各异的子系统组成。例如,一个飞行器控制系统可以分成控制对象(飞行器机体动力学模型)和控制器两个子系统。就控制对象而言,飞行器机体动力学模型又可分为质心和绕心运动模型。其中绕心运动模型又是刚体运动、燃料晃动、弹性振动的合成,……。控制器也是这样,可分为姿态控制器和航迹控制器,各自又包括测量和目标搜索、信息处理及综合判定、执行机构等部分。广义地讲,上述所有这些部分,包括自然模型块和人工分割的模型块,都可看成子系统。它们的微分方程可以写成下列一般形式:

$$X_i = \sum_{j=1}^n A_{ij} X_j + \sum_{k=1}^m B_{ik} U_k + C_i U_i, \quad i=1, \dots, n.$$

其中 n 表示子系统模块数, X, \dot{X} 为状态变量行矩阵, U 为外干扰行矩阵, A, B, C 为系数矩阵。在许多情况下,这组方程不仅是隐含的,而且是 STIFF 的。图 1 是问题的图解形式。

本文于1984年2月27日收到。

1) 本文曾在1983年第四届全国系统仿真学术会议上宣读。

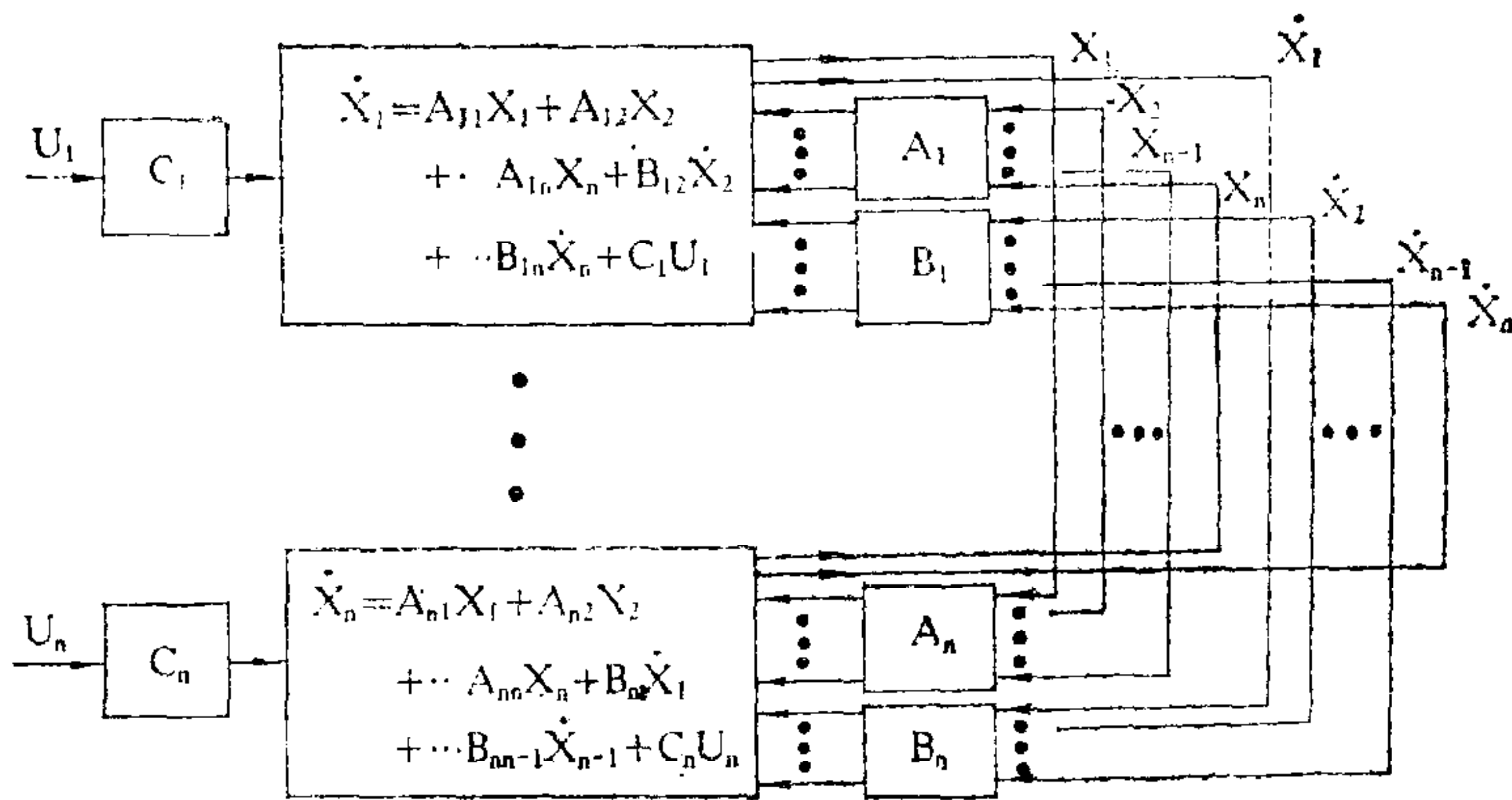


图 1

二、算 法

仿真中经常遇到STIFF微分方程,其特点是存在相差很大的时间常数。在物理系统中,表现为不同部分有相差很大的反应速度。计算方法的稳定性将主要由快变部分决定。就大多数算法而言,稳定条件是 $|\lambda h| < K$ (常数)。不同的算法, K 值不同, $K_{\text{Euler}} = 2$; $K_{\text{AB4}} = 0.3$; $K_{\text{RK4}} = 2.785$; ……。

因此,如果把题目看成一个整块,用同一种方法计算,为保证稳定性就必须按最小时间常数 $\tau = (-\frac{1}{\lambda_{\text{max}}})$ 选步长。这个步长对慢变部分常常是不必要的。

如果按前节提出的“广义模型块”概念,根据系统结构和动特性,把一个大的系统分成若干个模块子系统,进而根据这些模块的特性和相互关系,安排算法、步长和计算顺序,显然是很有好处的。最简单的情况是把系统分成低速 (X_1) 和高速 (X_2) 两个模块子系统。系统方程为

