THE PARAMETER QUASI-OPTIMIZATION FOR SYSTEM DYNAMICS MODELS

Qifan Wang Guangle Yan

System Dynamics Group
Shanghai Institute of Mechanical Engineering
Shanghai, China

ABSTRACT

With the sustained development in science and technology, the method of system optimization has more and more widely been applied to the area of science, technique, engineering, economy, etc. The optimization theory is strongly supported by the birth and the development of computers. System dynamics models are good at understanding complex systems with the characterisics of high-order, multi-loops and nonlinear (say socio-economic systems). The purpose of this paper is to combine the modeling process of a system dynamics model with the optimization method so as to make the system synthesis more perfect. Because of the specific properties of large scale systems, there are some serious difficulties in completely optimizing systems. In many cases, it is impossible to find an overall optimization for complex systems. However the quasi-optimization for dynamic systems is still available. This paper develops some ideas of the parameter quasi-optimization for system dynamics models and presents a practical method. Its advantages include that the goal of the parameter quasi-optimization is clear, the precision is controllable, the quasi-optimal indices are conveniently regulated, the whole process can be automatically completed by a computer, repeating computations and simulations are not needed. Besides system synthesis, this method can also be applied to system analysis such as the parameter quasi-optimization after decoupling a system, selecting the dominant loops, separating the interest substructures, etc. The final goal of this paper is to make a solid fundation for a common used modeling software package.

PARAMETER QUASI-OPTIMIZATION FOR SYSTEM DYNAMICS MODELS

INTRODUCTION

The system simulation technique has been more and more widely applied to analyzing and synthesizing complex systems. While there are still much to do in the combination of system simulation and system optimization. The complete process of system analysis and system synthesis mainly contains three phases: modeling, simulating and optimizing, simplely note M-S-O. Here exist two relative links. One is the combination of M and S, called "Decision Support System Based on Simulation." This link has been paid much attention to in modeling and analyzing system dynamics models. The other is the combination of S and O, called "Simulation Optimization Techniques." Usually the system optimization of system dynamics models relies on subjective experience. First estimate some sets of system parameters, compare all the the results, then select the best one. Obviously this "trial and error" method is time-consuming and costly. It is actually difficult to get the most optimal results. Therefor the problem of deeply studying the simulation optimization techniques has laid before us.

SYSTEM OPTIMIZATION AND QUASI-OPTIMIZATION

System optimization is actually to seek a optimal or quasi-optimal one from all the results in a system by the help of some mathematical algorithem for evaluating extreme values. There are some characteristics in system optimization. First, system optimization is governed by the objective functions. Different objective functions lead to different optitimal scheme and produce different results. A scheme is optimal for some objective functions and is not optimal, even is bad, for some other objective functions. Second, system optimization is governed by the time period system working. Any optimal scheme always relates to some relative time period. Long term optimal schemes generally contradict short term optimal schemes. In addition, system optimization is governed by system boundaries. An optimal scheme for subsystems is not always optimal for their whole system. There exist various complicated connections, exist the exchanges of imformation and energy between subsystems or subsystems and the whole system. It can be proved in theory that the local optimization of a system is global optimal if and only if the system is definited in a convex set, otherwise the system has no such a property.

System optimization technique is actually a kind of mathematical method. Some optimal schemes generally exist in a system in theory for some special time periods, special space and special objective functions, but practically systems do not always work by the theoretical optimal schemes. The main reasons include:

1) Various indices in a system usually restrict, sometimes contradict one another. It is impossible to make all the indices arrive at ideal optimization.

2) The optimal results in a system for a long term period generally go against those for a short term period.

3) Actual complex systems are difficult to be satisfied with convex conditions.

4) Under the permission of technical indices, an experienced modeler would rather take the advantages of simple mathematical calculations and convenient practical applications than make the model compeletely accurate with a higher cost.

5) Sometimes mathematical calculations are difficult for complex systems, therefor some approximate calculations have to be adopted.

