CHAOS IN WASSP

R J Marshall - Sheffield City Council October 1989

INTRODUCTION

What is CHAOS? The Concise Oxford Dictionary defines it as either : formless void or great deep of primordial matter or : utter confusion.

It is the latter definition that is applicable in this paper!

Chaos is the name given to what could be considered as a new branch of science which appears to have been identified and developed in the late 1970s at the Santa Cruz campus of the University of California. Basically it seems to be aimed at forming a better understanding of the many and varied random and unpredictable factors that influence everything that goes on around us.

CHAOS IN TERMS OF PREDICTABILITY v. UNPREDICTABILITY

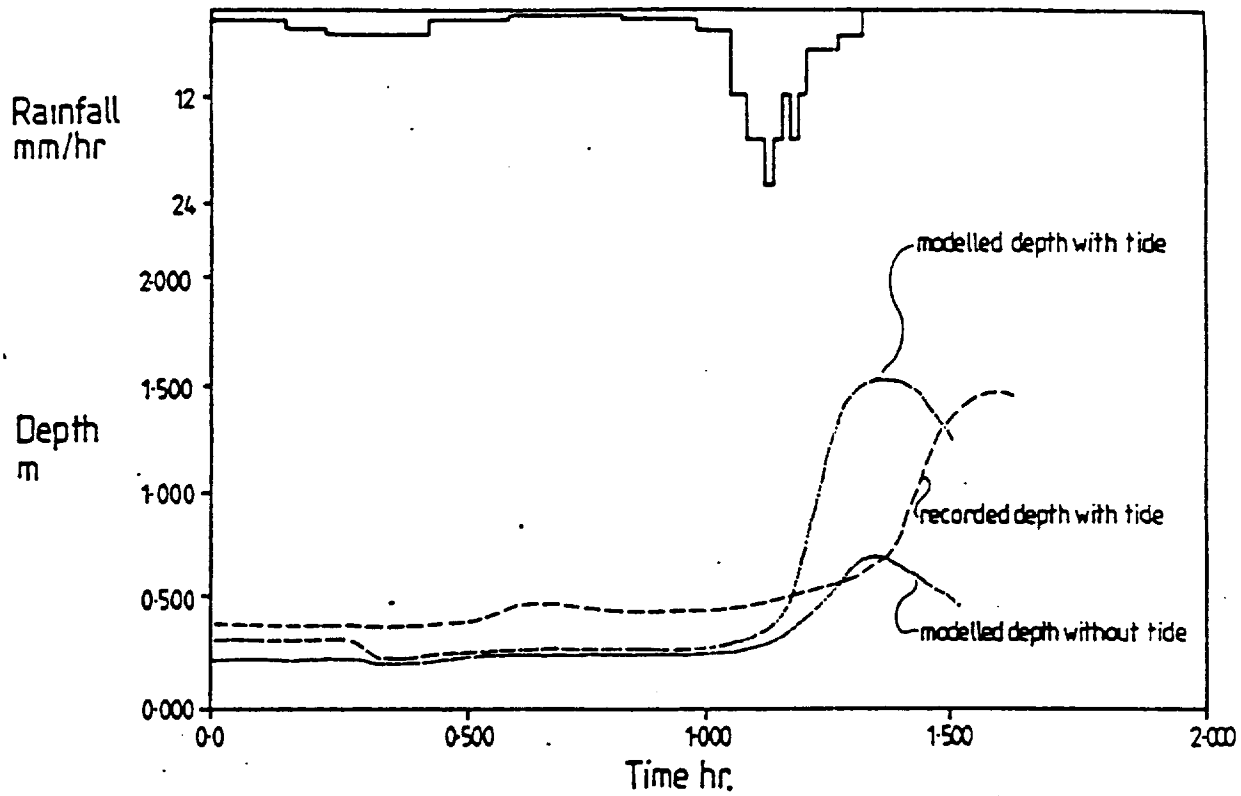
An example of this sort of behaviour which is relevant to sewerage engineers is as follows:

Consider the dry weather flow in the outfall sewer of a large city such as Sheffield. If there are no pumping stations near to the outfall, the dry weather flow will be reasonably predictable given historic data, the time required, the day of the week and some general feeling for the recent amount of precipitation (which will influence the rate of infiltration). None of these items would be too hard to come by and an estimate of the flow or depth could be made with a reasonable amount of confidence.