$$\begin{aligned} \dot{X}_1 &= A_{11}X_1 + A_{12}X_2 + B_{12}\dot{X}_2 + C_1U_1, \\ \dot{X}_2 &= A_{21}X_1 + A_{22}X_2 + B_{21}\dot{X}_1 + C_2U_2. \end{aligned}$$

1. 数值积分计算

在一个系统中一般慢变部分是基本的,计算量较大,要求精度较高;快变部分小些,计算量较小,精度要求也稍低。两部分的工作频率有时可相差一个数量级,对此分块混合算法特别有效。此时,低频部分用 AB4 这样的精度高、计算量较小但稳定性较差的算法计算,而高频部分采用稳定性好的隐式算法,可达到在整体上减少计算工作量、增大计算步长、缩短计算执行时间的目的。

本文在一个实例上试验了表 1 所示的算法组合。

2. 耦合的处理

模块化的系统方程为

$$\dot{X}_i = \sum_{j=1}^n A_{ij}X_j + \sum_{j=1}^n B_{ij}\dot{X}_j + C_iU_i, \quad i = 1, \dots, n.$$

表 1

	低频部分	高频部分	说 明
1	ELC	ELB	计算高频部分时, 低频交连项使用预测值, 相同步长
2	ELC	ELA	(同上)
3	ELC转AB4	ELA	(同上)
4	(同上)	ELA转ST2	(同上)
5	(同上)	ELA转ST3	(同上)
6	(同上)	ELA转RK4	(同上)
7	(同上)	ELA转2RK4	低频部分计算步长为高频部分的两倍
8	(同上)	ELA转2ELA	(同上)

在模块间存在隐含关系, 即 $B_{ij} \neq 0$, 或虽然 $B_{ij} = 0$ 但采用隐式算法的两种情况下, 严格的计算必须在模块间进行迭代. 只有在模块间不存在隐含关系, 并采用计算出发值为前一步 (或前几步) 计算结果的算法时, 各模块的计算才可以独立进行. 因此, 解决模块间的交连问题, 是分块混合算法成功的基础.