6) The realizations of theoretical optimal schemes are sometimes difficult because of the limitations of facilities conditions.

Because of all the reasons above, real complex systems can only arrive at an area near some optimal operative point in stead of at the completely optimal operating point. Naturally the final results can only be quasi-optimal.

However the limitations to the operation of real systems are not extremely strict. Some schemes work very well in practice, but they are not theoretically optimal. Therefor if we say studing and exploring system optimization has theoretical meaning in guidenees and elicitation, studying and exploring system quasi-optimization has practical meaning.

PARAMETER OPTIMIZATION ALGORIYHM

System dynamics models are good at understanding large scare complex systems with the characteristics of high order, multiple loops and nonlinear. The more complex a system is, the farer the distance between its "cause" and "effect" either in time or in space. A behaviour change in a complex sysyem may be caused by some trivial factors. The present behaviour of a complex system may be caused by the changes of various factors long before. This counterintuitive behaviour intrinsically existing in complex systems make the realization of system optimization greatly difficult.

The characteristics of a complex system are that it contains complicated nonlinear feedback loops and a lot of variables and parameters. Under the influence of nonlinear characteristics, on one hand, systems are insensitive to the changes in many system parameters. They stubbornly resist the policy changes. On the other hand, systems are specially sensitive to some "leverage points." This two-flod properties of a complex system enlighten us that we can avoid the resistive actions of insensitive parameters and effectively quasi-optimize systems only by carefully refining a few sensitive parameters.

Because of the nonlinear properties, the gradient information in a complex system can not be easily obtained. Algorithms without calculating derivatives can be only used either to optimize or to quasi-optimize parameters. Here we adopt the variable polyhedron algorithm.

The basic ideas of the variable polyhedron algorithm are like this: In an n-dimensional space E^n, construct a polyhedron with n+1 apexes, in which any n apexes do not lie in an (n-1)-dimensional sub-space of E^n . Calculate the indices value for each apex. Select an apex which produces a maximum indices value, and reflect it through the core of other apexes with the hope to find another apex which can produce a smaller indices value so as to construct a new polyhedron. Step by step we can get some better results. If we do not get satisfied apexes after some reflection, the polyhedron should be condensed or expanded to form a new polyhedron. Repeating this process, the apexes of the polyhedron would convergent to the local minimum value of the objective functions. The flow chart of the variable polyhedron optimization algorithm is shown in Figure 1.

PARAMETER QUASI-OPTIMIZATION OF A TYPICAL MODEL

According to the basic ideas of system parameter quasi-optimization, we take the famous Prey-Predator model for example to test the simulation quasi-optimization program

The problem of prey and predator was first put forward by Italian biogist Umberto D'Ancona in 1920's. At that time he observed the ten years' data of the fishing ammount at the port of Fiume from 1914 to 1923. He found the ratio of the fishing ammount of predator fish (e.g. shark) during the war period is much higher than in peace time. To make it clear, Italian mathematician Vito Volterra did a lot of research work. Finally he developed a mathematical model to discribe this problem:

$$\dot{x}=ax-by-ex=(a-e)x-bxy$$

$$\dot{y}=-cy+dxy-ey=-(c+e)y+dxy$$

where x is the state of prey, y is the state of predator, e is the rate of fishing, a, b, c and d are constent factors. Its system dynamics flow chart is shown in Figure 2.

This is an extremely typical system. A special model of this kind of systems is Rabbit-Lynx model. Now we take it for example to do the simulation optimization tests. The simulation structure of this model is shown in Figure 3, where T are TABLE functions. This system has strong nonlinear properties. It contains prey RABBIT and predator LYNX two level variables. The outputs of the two level variables in the ideal case are taken as optimization indices. Two factors CC (Carrying Capacity for Rabbits) and RRPLS (Required Rabbits per Lynx for Subsistence) are taken as controllable parameters. With the help of the variable polyhedron algorithm, we simulated and quasi-optimized the model by regulating the controllable parameters for some different initial parameter values. Four tests had been done. All the results are shown in Table 1. The outputs of level variables RABBIT and LYNX under ideal conditions (A) and under the 3rd test conditions (B) are separately shown in Figure 4 and Figure 5. We can see from all the results of the tests, the model after parameter quasi-optimization has been very near to the ideal system.