Now consider a length of sewer right at the top of the system. There is virtually no damping or attenuation and every flushing or emptying of a washing machine has a visible effect on flows and levels. Although we could gather data which may give us a good estimate of the total flow in a day or even perhaps for every hour of the day we certainly would not be confident of predicting the level of flow at any stated minute or second, it could vary from virtually nothing to several litres/second.

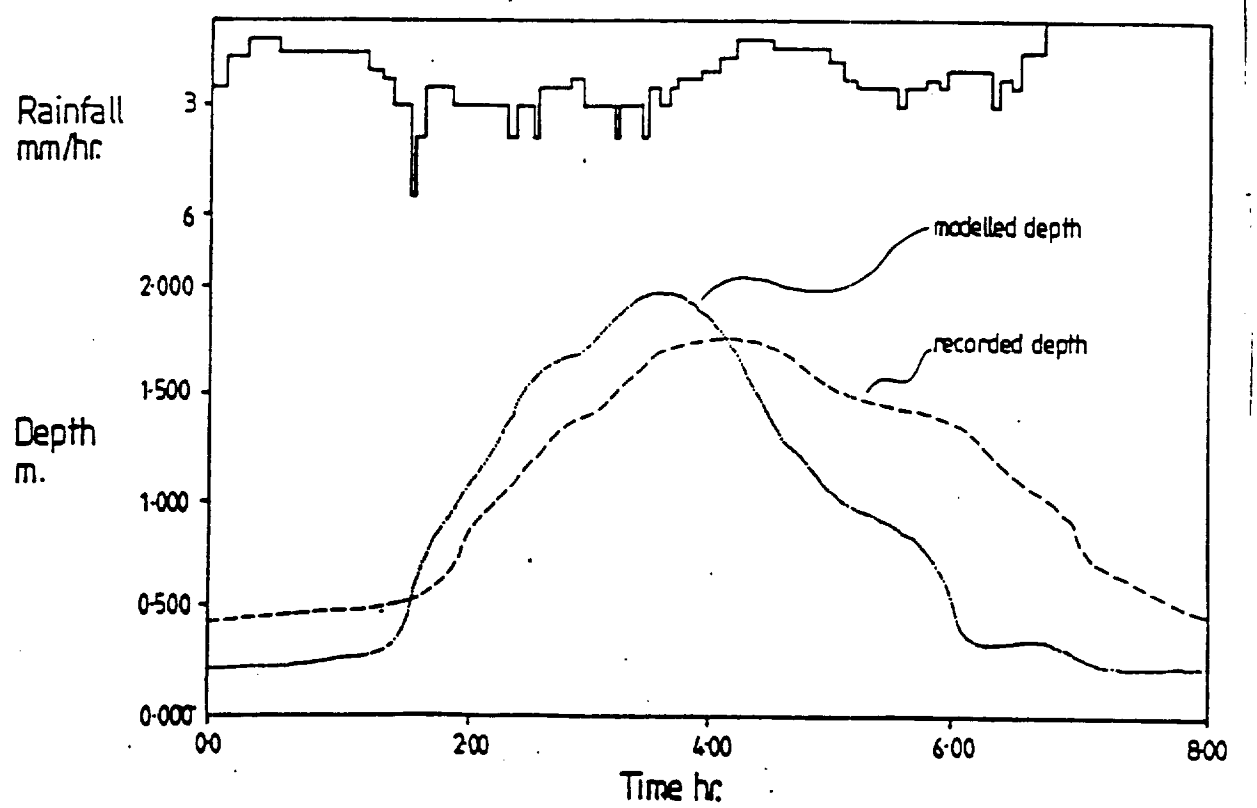
Another example would be the time interval between waves breaking on a beach. It would be possible to collect enough data to work out daily averages, but the actual interval between successive waves is by no means constant and depends on an infinite number of variables such as (obviously) the state of the tide, wind speed and direction, slope of the beach, wakes caused by various craft, (less obviously) by factors such as the interval between the previous pair of waves which, if short, may have created such a strong backwash as to prevent a wave breaking at all, and finally (not obvious at all) factors such as the currents caused by fish swimming nearby or even whether or not somebody decided to go for a swim three weeks ago on

MONTROSE EVENT 22.39-23.59 06/09/88



NEAP TIDE EVENT 6/9/88 RECORDED/MODELLED DEPTH COMPARISON

MONTROSE EVENT 22.42-05.21 25/10/88-26/10/88



SPRING TIDE EVENT 25/10/88 RECORDED/MODELLED DEPTH COMPARISON

VERIFICATION - CHECK USING TIDAL/STORM LEVELS

FIGURE 5

the other side of the ocean.