在以往的计算中, 凡是在方程右端出现的导数变量交连项都是用前一步的值代替作近似计算, 通过选择更小的计算步长来弥补这种近似计算所造成的计算精度和稳定性损失. 在使用分块混合算法时, 为了保证各个模块能独立进行计算, 模块间的交连通过插值预报和合理安排计算次序来解决. 这比利用前一步值代替要精确得多.

3. 外插计算

模块化后的系统方程亦可改写成

$$\dot{X}_i = A_{ii}X_i + \sum_{j \neq i}^n (A_{ij}X_j + B_{ij}\dot{X}_j) + C_i U_i,$$

其中第二项即为其它模块对 i 模块的交连项.

$$\text{令 } GF_i = \sum_{j \neq i}^n (A_{ij}X_j + B_{ij}\dot{X}_j),$$

在计算 i 模块时, 先计算 GF_i , 然后按选定的积分公式执行该模块的计算. 如果积分公式仅需 $GF(nT)$, 可通过上述公式得到; 如果需要 nT 以后的值, 则需进行外插.

$$\text{令 } \widehat{GF}_n = GF(nT + t_p), \quad t_p = \alpha T, \quad 0 < \alpha < 1.$$

常用的外插公式有:

1) 线性外插

$$\text{① } \widehat{GF}_n = GF_n + \alpha (GF_n - GF_{n-1}).$$

当 $\omega T \ll 1$ 时, 外插器的增益和相位误差为

$$\epsilon_m \cong -\frac{\alpha}{2} (1 + \alpha) (\omega T)^2,$$

$$\epsilon_A \cong -\frac{\alpha}{6} (1 + 3\alpha + 2\alpha^2) (\omega T)^3.$$

$$\begin{aligned} \textcircled{2} \quad \widehat{GF}_n &= GF_n + \alpha T \dot{GF}_n \\ \dot{GF}_n &= \Sigma (A \alpha T \dot{X}_n + B \alpha T \ddot{X}_n) . \end{aligned}$$

同样, 在 $\omega T \ll 1$ 时有

$$\begin{aligned} \epsilon_M &\cong -\frac{\alpha^2}{2} (\omega T)^2, \\ \epsilon_A &\cong -\frac{\alpha^3}{3} (\omega T)^3. \end{aligned}$$

2) 抛物线外插

$$\widehat{GF}_n = \left(1 + \frac{3}{2} \alpha + \frac{1}{2} \alpha^2\right) GF_n - (2\alpha + \alpha^2) GF_{n-1} + \left(\frac{\alpha}{2} + \frac{\alpha^2}{2}\right) GF_{n-2}.$$

在 $\omega T \ll 1$ 时有

$$\begin{aligned} \epsilon_M &\cong \frac{\alpha}{8} (2 + 5\alpha + 4\alpha^2 + \alpha^3) (\omega T)^4, \\ \epsilon_A &\cong \frac{\alpha}{6} (2 + 3\alpha + \alpha^2) (\omega T)^3. \end{aligned}$$

三、计算公式说明

1. 简化向后EULER法

向后EULER公式为

$$y_{n+1} = y_n + hf(y_{n+1}, t_{n+1}),$$

这是一个隐式方法, 稳定性很好. 当步长足够小时 (与变量变化率比较), 精度也可以满足要求. 下面介绍两种简化计算方法:

1) 转一次右函数的向后EULER法 (简称ELC).

- ① 利用 n 步的值预报 \bar{y}_{n+1} , $\bar{y}_{n+1} = y_n + hf(y_n, t_n)$;
- ② 计算 $f(\bar{y}_{n+1}, t_{n+1})$;
- ③ 利用向后EULER公式求 y_{n+1} , $y_{n+1} = y_n + hf(y_{n+1}, t_{n+1})$;
- ④ 认为 $f(\bar{y}_{n+1}, t_{n+1}) \doteq f(y_{n+1}, t_{n+1})$, 不再计算右函数 $f(y_{n+1}, t_{n+1})$.

