第5卷 第4期 自 动 化 学 报 Vol. 5, No. 4
1979 年 10 月 ACTA AUTOMATICA SINICA Oct., 1979

连续系统的数字仿真方法*

刘志俊 李宝绶
(北京控制工程研究所)

摘 要

近年来，已逐渐用数字计算机代替模拟计算机或混合计算机来实现连续系统的仿真。连续系统的数字仿真的显著特点是：便于人机对话，求解精度高和重复性好，容易编排程序和输出计算结果。但是，为了避开编程序的细节和把精力集中在系统仿真上，系统仿真者希望有一个方便和灵活的仿真语言。本文介绍一种面向方程的 BASIC 仿真语言，它允许用户直接用一阶微分方程组的数学表达式写入，并能在任何一台配置 BASIC 语言的数字机上运算。使用时只要遵循规定的简单格式而不用去熟悉 BASIC 语言的细节。文中附有阐述用法的计算实例。

一、引 言

在工程系统的设计过程中，往往要做连续系统的动态仿真来选择系统的最佳设计方案和最佳参数。五十年代初，模拟计算机曾经是系统动态仿真的最重要的计算工具。那时，由于数字计算机的速度慢，尤其是程序设计很烦杂，不易为工程设计人员掌握，因此应用得很少。到六十年代初，开始应用模拟计算机和数字计算机联合起来工作的混合计算机实现连续系统的动态仿真。尽管混合机综合了数字机巨大的逻辑功能和存储能力以及模拟机的运算速度，但由于设备庞大，价格昂贵，程序复杂和调试时间冗长，仍然不能推广使用。近年来由于数字机硬件的突破，大大提高了运算速度，降低了价格，再加上软件上的改进，现在已开始单独用数字机实现连续系统的动态仿真[1,2]。它的优点是求解精度较高，重复性好，自动化程度高以及容易为工程技术人员掌握。

使用数字计算机实现连续系统仿真的关键是数字机要为用户配置一个方便而又灵活的仿真语言。国外较标准的仿真语言有 CSSL 及 CSMP，以及在 IBM 1300/1800 系统和 IBM 系统 3 和 7 上用的 DSL 语言，在 PDP11 上实现的 DAREP 等[2]。这是因为工程设计人员的主要兴趣是如何用计算机研究他们的系统，以便用最低的代价和最快的速度来选择系统的最佳结构和参数，而对计算机的程序设计并不一定十分熟悉和感兴趣。因此，工程设计人员希望系统仿真的程序设计和操作越方便越好。

当前国内普遍使用的小型数字机 DJS-130 一般都配置了 BASIC 会话语言[3]，但对于不熟悉 BASIC 语言的工程技术人员来说，要编系统仿真序程还有一定的困难，要占去不少的时间和精力。为了便于工程设计人员使用数字机实现连续系统的动态仿真，我们用 BASIC 语言编制了一种面向方程的连续系统的仿真程序，使用时只要把描述系统的动态

* 文本于 1978 年 10 月收到

方程、初始条件和要选择的参数按本文规定的格式输入到计算机，就能方便地得到计算结果．从而大大节省了用数字机实现连续系统动态仿真时编制程序的时间和精力，进一步扩大了数字机的应用范围，使它成为工程技术人员的有力工具．

这种仿真程序的格式也可以用 FORTRAN 语言或其他程序设计语言编制．一个好的程序应该便于使用，又便于维护，即便于修改和扩充．因此，编制本仿真程序的基本思想是把整个程序分成若干个程序块(即子程序)，每个程序块都完成一件事情，用户可以根据自已系统的特点修改和扩充，甚至用自已设计的新程序块来代替它，只要在调用它们的地方换为新的程序入口，对整个程序结构毫无影响．