All the tests mentioned above were finished by the use of a simple micro-computer Apple-II. Each test spent about five minuts. The process of simulation quasi-optimization was programed in the FORTRAN language. The computer program is common-used. It can deal with various nonlinear units. It is an economical and practical tool in the realization of parameter quasi-optimization of system dynamics models.

Conclusion

Parameter optimization of system dynamics models is an important part of system analysis and system synthesis. System optimization improves system qualities and precision so as greatly to meet the gap between dynamics models and real systems. Compared with the "trial and error" method in parameter adjestment, parameter optimization (or quasi-optimization) has many advantages which include that the goal of the parameter quasi-optimization is clear, the precision is controllable, the quasi-optimization indices are conveniently regulatable, the whole process can be automatically completed by a computer,repeating computations and simulations are not need. In addition, parameter optimization can also be applied to system analysis, such as the parameter quasi-optimization after decoupling a system, selecting the dominant loops, separating the interest substructures, etc. The computer program for simulation and parameter quasi-optimization developed in this paper is flexible. It can simulate and optimize many kinds of complex systems by the help of a very simple micro-computer. It is an economical and practical tool in simulation and parameter quasi-optimization of system dynamics models.

TABLE 1.

test	controllable parameters	initial values	optimized values	ideal values
1	CC	2500	2399.89	2400
	RRPLS	180	199.70	200
2	CC	2200	2400.01	2400
	RRPLS	220	197.24	200
3	CC	3000	2400.04	2400
	RRPLS	250	198.87	200
4	CC	2000	2399.86	2400
	RRPLS	150	198.56	200

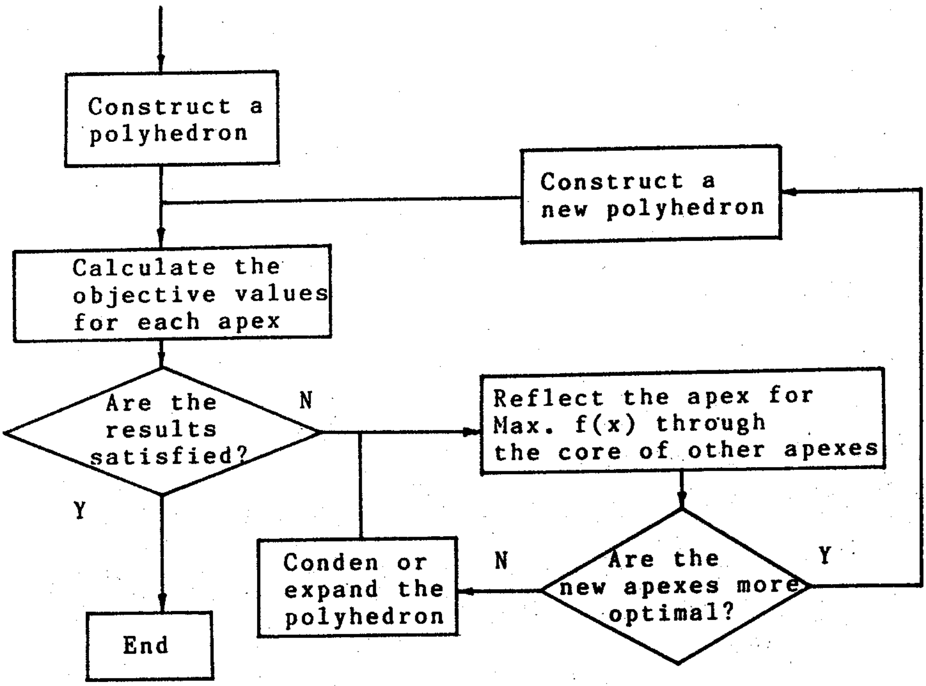

Figure 1.