每步只计算一次右函数 $f(\bar{y}_{n+1}, t_{n+1})$.

2) 转两次右函数的向后EULER法 (简称ELB).

除第④步为计算右函数 $f(y_{n+1}, t_{n+1})$ 外, 其它与ELC相同.

本文的试验例子中, 这两种方法的稳定性, 尤其是ELB法, 都较显式EULER法为好.

讨论方程 $\dot{Y} = \lambda Y + U$. 将ELC用于该方程, 可导出其差分方程为

$$Y_n = (1 + 2\lambda T)Y_{n-1} - T\lambda Y_{n-2} + TU.$$

作 Z 变换得

$$Y(z) = (1 + 2\lambda T)z^{-1}Y(z) - \lambda Tz^{-2}Y(z) + TU(z).$$

Z 传递函数为

$$H_c(z) = \frac{Y(z)}{U(z)} = \frac{T}{1 - (1 + 2\lambda T)z^{-1} + \lambda Tz^{-2}}.$$

特征方程是

$$z^2 - (1 + 2\lambda T)z + \lambda T = 0,$$

$$\lambda T = \frac{z^2 - z}{2z - 1} \quad \left(z \neq \frac{1}{2}\right).$$

令 $z = e^{j\theta}$ 可得

$$\lambda T = \frac{(\cos 2\theta - \cos \theta) + j(\sin 2\theta - \sin \theta)}{(2\cos \theta - 1) + j2\sin \theta}.$$