文中附有使用本仿真程序计算的实例，具体地阐述了它的用法．

二、BASIC 仿真程序的结构

任何一个动力学系统都可以用状态方程

$$\frac{d\mathbf{Y}}{dT}=\mathbf{G}(\mathbf{Y},\mathbf{P},\mathbf{U},T) \tag{1}$$

$$\mathbf{Y}(T=0)=\mathbf{Y}(0)$$

描术．其中，$\mathbf{Y}=[Y1, Y2, \cdots, YN]$ 是 N 维状态向量，$\mathbf{P}=[P1, P2, \cdots, PK]$ 是 K 维参数向量，$\mathbf{U}=[U1, U2, \cdots, UM]$ 是 M 维控制向量，$\mathbf{G}=[G1, G2, \cdots, GN]$ 是 N 维函数向量，$\mathbf{Y}(0)=[Y1(0), Y2(0), \cdots, YN(0)]$ 是给定的初始状态．系统动态仿真的任务是通过计算求解方程(1)，选择满足给定性能指标的系统最佳参数 **P** 或最佳控制 **U**．

BASIC 仿真程序的结构如图 1 所示．整个程序分成八个程序块，主程序的功能是调用和管理各个子程序，它的控制流程如图 2 所示．其中：

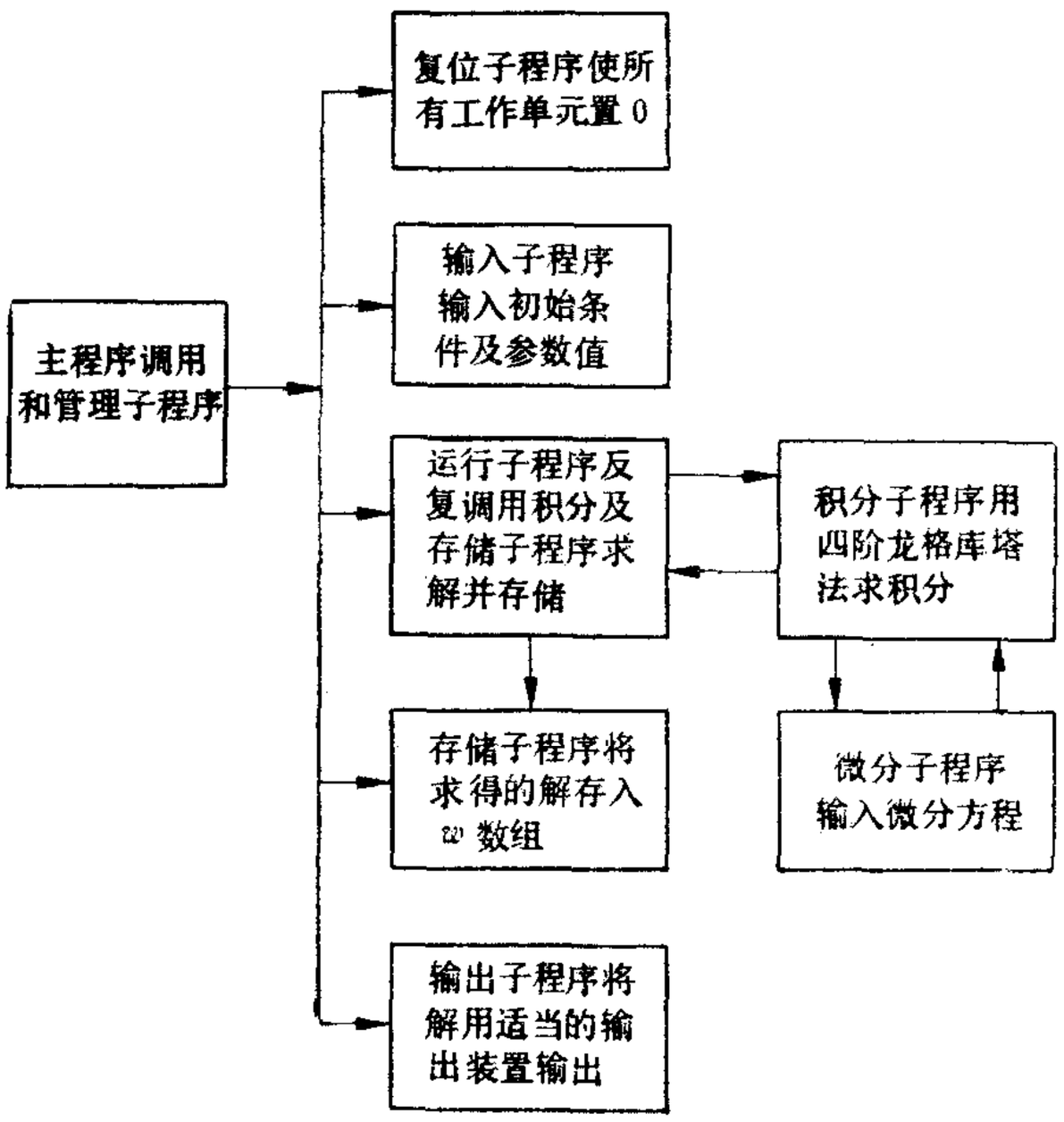

图 1　程序模块结构图

1）复位子程序：

作好计算前的准备工作，包括：

(1) 用键盘会话方式输入计算时间 $T1$，积分步长 $T2$，方程阶次 $N1$，方程中参数的个数 $N2$．要输出的解的点数 $N3$（在 $0 \leqslant T \leqslant T1$ 区间内，和外界交换一次信息的时间区间为 $T4$，$N3$ 为所分的区间数加 1）．

(2) 用 DATA 语句输入初始值 $\boldsymbol{Y}(0)$ 和参数 $\boldsymbol{P}$．

(3) 使其它有关工作单元置 0．

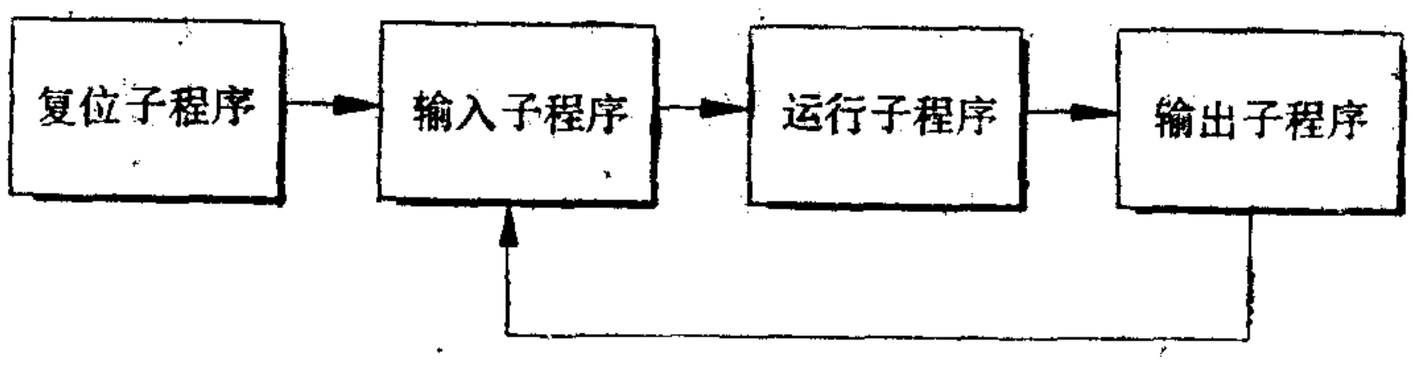

图 2 控制流程图

2)输入子程序:

用会话方式输入逻辑控制单元 $L4$

$L4 = 0$ 交互性计算,计算机提问改变那些初始值和参数,由键盘输入要改变的初值和参数.

$L4 = 1$ 成批处理多次运行,计算机提问运行处理多少组参数 $J3$,这些参数由 DATA 语句输入,自动运算完毕后,停机.