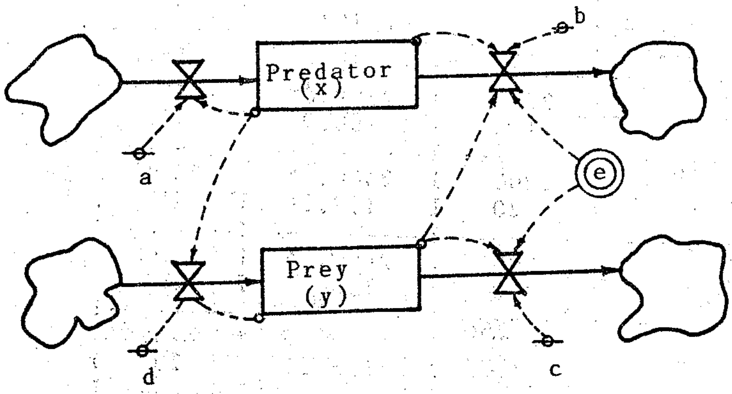

Figure 2

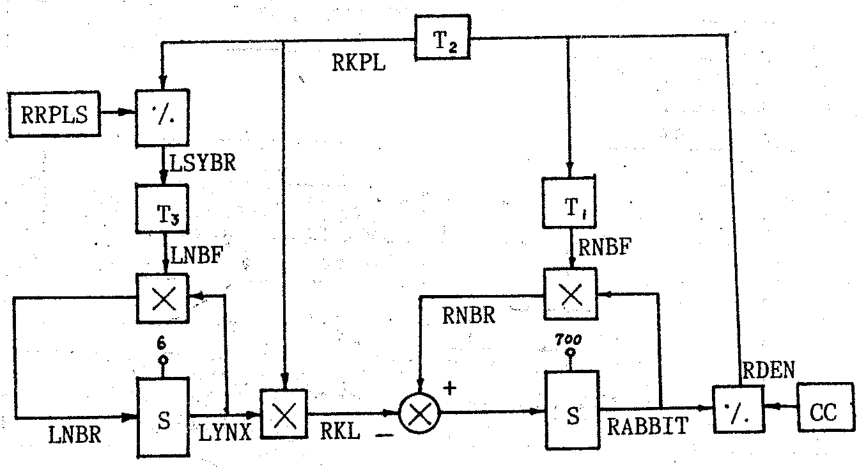

Figure 3

OP.DIV= 20.2884
TIME.DIV= .2500