CHAOS IN TERMS OF MODEL STABILITY

The idea of the study of Chaos is to go further than the collection of data and producing global averages and factors for input into a model of some description and to attempt to look for, or at least be aware of, other influences and patterns.

Rainfall patterns of obviously of great interest to us and there are limitless opportunities for Chaos within the way the atmosphere is driven. Weather forecasters are now able to process vast quantities of data gathered from hundreds of sources and extrapolate the current situation and recent trends to produce their predictions. They are able to test the robustness of their model by altering the various input parameters within a suitable range and, if the outcome is similar say 5 days ahead for many variations of input within the specified limits, then they would have confidence in the prediction and the situation would be damped or stable. On occasions however the weather systems are such that small changes to the input data seem to have an increasingly dominant effect and thus it becomes impossible to put much confidence to the predictions even a few hours ahead and the system would be turbulent or chaotic.

Chaos is thus more of a way of looking at how systems operate and understanding the possible effects of all of the influences which are not allowed for by the data used to model a system.

A SIMPLE UNSTABLE TANK MODEL

There was an illustration in Nick Orman's paper in Spring 1989 which showed a plot of a hydrograph exhibiting totally unstable behaviour, this being a typical example of the instability problem in the tank model. This reminded the author of a chaotic feature related to simple equations and an attempt was made to link the two phenomena.

Initially, a simple model was set up using a spreadsheet which simulated the attenuating effect of a tank with an orifice outlet. By having a fairly coarse time interval of 1 minute it is easy to set up the situation where the tank actually becomes empty half way through a time increment and the crude model's prediction overshoots and calculates a negative flow out of the tank. The next time increment predicts an equal positive flow out and so it carries on in a stable but oscillating manner (diagram a). Various shapes of input hydrograph were tried and, predictably, those which came sharply down to zero caused a wider amplitude to the oscillation than those which tailed off gently.

By adding some more inflow downstream at later time increments (i.e. a branch with a longer time of concentration) it was possible to make the oscillations appear on the declining side of the hydrograph. If a second tank is added it is possible to obtain some instability in the form of ripples on the rising side and over the peak of the output hydrograph (diagram b). The completely chaotic phenomenon has not yet been achieved in this manner although it must be said that the tests on this model have been by no means exhaustive. The routine used to predict each value of the attenuated hydrograph depends on the solution of a quadratic equation.

A SIMPLE UNSTABLE MATHEMATICAL MODEL

Predicting a situation by means of a quadratic equation, using the results of the previous generation as input, can have a very interesting effect.

If we take the simple equation $x=a.y-a.y^2$ it could be written in the form $x=y.a.(1-y)$. If we now suggest that y is in fact y_n (the current value) and x is y_{n+1} (the value of the next generation) we could write an equation $y_{n+1}=y_n.a.(1-y_n)$.

To make the comparison with the simple tank model, consider the additional volume of fluid stored in the tank in time increment (n+1) to be derived from the previous value multiplied by a growth factor (a) which could be related to an increase in incoming flow and also multiplied by a reducing factor $(1-y_n)$ which relates to the fact that as the amount stored increases, the depth increases and thus the head over the orifice and the pass on flow also increases.

How should this relationship behave for successive generations of y_{n+1}? If the growth factor (a) remains constant, we would expect the values of y_{n+1} to tend towards a value such that the reducing factor is the reciprocal of the growth factor and each new value of y_{n+1} would be the same as the previous. This situation would be comparable to a tank with an orifice being filled at a constant rate, the depth building up more and more slowly until a steady state is reached where there is sufficient depth over the orifice to pass flow out at exactly the same rate as it comes in.

A simple spreadsheet simulation can illustrate this. N.b. the possibility of using finer time increments does not occur here, we are only considering what happens to repeated generations. (please refer to diagrams 1 to 14)

It should be noted that the same starting value (y_1) of 0.8 is used as experimentation has shown that the actual value does not significantly alter the effects illustrated here. Only the growth factor (a) is changed between illustrations but it remains constant within each example.