3)运行子程序:

首先检查一下是否

$$T2 > T4(T_4 = T1/(N3 - 1)),$$

若 $T2 > T4$,则置 $T2 = T4$,否则以原输入的 $T2$ 值作为步长计算. 它反复调用积分子程序产生解,并调用存贮子程序,把求得的解逐点存贮到数组 W 中去.

4)积分子程序:

是一个四阶龙格—库塔法 (Runge-Kutta) 积分子程序,通过调用微分子程序求得积分. 这里只列出了一种常用的积分方法子程序,事实上用户可以根据具体情况书写出自己所需的积分子程序,这时只要改变子程序的入口,其余内容均无需改动.

5)微分子程序:

我们规定

Y(i)	$i = 1$—20	为状态变量
	$i = 21$—30	为定义变量
$G(i)$	$i = 1$—20	为状态变量的导数
$P(i)$	$i = 1$—20	为系统的参数

用这些变量写出描述系统的微分方程. 因为 BASIC 语言规定变量的名称只能是一个字母,或后面最多再跟一个数字,且仅能使用不多于 94 个名字,所以我们采用数组元素做为变量,而且假定变量不超过 30 个(前 20 个是状态变量,后 10 个是定义变量),系统的参数不超过 20 个(这对于很广泛一类系统已足够了,否则只要改变 DIM 语句就能很容易扩大变量的数目).

此外,在时变系统中,还要用到自变量 T. 可以把 T 做为一个状态变量,增加一个微分方程,即认为 Y(1) = T,于是

$$G(1) = 1$$

用上述变量把描述系统的微分方程写成一阶微分方程组的形式,即等号左边只出现状态变量的一阶导数 $G(i)$,而等号右边是状态变量、定义变量及时间 T 的代数表达式. 引进定义变量的目的是,一方面我们需要一些中间结果,另一方面可以使微分方程写得简

洁，避免重复计算同样的代数表达式，节省计算时间．但要注意避免代数回路（这一点总是可以做到的）．书写时的次序为定义变量在先，以 $G(i)$ 书写的微分方程在后．

6)存贮数组子程序：

我们限定一次最多输出四个变量，用户只要用赋值语句把要记录的变量赋值给 $A(1)$、$A(2)$、$A(3)$ 和 $A(4)$，即 Let $A(i) = Y(j)$，则积分子程序自动把要输出的结果存入 W 数组中，准备输出．

7)输出子程序：

（1）自动计算出所有变量的最大值和最小值，以供模拟计算机设置比例尺时使用．

（2）成批处理时，打印每次运行时的参数值．

（3）打印出要输出的变量对时间的表格．

（4）打印出要输出的变量对时间的曲线．

三、用户书写数据及方程的格式

BASIC 语言的特点是每个语句都有标号，按语句标号的大小顺序执行．使用本仿真程序时，只要按表 1 的格式在规定的标号内书写数据和方程就行了，不需要对 BASIC 语言的其他知识有更多的了解．

表 1　书写 BASIC 仿真程序的格式

语句标号	句首定义符	内容
1—4	print	运算题目的标题
70, 71	data	状态变量的初始值及初选的参数值
86—88	input	交互运算时需输入的参数名字表
100—119	data	成批处理时分组写出每次运算的参数值
120—139	read	按要求的顺序每次读入一组值
200—298	let	按微分方程子程序的规定书写系统的微分方程组
383—386	let	$A(i) = Y(j), i = 1, 2, 3, 4$, $Y(j)$ 为需输出的变量

四、操作说明

若 BASIC 编译程序已经开工并处于动态停机：

1)用键盘指令 NEW 清工作区；

2)把按表 1 写的方程及数据用输入设备如光电输入机或键盘输入到计算机；

3)用键盘指令 LIST 打印出方程及数据清单，检查无误后，用输入设备输入本仿真程序；

4)用键盘指令 RUN 使计算机运行，根据计算机提问输入：$T1$, $T2$, $N1$, $N2$, $N3$ 和 $L4$，成批处理时，还需输入运算次数 $j1$．