当 θ 从 0 变至 2π 时, λT 的轨迹 r 所围成的区域 Ω 即是 ELC 算法的稳定域 (见图 2)。

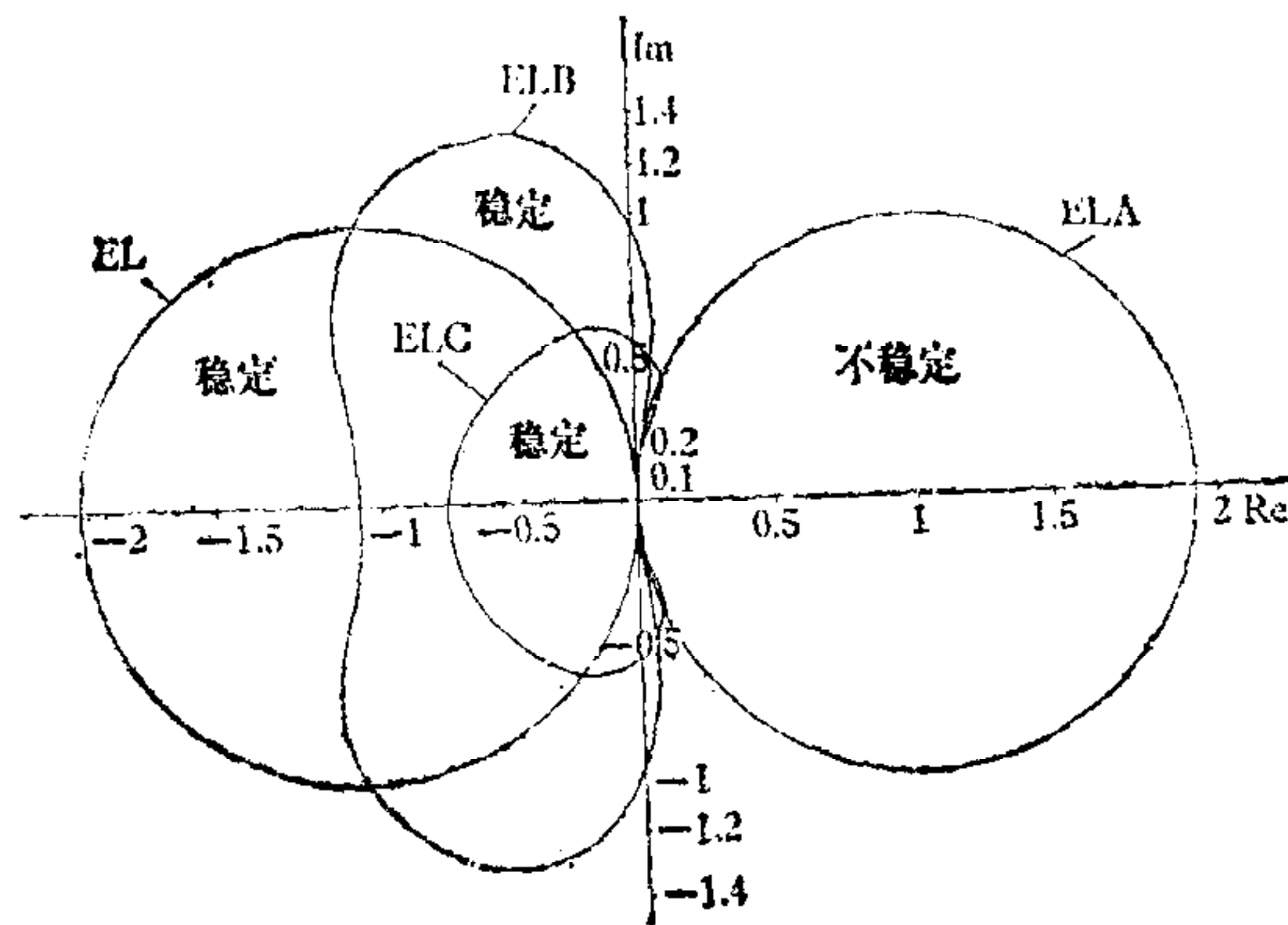


图 2

同样, 对 ELB 有

$$H_B(z) = \frac{T}{1 - (1 + T\lambda + (T\lambda)^2)z^{-1}}.$$

特征方程是

$$z - (1 + T\lambda + (T\lambda)^2) = 0.$$

相应的稳定域见图 2。

显式 EULER 法的 Z 传递函数为

$$H(z) = \frac{T}{1 - (1 + T\lambda)z^{-1}},$$

其稳定域由 $z - (1 + T\lambda) = 0$ 确定。

从图 2 可见, ELB, ELC 的稳定域在实轴方向被压缩, 虚轴方向被拉长。对于特征根分布在虚轴附近 (即根的虚部较大, 实部较小) 的题目, 有比显式 EULER 法更好的稳定性。

通过 Z 传递函数的比较 (计算 $\frac{H_c - H}{H}$, $\frac{H_B - H}{H}$), 不难证明 ELC, ELB 在计算精度方面并无改善。

2. EULER 混合法

分别用ELC和ELB计算系统的慢变部分(X_1)和快变部分(X_2)。由于 $(\tau_{x1})_{min}/(\tau_{x2})_{min} \gg K_{ELB}/K_{ELC}$, 所以即使取同样的步长也可加速计算过程。因为: ①ELC比ELB计算量小, 而且在用ELB计算快变部分时, 右函数中的低频交连项在一步中的变化甚微, 可只计算一次。所以ELC和ELB混合使用, 计算量远小于单一的ELB。②ELC的计算稳定性比ELB差, 单一ELC算法需取较混合算法小得多的步长。