<u>diagram 1</u> shows the following:

a=0.5 : y_n decays rapidly towards zero.
a=1.0 : y_n decays more gradually towards zero.
a=1.25 : y_n goes straight to 0.2 and stays there.
a=1.5 : y_n initially overshoots and then climbs back to 0.3333
a=2.0 : y_n overshoots then climbs rapidly up to 0.5
a=2.5 : y_n overshoots then climbs immediately back to 0.6

It is not too difficult to understand what is happening here. The value of y_n is tending to the value obtained when the growth factor is the reciprocal of the reducing factor hence y tends toward (1-1/a), (although y tends to zero if the result of (1-1/a) is negative as in the case of a=0.5.)

<u>diagram 2</u> after a=2.5, things start to get more interesting!

a=2.6, a=2.7, a=2.8, a=2.9 : y_n overshoots downwards, overshoots in the other direction and it takes about 15 generations for this oscillation to dampen out.
a=3.0 : the oscillation is still very apparent after 50 generations but is visibly dampening down.
a=3.1 : after a slight dampening in the first 10 or so generations, the dampening is so slight as to be invisible at the scale plotted.

<u>diagrams 3 & 4</u>

a=3.2 and a=3.4 are similar to a=3.1 but with increasing amplitude.

<u>diagram 5</u>

a=3.5 : a new complication seems to be appearing. It seems that y_n is now fluctuating on a 4 generation cycle.

<u>diagram 6</u>

a=3.6 : There is no discernible pattern in the first 50 generations, even though a few groups of cycles appear similar, closer scrutiny reveals differences. We now have a truly CHAOTIC situation. (remember the definition - "utter confusion"!)

<u>diagram 7</u>

a=3.7 : Still chaotic but with a visible preference for values between 0.6 and 0.8. Note some instances of consecutively rising values, not seen since a=2.0

<u>diagram 8</u>

a=3.9, a=3.9000001 : If the growth factor is only one part in 39 million different, the values of y become visibly different after only 27 generations. The differences are obvious at a much earlier stage on the spreadsheet of course. However they do appear to be more or less coincident after generation 48.

diagram 9

a=3.9, a=3.9000001 : The first 200 generations show that the apparent agreement around generation 50 was only a fluke.

diagram 10

a=3.9 : by itself to prove that there really were two separate traces in diagram 9. This shows a strong resemblance to some of the unstable hydrographs obtained from tank models. Also note the way the values often nearly reach the theoretical figure of 0.743589743, but then go chaotic again.

diagram 11

a=3.9 again but this time each value of y_n is plotted with y_{n+1}. At last there is some order in the Chaos. This is the first 50 values and the start and finish points looked a little odd.

diagram 12

as above but with the first 1000 points plotted. The abrupt start and finish to the parabola reflect the envelope of maximum and minimum points in diagram 10 but just why they occur has not been investigated by the author.

diagram 13

as above but with the addition of each point plotted against itself. The intersection is at the theoretical solution that the iteration never quite achieves.

CHAOS IN THE HARDWARE AND/OR SOFTWARE

diagram 14

a=5 : This is in many ways a curious graph. The growth factor of 5 is already the reciprocal of the reducing factor (1-0.8) so the overall factor is 1 and each generation should have the same value as the previous one. However, minute variations seem to creep in becoming visible on the fourth generation (see list below) and the result goes absolutely unstable. It should be noted that as a value greater than 1 is obtained at one stage, the following value is negative and the next few generations rapidly run towards an impossibly large negative value.

n	y_n	a	n	y_n	a
1	0.800000000000000	5	19	0.799823808490000	5
2	0.800000000000000		20	0.800528419310000	
3	0.800000000000000		21	0.798413345930000	
4	0.800000000010000		22	0.804747374840000	
5	0.799999999960000		23	0.785645187640000	
6	0.800000000110000		24	0.842034133890000	
7	0.799999999670000		25	0.665063256250000	
8	0.800000000990000		26	1.113770607200000	
9	0.799999997020000		27	-0.63357	
10	0.800000008950000		28	-5.17493	
11	0.799999973150000		29	-159.77387	
12	0.800000080560000		30	-128437.31767	
13	0.799999758330000		31	-82481365042	
14	0.800000725010000		32	-3.4015877896E+22	
15	0.799997824950000		33	ERR	
16	0.800006525120000		34	ERR	
17	0.799980424440000		35	ERR	
18	0.800058724760000				

One possibility would be minute inaccuracies in the computer's processor although it is curious that the values used in diagram 1 seem to be happy to be self perpetuating. A similar Apricot Xen-i gave identical results, as did a Compaq 286 and Compaq 386 all using the same spreadsheet viz. As Easy As. This naturally calls the software or compiler into question. When the same routine was tested on a Victor machine using Lotus 123 a similar but not identical unstable pattern was seen, with the error at decimal place 12 not showing until generation 10.