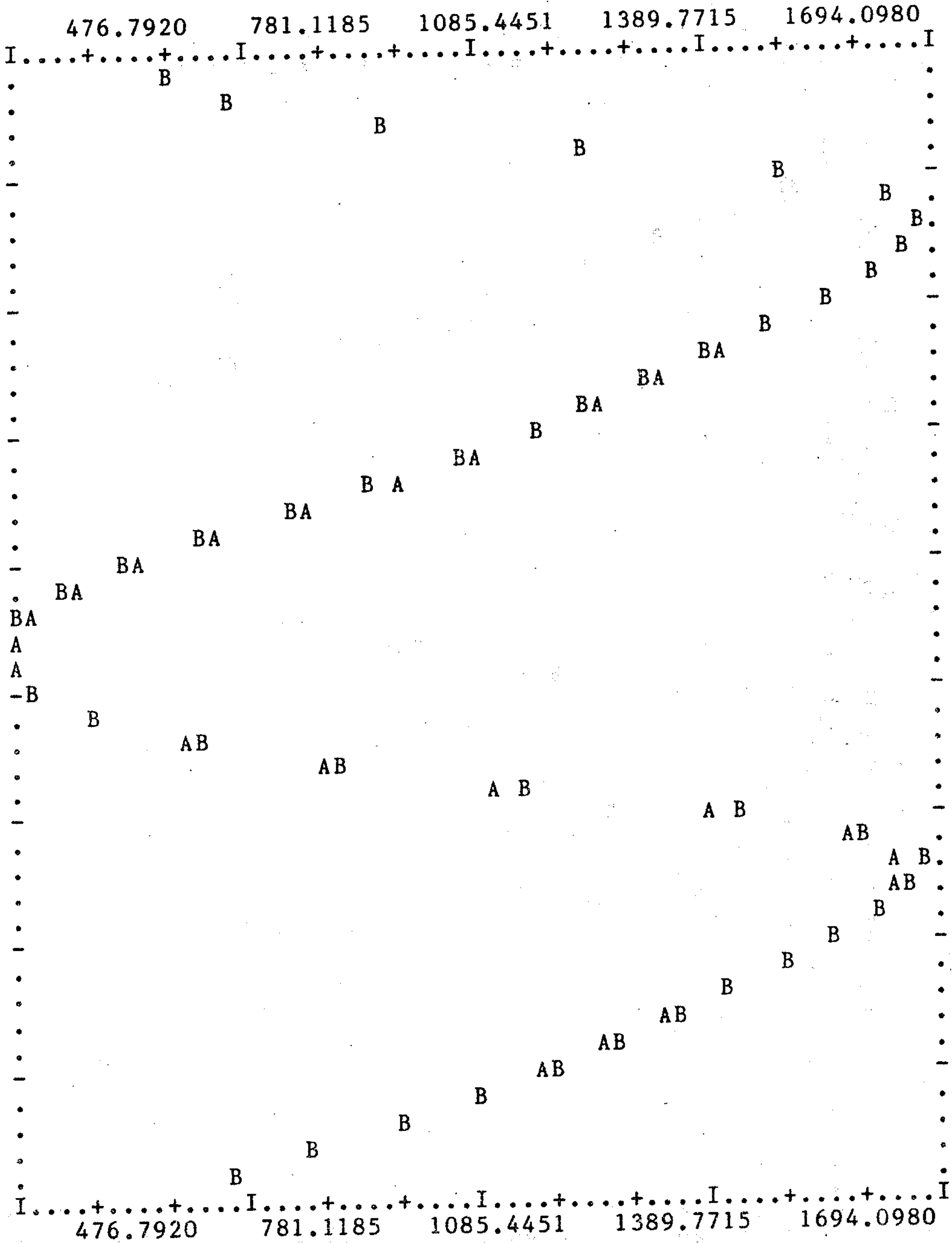

Figure 4. The Outputs of RABBIT under Ideal Conditions (A) and the 3rd Test Conditions (B)