3. AB4+ELA混合法

ELA是为高频部分的计算而设计的向后EULER法。对于

$$\dot{X}_2 = A_{21}X_1 + A_{22}X_2 + B_{21}\dot{X}_1 + C_2U_2,$$

令 $GF = A_{21}X_1 + B_{21}\dot{X}_1 + C_2U_2,$

得 $\dot{X}_2 = A_{22}X_2 + GF.$

将向后EULER公式用于上述方程组得

$$X_2^{n+1} = X_2^n + h(A_{22}^{n+1}X_2^{n+1} + GF^{n+1}),$$

$$(I - hA_{22}^{n+1})X_2^{n+1} = X_2^n + hGF^{n+1},$$

$$\therefore X_2^{n+1} = (I - hA_{22}^{n+1})^{-1}(X_2^n + hGF^{n+1}).$$

其中 GF^{n+1} 利用插值方法计算。

ELA的稳定域如图2所示。在本文的试验例子中, ELA与ELC混合使用, 稳定计算步长较ELC+ELB提高4倍以上。这说明系统低频部分的计算稳定性可以保证, 精度是主要矛盾。理所当然地应该探讨使用精度高的计算方法。从精度和计算量考虑, AB4显式多步法是比较好的。计算公式为

$$Y_{n+1} = Y_n + h \sum_{i=1}^k \alpha_i f_{n-i+1}.$$

当 $K=4$ 时,

$$\alpha_1 = \frac{55}{24}, \quad \alpha_2 = -\frac{59}{24}, \quad \alpha_3 = \frac{37}{24}, \quad \alpha_4 = -\frac{9}{24},$$

即

$$Y_{n+1} = Y_n + \frac{h}{24}(55\dot{Y}_n - 59\dot{Y}_{n-1} + 37\dot{Y}_{n-2} - 9\dot{Y}_{n-3}).$$

试验证明AB4+ELA混合法与ELC+ELA有相同的计算稳定性。

4. AB4+STIFF混合法

STIFF方法被认为是一种稳定性较好的、适合于求解STIFF方程的隐式多步方法。计算公式为

$$Y_n = \sum_{i=1}^k \alpha_i Y_{n-i} + h\beta_0 f_n.$$

当 $K=2$ 时,

$$\beta_0 = \frac{2}{3}, \quad \alpha_1 = \frac{4}{3}, \quad \alpha_2 = -\frac{1}{3},$$

当 $K=3$ 时,

$$\beta_0 = \frac{6}{11}, \quad \alpha_1 = \frac{18}{11}, \quad \alpha_2 = -\frac{9}{11}, \quad \alpha_3 = \frac{2}{11}.$$

1) AB4+ST2混合法

二阶STIFF计算公式为

$$Y_n = \frac{4}{3} Y_{n-1} - \frac{1}{3} Y_{n-2} + \frac{2}{3} h \dot{Y}_n.$$

代入 $\dot{X}_2 = A_{22} X_2 + GF$ 得

$$X_2^n = \frac{4}{3} X_2^{n-1} - \frac{1}{3} X_2^{n-2} + \frac{2}{3} h (A_{22} X_2^n + GF^n).$$

$$X_2^n = \left(I - \frac{2}{3} h A_{22} \right)^{-1} \left(\frac{4}{3} X_2^{n-1} - \frac{1}{3} X_2^{n-2} + \frac{2}{3} h GF^n \right).$$

其中 GF^n 同样由插值计算产生。

2) AB4+ST3混合法

三阶STIFF计算公式为

$$Y_n = \frac{18}{11} Y_{n-1} - \frac{9}{11} Y_{n-2} + \frac{2}{11} Y_{n-3} + \frac{6}{11} h \dot{Y}_n$$

代入 $\dot{X}_2 = A_{22} X_2 + GF$ 得

$$X_2^n = \left(I - \frac{6}{11} h A_{22} \right)^{-1} \left(\frac{18}{11} X_2^{n-1} - \frac{9}{11} X_2^{n-2} + \frac{2}{11} X_2^{n-3} + \frac{6}{11} h GF^n \right).$$