This phenomenon can be demonstrated using the following simple BASIC program:

```
10     INPUT "Y1";Y:INPUT "a";A:LPRINT "Y1=";Y;"a=";A
20     FOR I=1 TO 40:Y=Y*A*(1-Y):LPRINT I;Y:NEXT I
```

Using GWBASIC on the Apricot, with Y1=0.8 as before and the theoretically self replicating a=5 factor, the output was unstable right from the outset at only 7 decimal places, i.e. far worse than either spreadsheet!. How stable is FORTRAN?

WHAT MAY BE SUMMARISED FROM THE FOREGOING AND SOME OF THE IMPLIED QUESTIONS

Firstly, it must be realised that programs such as WASSP and Wallrus provide tools for extremely precise analysis of very imprecise data, and the user must always be aware that he will be most unlikely to achieve a model that reflects a real system under all operating conditions.

The user may care to run tests on the sensitivity of his model to changes in various input parameters in order to establish where he should target his data gathering activity.

Instabilities in the output from a computer model may be inherent in the equations used in the model. These may behave either completely predictably or totally unpredictably, the difference being caused by only a small change in the value of a simple input parameter.

In the past, advice has been given to alter parameters away from those which actually exist in order to cure a tank model of instability. Just how safe is this technique? Consider again the unstable quadratic equation. If our parameter was 3.9 (diagram 10) and the "solution" was y=0.74359, how relevant is it if we fiddle around and find a stable output with a<=2.5 and the "solution" is y<=0.6 (diagram 1)? In this example, the "solution" would be 20% underestimated. Is this the order of inaccuracy we get when we use an arbitrary coefficient just to obtain a stable output, and if so is it acceptable?

Is it possible to solve the problem by going straight to a solution instead of going through repeated iterations?

Has an unstable model been tested on different machines to see if identical results are obtained?

Are the mathematical functions built into the Fortran compiler sufficiently accurate and stable? Have different Fortran compilers been tried, and if so does this affect the stability of the model?

FURTHER THOUGHTS GENERALLY RELATED BUT NOT DEALT WITH IN THIS PAPER

It is known that real systems can exhibit instability and unpredictability in many ways. e.g. all of the flow into a bifurcation with equal exits can be induced to use only one of the exits, and in this situation, flow may even be drawn back out of the other exit. Small irregularities in the flow could cause the output situation to be reversed. Is this what our unstable models are trying to show?

For verification purposes, especially at the upstream end of catchments, it would be useful to have an automatic routine to smooth out the randomness of measured flow levels, probably with a 5 or 7 point moving average. This would enable a smoothed diurnal pattern of dry weather flow to be subtracted from a smoothed storm flow in order to isolate the flow generated by the event. This seems so obvious that there is probably such a routine in existence, but if this is so it could do with more publicity

How do we cope with real life unstable systems where standing waves move forward and back and possibly cause considerable lengths of pipework to go rapidly into a surcharged condition quite unpredictably?

CONCLUSION

The purpose of this paper has been to try to stimulate thought about what it is we are really trying to achieve when we model such an unpredictable subject as rainfall and hydraulic flow in non-ideal sewers and also to attempt to show that the basic tools of hardware, software and mathematical equations may not always behave quite as we would predict.