通过以上操作，计算机在输出设备上将给出所有运算结果。

五、卫星轨道的计算

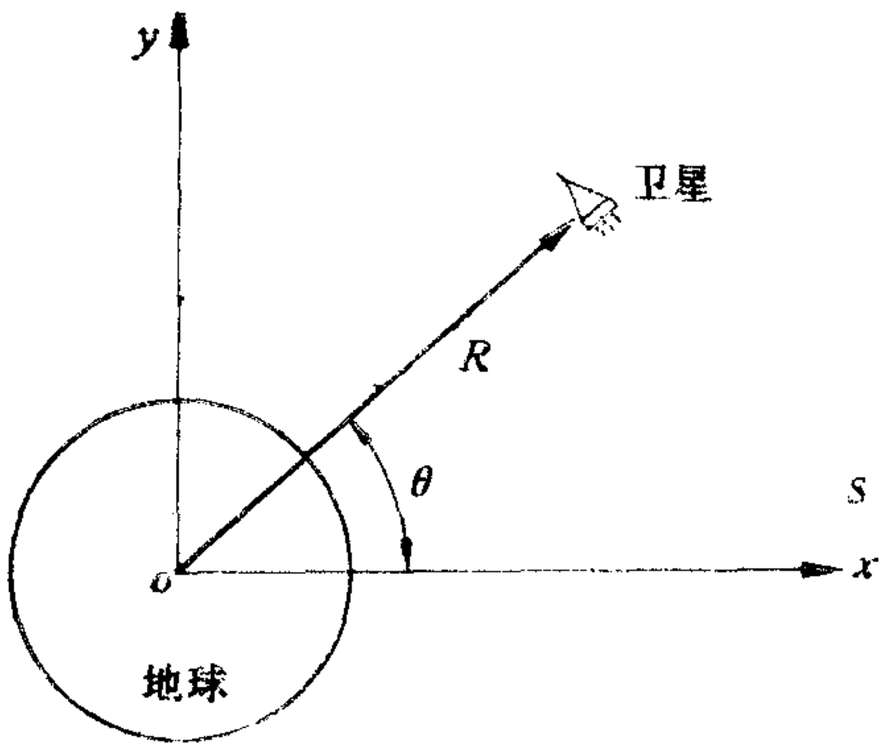

图 3 用极坐标表示的卫星轨道

为了具体说明 BASIC 仿真程序的使用，我们研究卫星轨道的仿真问题.为简单起见，假定作用在卫星上的力仅仅是地球的引力，极坐标运动方程是(见图 3):

$$\left.\begin{aligned}\frac{d^2R}{dT^2} &= -\frac{K}{R^2} + R\left(\frac{d\theta}{dT}\right)^2\\ \frac{d^2\theta}{dT^2} &= -\frac{2}{R}\frac{dR}{dT}\frac{d\theta}{dT}\end{aligned}\right\} \tag{2}$$

其中K是重力常数. 设 $Y(1)=R$, $Y(2)=\theta$, 得到

$$\left.\begin{aligned}\frac{dY(1)}{dT} &= Y(3)\\ \frac{dY(2)}{dT} &= Y(4)\\ \frac{dY(3)}{dT} &= -\frac{K}{Y(1)*Y(1)} + Y(1)*Y(4)*Y(4)\\ \frac{dY(4)}{dY} &= -2*Y(3)*Y(4)/Y(1)\end{aligned}\right\} \tag{3}$$

这是 4 个一阶的状态方程，如果希望用直角坐标输出的话，再引入两个定义变量 $X=Y(21)$, $Y=Y(22)$, 又得到两个代数方程:

$$Y(21)=Y(1)\cos(Y(2))$$
$$Y(22)=Y(1)\sin(Y(2))$$

根据卫星的发射速度，可以建立起方程组(3)的初始条件:

$Y(1)_0=$ 卫星到地心的距离，为简单起见假定为地球半径，即 6400 公里

$$Y(2)_0=Y(3)_0=0$$
$$Y(4)_0=V/Y(1)_0$$

现在研究不同的发射速度V对卫星轨道的影响. 取 $V=8$ 公里/秒, 10 公里/秒, 12 公里/秒, $K=401408$ 公里^3/秒. 按照上面规定的格式写出方程及数据，用键盘输入到计算机后，先列出清单(见表 2).