OP.DIV= .1162
TIME.DIV= .2500

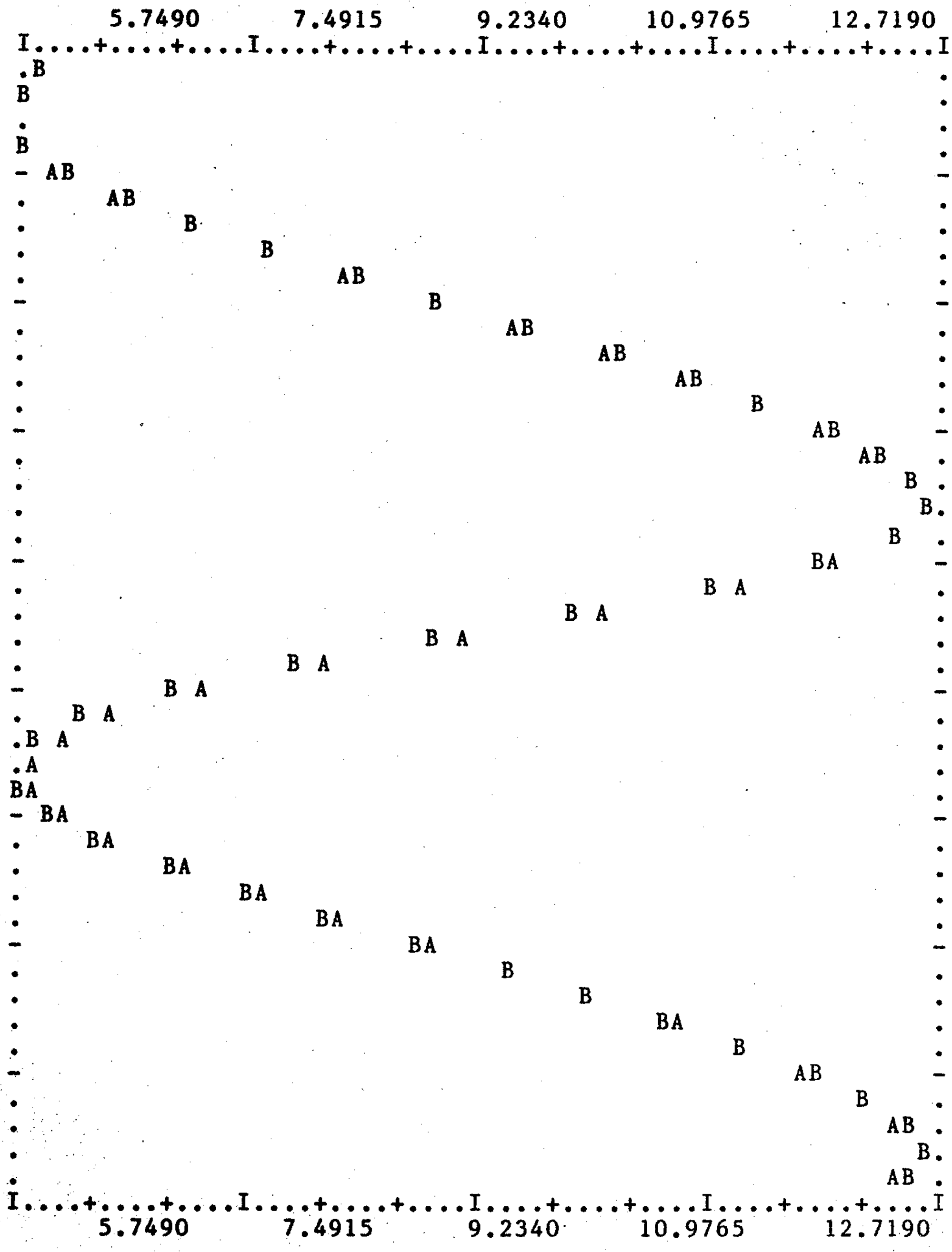

Figure 5. The Outputs of LYNX under Ideal Conditions (A) and the 3rd Test Conditions (B)

REFERENCES

1. Michaol, G. Ketcham, John W. Fowier, Ton T. Phillips, New Directions for the Design of Advanced Simulation Systems, WSC'84.

2. Charles R. Standridge, Integrated Simulation Support System Concept and Example,America Computer Society, 17th annual, Symposium on Simulation.

3. Alberts Garcia-Diaz, Hogg, G. L., Combined Simulation and Network Optimization Analysis of a Production / Distribution System, Simulation, No.2, 1983.

4. M. Braun, Differential equations and their Applications, 2nd Edition, New York: Springer Verlag, 1978.

5. Qifan Wang, On the Statistical and Estimative Problems of the Parameters of System Dynamics, Proceedings of the 1986 China National System Dynamics Conference.

6. Qifan Wang, A Study of the Complex Systems and Their Characteristics from the View of System Dynamics, Journal of Shanghai Institute of Mechanical Engineering, pp. 47-54, Vol.9, No.1, 1983.