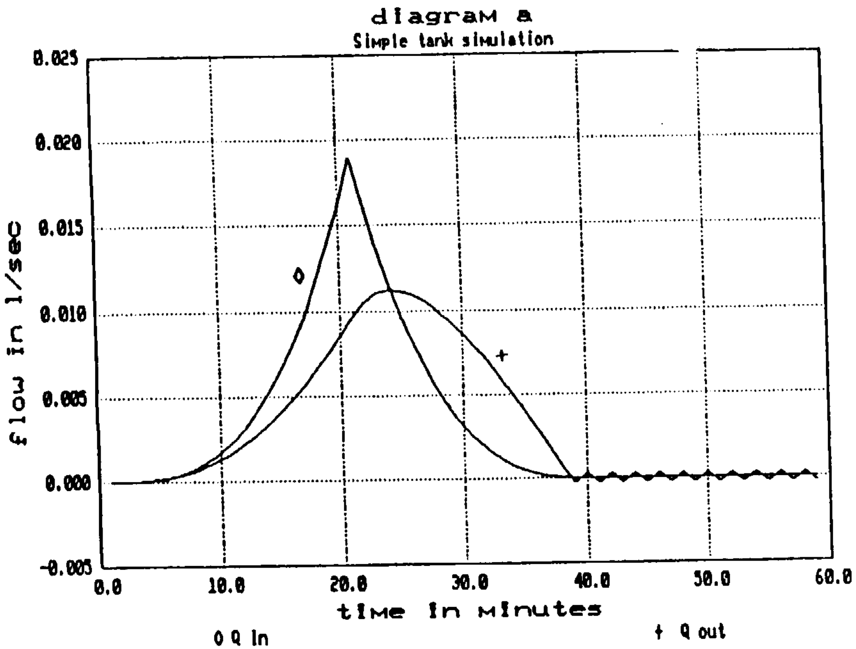


diagram a
Simple tank simulation
flow in l/sec
time in minutes
0.025
0.020
0.015
0.010
0.005
0.000
-0.005
0.0
10.0
20.0
30.0
40.0
50.0
60.0
◊ Q in
+ Q out

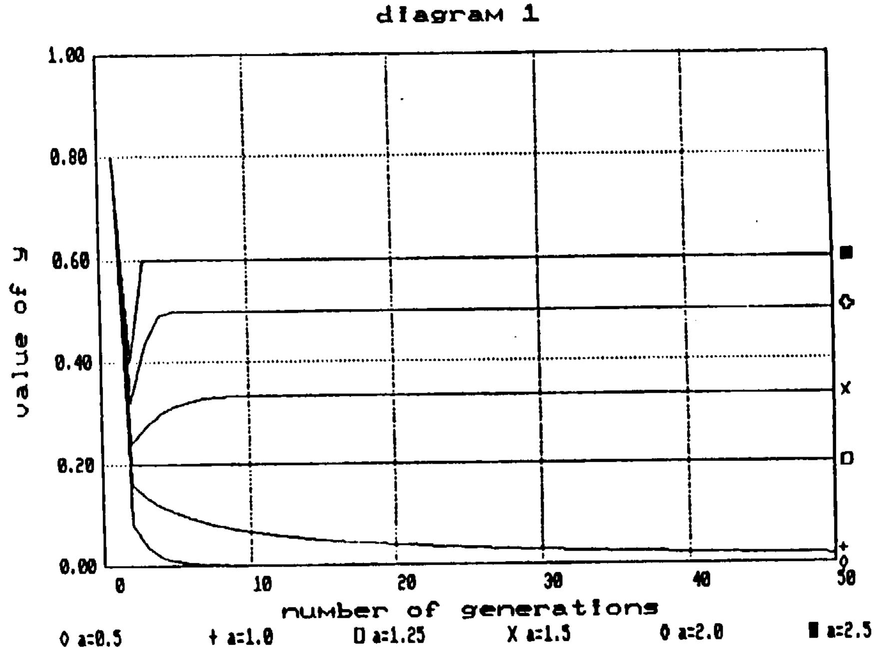


diagram 1
value of y
number of generations
1.00
0.80
0.60
0.40
0.20
0.00
0
10
20
30
40
50
◊ a=0.5
+ a=1.0
□ a=1.25
X a=1.5
◊ a=2.0
■ a=2.5