检查无误后输入 BASIC 仿真程序，然后用键盘指令 RUN 使计算机工作，计算机提问运行时间及积分步长:

run time t_1, t_2

用键盘回答 10000 及 100，即运算时间为 10000 秒，步长为 100 秒. 回答完毕，计算机又提问:

number of orders, parameters and outputs n_1, n_2, n_3

用键盘回答 4, 0, 51. 即方程阶次为 4，参数个数为 0，输出点数为 51. 接着计算机又提问:

表 2　方程及数据清单

```
List
  1  print tab (10): 'calculation of a satellite orbit'
  2  print
 70  data 6400, 0, 0, .00125
100  data .00125, 1.5625e-3, .001875
120  read y(4)
121  print 'y(4)=':y(4)
201  let y(21) = y(1) * cos(y(2))
202  let y(22) = y(1) * sin (y(2))
203  let g(1) = y(3)
204  let g(2) = y(4)
205  let g(3) =- 401408/(y(1)*y(1)) + y(1)*y(4)*y(4)
206  let g(4) =- 2*y(3)*y(4)/y(1)
383  let a(1) = y(21)
384  let a(2) = y(22)
385  let a(3) = y(1)
386  let a(4) = y(2)
```

if the operation is interactive $L4 =$

用键盘回答 1，即不作交互运算，要求成批处理运算．最后计算机再提问

number of runs $j1 =$

用键盘回答 3，即先后对 3 个不同的 $Y(4)_0$ 进行计算．到此全部人机对话结束，计算机按以上要求自动计算完毕，对每一个 $Y(4)_0$ 打印出结果和曲线，表 3 和图 4 分别给出对应于 $Y(4)_0$ 为 0.00125 的计算结果．

计算精度达到 5 位有效数字，计算时间共 210 秒(不算打印输出时间)．

输出完毕，计算机处于准备状态，如还需改变参数及初值，只要重新输入语句 70 和 100，再用键盘指令 RUN 使计算机运行即可．

表 3　对应于 $Y(4)_0 = 0.00125$ 的卫星轨道计算结果，其中
$a(1) = Y(21)$, $a(2) = Y(22)$, $a(3) = Y(1)$, $a(4) = Y(2)$

```
run
                    calculation of a satellite orbit
run time t1, t2# 10000# 100
number of orders, parameters and outputs:n1, n2, n3# 4# 0# 51
if the operation is interactive L4 = #1
number of runs j1 = #3
Y(4) = .00125
```

Y(*i*)	Y(*i*)max	Y(*i*)min		
1	6666.5	0		
2	611.9754	0		
3	.159322	−.160041		
4	.00125	0		
21	6399.99	−6665.04		
22	6508.26	−6530.11		
t	a(1)	a(2)	a(3)	a(4)
0	6399.99	0	6400	0
200	6205.07	1583.68	6403.98	.24989
400	5632.8	3071.17	6415.65	.499174
600	4719.82	4373.02	6434.27	.747277
800	2523.72	5412.76	6458.68	.993718
1000	2118.53	6131.68	6487.35	1.23813