APPENDIX

```
        *  RABBIT-LYNX MODEL
NOTE
NOTE    ---------------------------------------------------------
NOTE
NOTE    NAME    - RABBIT-LYNX MODEL
NOTE    AUTHOR  - WILLIAM SHAFFER
NOTE
NOTE    THIS MODEL REPRESENTS THE PREDATOR-PREY RELATIONSHIP
NOTE    BETWEEN RABBITS AND LYNX.  THIS RELATIONSHIP PRODUCES THE
NOTE    OSCILLATIONS IN POPULATIONS OF THESE ANIMALS.
NOTE
NOTE    ---------------------------------------------------------
NOTE
NOTE            RABBIT SECTOR
NOTE
L       RABBIT.K=RABBIT.J+(DT)(RNBR.JK-RKL.JK-RTRAP.JK)
N       RABBIT=INRAB
NOTE            RABBITS (RABBITS)
C       INRAB=700
NOTE            INTITAL RABBITS (RABBITS)
R       RNBR.KL=(RABBIT.K)(RNBF.K)
NOTE            RABBIT NET BIRTHS (RABBITS/YEAR)
A       RNBF.K=TABLE(TRNBF,RDEN.K,0,1.25,0.25)
NOTE            RABBIT NET BIRTH FACTOR (1/YEAR)
T       TRNBF=1.50/2.40/2.20/1.10/0.00/-1.00
NOTE            TABLE FOR RABBIT NET BIRTH FACTOR
R       RKL.KL=(LYNX.K)(RKPL.K)
NOTE            RABBITS KILLED BY LYNX (RABBITS/YEAR)
A       RKPL.K=TABLE(TRKPL,RDEN.K,0,1.25,0.25)
NOTE            RABBITS KILLED PER LYNX (RABBITS/LYNX-YEAR)
T       TRKPL=000/150/250/325/375/400
NOTE            TABLE FOR RABBITS KILLED PER LYNX
A       RDEN.K=RABBIT.K/CC
NOTE            RABBIT DENSITY (DIMENSIONLESS)
C       CC=2400
NOTE            CARRYING CAPACITY FOR RABBITS (RABBITS)
R       RTRAP.KL=(RABBIT.K)(FRABTR)
NOTE            RABBITS TRAPPED (RABBITS/YEAR)
C       FRABTR=0.0
NOTE            FRACTION OF RABBITS TRAPPED (1/YEAR)
NOTE
NOTE            LYNX SECTOR
NOTE
L       LYNX.K=LYNX.J+(DT)(LNBR.JK-LTRAP.JK)
N       LYNX=INLYNX
NOTE            LYNX (LYNX)
```

```
C       INLYNX=6
NOTE            INITIAL LYNX (LYNX)
R       LNBR.KL=(LYNX.K)(LNBF.K)
NOTE            LYNX NET BIRTHS (LYNX)
A       LNBF.K=TABLE(TLNBF,LSUBR.K,0,2,0.5)
NOTE            LYNX NET BIRTH FACTOR (1/YEAR)
T       TLNBF=-4.0/-0.6/0.0/0.3/0.5
NOTE            TABLE FOR LYNX NET BIRTH FACTOR
A       LSUBR.K=RKPL.K/RRPLS
NOTE            LYNX SUBSISTENCE RATIO (DIMENSIONLESS)
C       RRPLS=200
NOTE            REQUIRED RABBITS PER LYNX FOR SUBSISTENCE
NOTE            (RABBITS/LYNX-YEAR)
R       LTRAP.KL=(LYNX.K)(FLNXTR)
NOTE            LYNX TRAPPED (LYNX/YEAR)
C       FLNXTR=0.0
NOTE            FRACTION OF LYNX TRAPPED (1/YEAR)
NOTE
NOTE            CONTROL STATEMENTS
NOTE
SPEC    DT=0.125/LENGTH=30/PLTPER=0.5
PLOT    RABBIT=R/LYNX=L(5,25)
OPT     TXI=20
RUN
```