5. AB4+RK4混合法

AB4及RK4是两种用得最多、精度较高的数值积分方法。这两者的结合对于某些要求更高精度的计算任务是最理想的。

RK4公式
$$Y_{n+1} = Y_n + \sum_{i=1}^k W_i K_{n,i}.$$

当 $K=4$ 时, RK4的经典形式为

$$Y_{n+1} = Y_n + \frac{h}{6} (K_{n,1} + 2K_{n,2} + 2K_{n,3} + K_{n,4}).$$

$$K_{n,1} = f(t_n, Y_n),$$

$$K_{n,2} = f\left(t_n + \frac{1}{2}h, Y_n + \frac{1}{2}h K_{n,1}\right),$$

$$K_{n,3} = f\left(t_n + \frac{1}{2}h, Y_n + \frac{1}{2}h K_{n,2}\right),$$

$$K_{n,4} = f(t_n + h, Y_n + h K_{n,3}).$$

共计算四次右函数,但只在计算 $K_{n,2}$, $K_{n,4}$ 时有关变系数和外输入需插值。

6. 不同步长混合算法

相同步长混合算法的最大步长受到高频部分的限制,特别是当高频部分采用高阶方法,

如ST3, RK4的时候,可取的步长常常远未达到低频部分允许的极限。

不同步长混合算法可以在保证精度的条件下,各自取最大的计算步长,从而达到最短的全程计算时间。

1) AB4+2RK4混合法

由所选试例知:RK4计算高频部分的最大稳定步长为8ms,AB4计算低频部分的最大稳定计算步长超过32ms,相差4倍以上。说明在极限情况下可以按1:4安排两部分的计算步长。在本文进行的试验中,为了编程序方便,按1:2取步长,即低频部分算一步(H),高频部分算两步(h), $H=2h$,故定名为AB4+2RK4混合法。

2RK4程序与RK4的区别仅在于变系数及外输入(GF)的插值安排。

试验中AB4+2RK4混合法的最大稳定计算步长 $H=9ms$ ($h=4.5ms$),远未达到预期的效果。可能因为RK4需4次扫描,低频交连项看成外部输入后,无法反映后几次扫描对低频量的影响,损失了多次扫描法的精度。

2) AB4+2ELA混合法

在本文的例子中,AB4+ELA混合法的稳定计算步长已大于32ms,超过了香农规定的信号恢复极限。试验AB4+2ELA混合法($H=2h$)的目的是为了增加高频部分理计算点($h=16ms$),提高信号恢复精度。这在半实物数字仿真中是很有意义的。

四、结束语

(1) 本文试验的算法和在实例上执行的结果如表2所示。可见,分块混合算法可以有效地减少计算工作量、缩短计算执行时间,是单一算法不可比拟的。

表2

	算 法	最大允许步长	在DJS-8机上的计算执行时间(相对值)
1	ELC	4ms	7.07
2	ELB	8ms	5.83
3	ELC+ELB	8ms	4.01
4	ELC+ELA	32ms	1.0
5	AB4+ELA	32ms	1.0
6	AB4+ST2	32ms	1.0
7	AB4+ST3	3ms	12.04
8	AB4+RK4	8ms	5.71
9	AB4+2RK4	9ms	6.54
10	AB4+2ELA	32ms	1.17

(2) 在半实物数字仿真中,为了提高D/A转换(包括零阶保持器在内)的精度,总是希望尽可能减小计算步长。因此,简单的混合算法,诸如ELC+ELA,AB4+ELA,AB4+ST2,AB4+2ELA等,有最大的吸引力。

(3) 好的算法与输出校正(D/A及零阶保持器延迟补偿)相结合可以有效地降低半实物

数字仿真对数字机速度的要求。

参 考 文 献

- 〔1〕 谢道奎, 数字机用于实时仿真的几个问题, 计算机工程与设计, 2(1984), 41.
- 〔2〕 C.W. 吉尔, 常微分方程的数值解法, 费景高等译, 科学出版社, (1978年)。
- 〔3〕 祝楚恒、袁兆鼎, 常微分方程数值积分的计算稳定性, 计算数学, 1(1980), 77.

A STUDY ON REAL TIME SIMULATION ALGORITHM

Xie Daokui

(Beijing Automatic Control Engineering Institute)

Abstract

Based on the features of real time simulation, a hybrid algorithm is proposed in this paper. Different algorithms are used for different parts of the system. Such algorithm can increase the computing step size, reduce real time computation and decrease the speed requirement for digital computer in on-line digital simulation.