表 3 续

1200	589.172	6491.88	6518.55	1.48029
1400	−974.398	6477.58	6550.46	1.1201
1600	−2482.77	6094.95	6581.23	1.95762
1800	−3852.12	5370.51	6609.17	2.19302
2000	−5008.31	4348.61	6632.76	2.42658
2200	−5890.2	3088.37	6650.76	2.65868
2400	−6452.04	1660.21	6662.22	2.88974
2600	−6665.04	142.236	6666.56	3.12026
2800	−6518.37	−1383.42	6663.56	3.35072
3000	−6019.49	−2834.28	6653.37	3.58165
3200	−5193.9	−4131.2	6636.51	3.81352
3400	−4084.31	−5202.18	6613.86	4.04678
3600	−2749.19	−5985.43	6586.61	4.28182
3800	−1260.47	−6433.92	6556.22	4.51893
4000	299.527	−6517.5	6524.38	4.75832
4200	1842.07	−6226.1	6492.89	5.00005
4400	3276.87	−5571.27	6463.6	5.24406
4600	4517.67	−4587.15	6438.27	5.49016
4800	5487.94	−3328.52	6418.46	5.73799
5000	6126.7	−1868.95	6405.42	5.9871
5200	6393.14	−295.86	6399.98	6.23694
5400	6270.1	1295.23	6402.48	6.48689
5600	5765.53	2807.56	6412.77	6.73635
5800	4911.75	4149.96	6430.19	6.98472
6000	3762.8	5243.16	6453.63	7.2315
6200	2390.11	6024.88	6481.64	7.47632
6400	876.957	6453.21	6512.52	7.71892
6600	−687.121	6508.26	6544.44	7.95917
6800	−2212.24	6192.25	6575.56	8.19711
7000	−3613.19	5528.07	6604.15	8.43289
7200	−4813.77	4557.03	6628.64	8.66677
7400	−5750.22	3335.78	6647.75	8.89911
7600	−6373.83	1932.91	6660.47	9.13034
7800	−6652.57	425.402	6666.16	9.36092
8000	−6572.25	−1105.08	6664.51	9.59197
8200	−6136.97	−2575.85	6655.62	9.82218
8400	−5368.92	−3906.86	6639.94	10.0538
8600	−4307.75	−5024.43	6618.27	10.2868
8800	−3009.18	−5864.83	6591.76	10.5215
9000	−1542.84	−6377.88	6561.83	10.7582
9200	10.5649	−6530.11	6530.12	10.9972
9400	1563.14	−6307.62	6498.43	11.2385
9600	3024.5	−5717.96	6468.6	11.4821
9800	4307.18	−4790.92	6442.41	11.7279
10000	5332.42	−3577.79	6421.48	11.9754

六、结 论

通过以上实例的计算可以看出，这个仿真程序非常方便和灵活，它能完成很广泛一类系统的动态仿真．应用这个仿真程序的工程技术人员，不必对 BASIC 语言十分熟悉，只要按规定的格式写入方程及数据即可，为他们使用计算机提供了便利，节约了时间及精力．