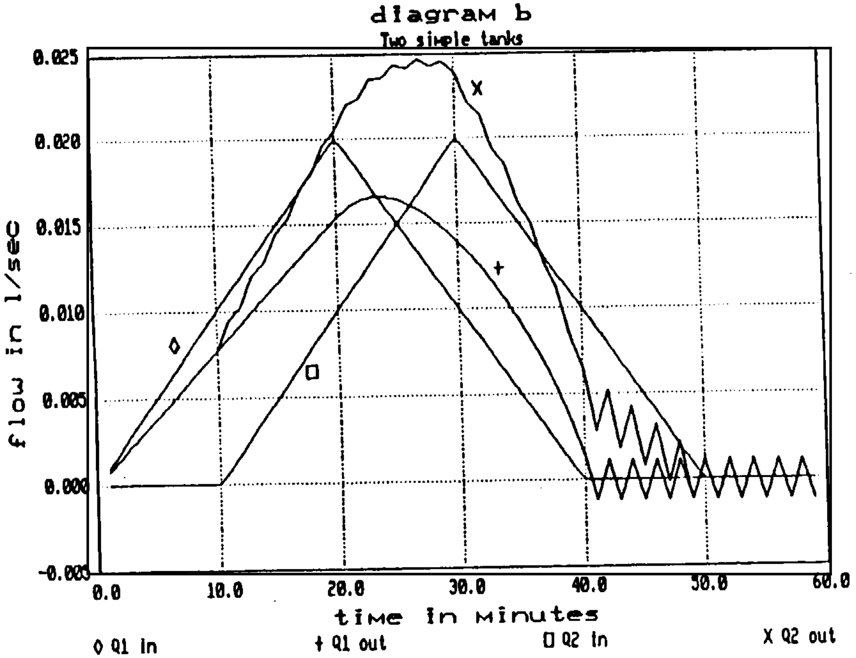


diagram b
Two simple tanks
flow in l/sec
time in minutes
0.025
0.020
0.015
0.010
0.005
0.000
-0.005
0.0
10.0
20.0
30.0
40.0
50.0
60.0
◊ Q1 in
+ Q1 out
□ Q2 in
X Q2 out

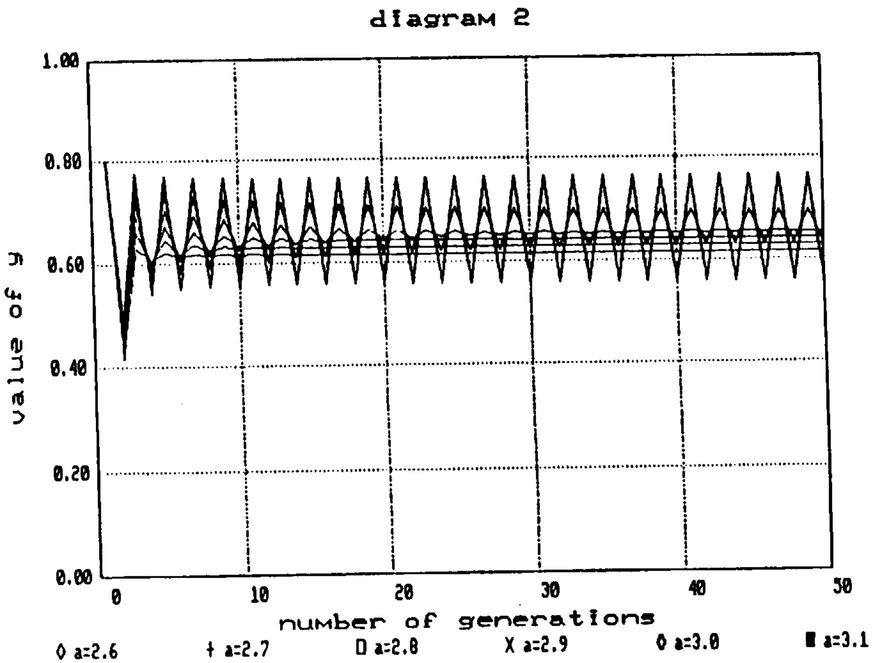


diagram 2
value of y
number of generations
1.00
0.80
0.60
0.40
0.20
0.00
0
10
20
30
40
50
◊ a=2.6
+ a=2.7
□ a=2.8
X a=2.9
◊ a=3.0
■ a=3.1