如用 FORTRAN 等编译程序编写连续系统的仿真语言，还可以大大提高运算速度．

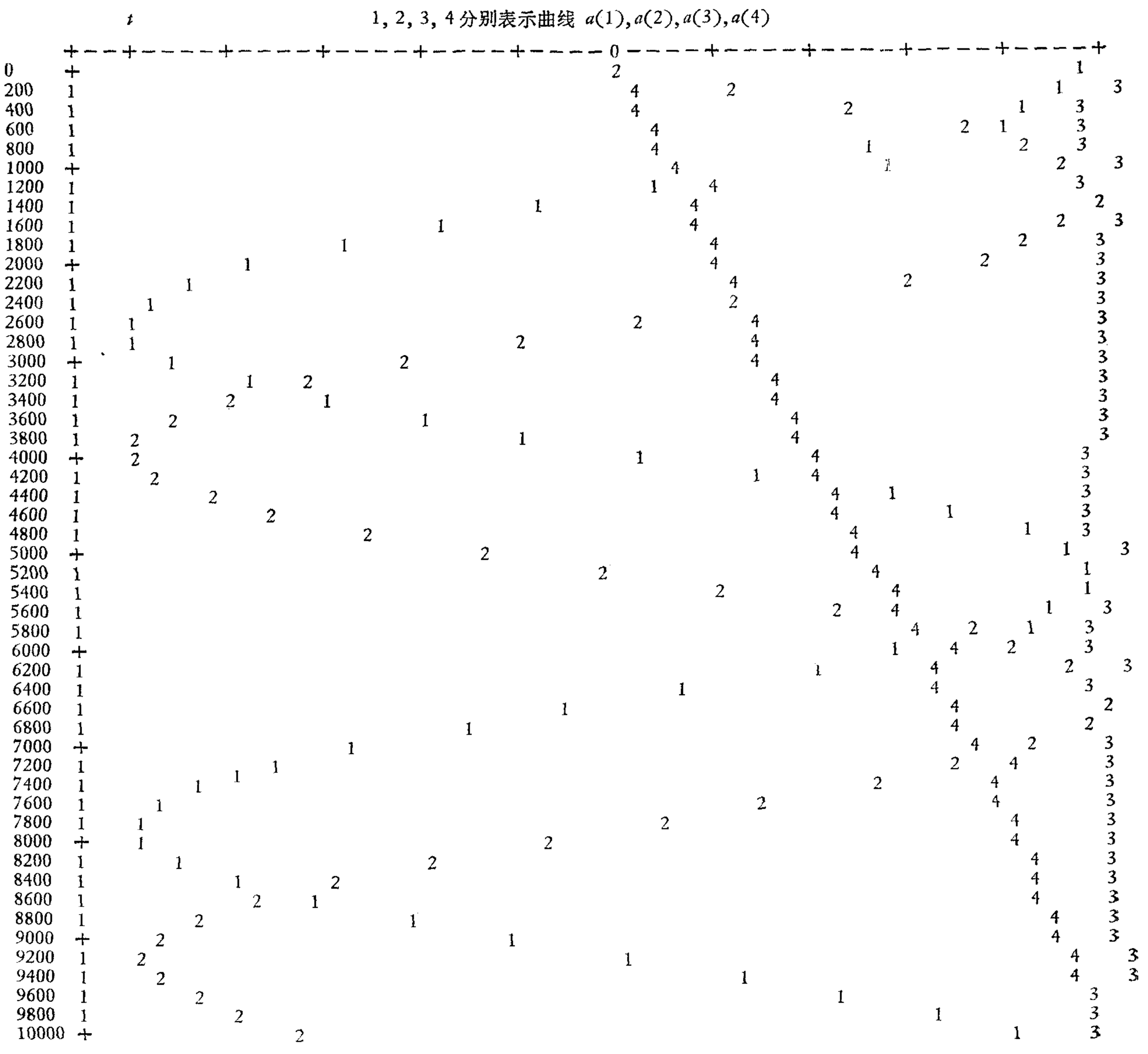

图 4　对应于 $Y(4)_0 = 0.00125$ 的卫星轨道曲线

参　考　文　献

[1] Manesh J, Shah, Engineering Simulation Using Small Scientific Computers, Prentice-Hall Inc. 1976.

[2] Korn G. A. et al., Digital Continuous-System Simulation, Prentice-Hall Inc., 1978.

[3] Albrecht, Robert L. et al., BASIC; teach yourself the quick proven way with programmed instruction. New York, Wiley, 1973.

THE METHOD OF DIGITAL CONTINUOUS SYSTEM SIMULATION

Liu Zhi-jun Li Bao-shou

(*Beijing of Institute Control Qngineering*)

ABSTRACT

Recently, digital computers are gradually used to replace analog or hybrid computations for implement of continuous-system simulations. Digital continuous-system simulation has significant advantages: better man-machine interaction, higher accuracy and reproducibility, more convenient programming and report generation, etc. The simulation user would like to have a more convenient and flexible simulation language, in order to be as free as possible from the details of computer programming and concentrate on his system simulations. An equation-oriented BASIC simulation language which permits the user to enter first-order differential equations in essentially unchanged mathematical form is introduced in this paper. It may run on any machine which supports BASIC. The user does not even have to be familiar with BASIC so long as he follows the simple format specified. A computing example is given to illustrate its application.