diagram 3

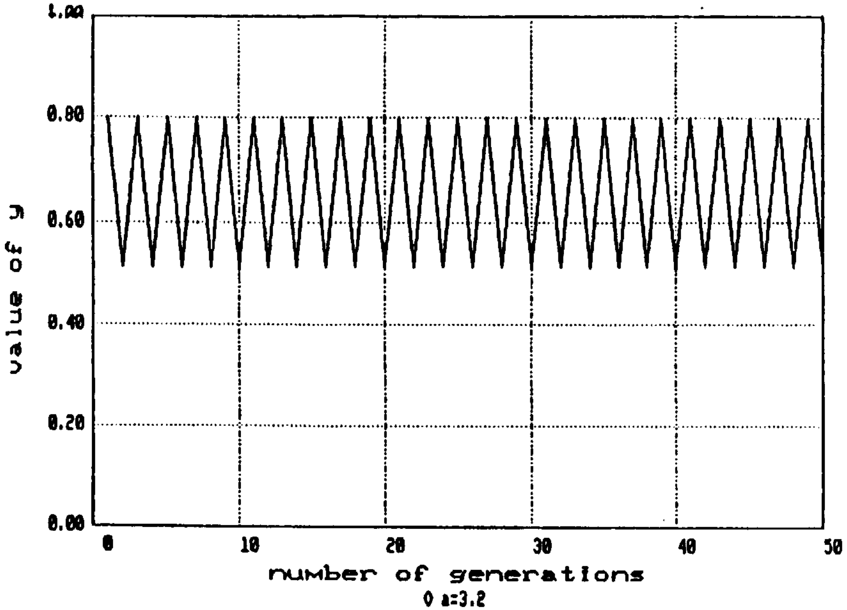


diagram 4

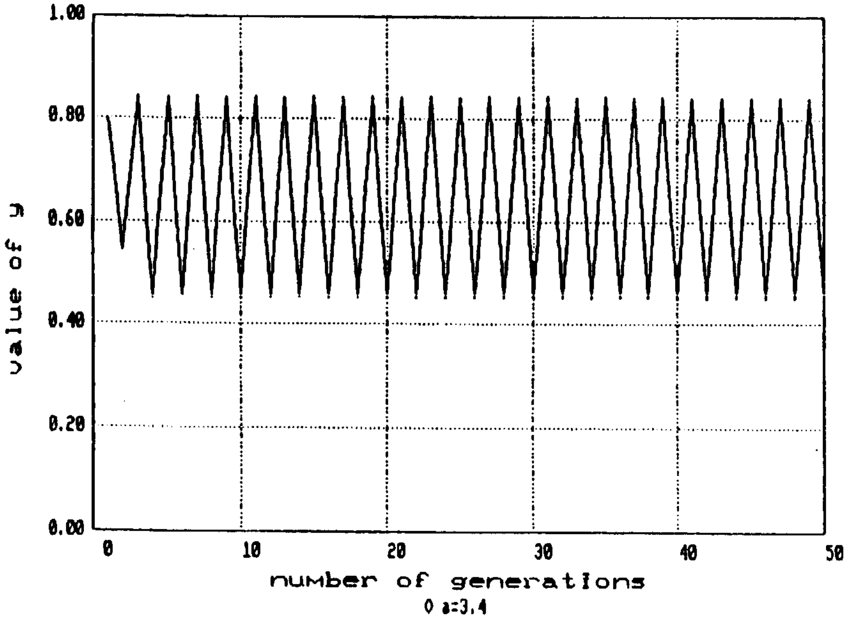


diagram 5

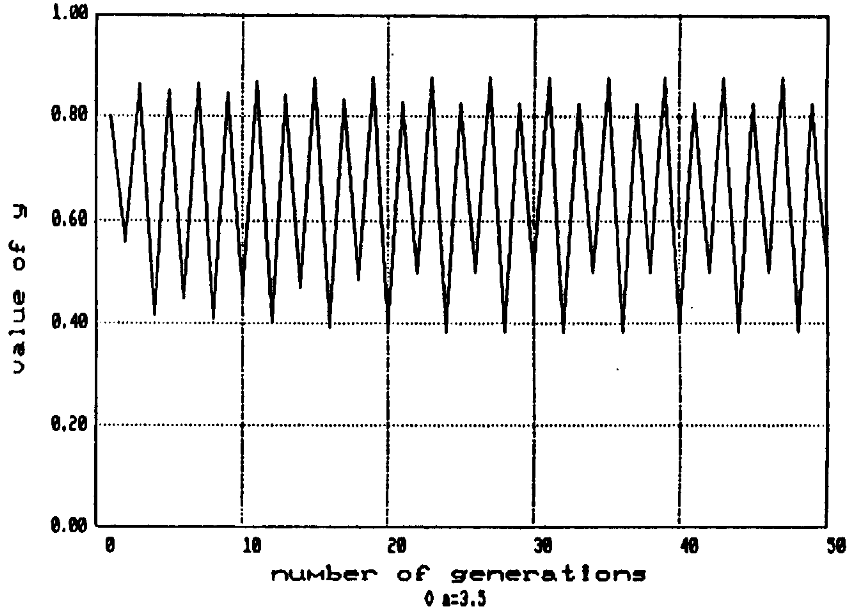


diagram 6

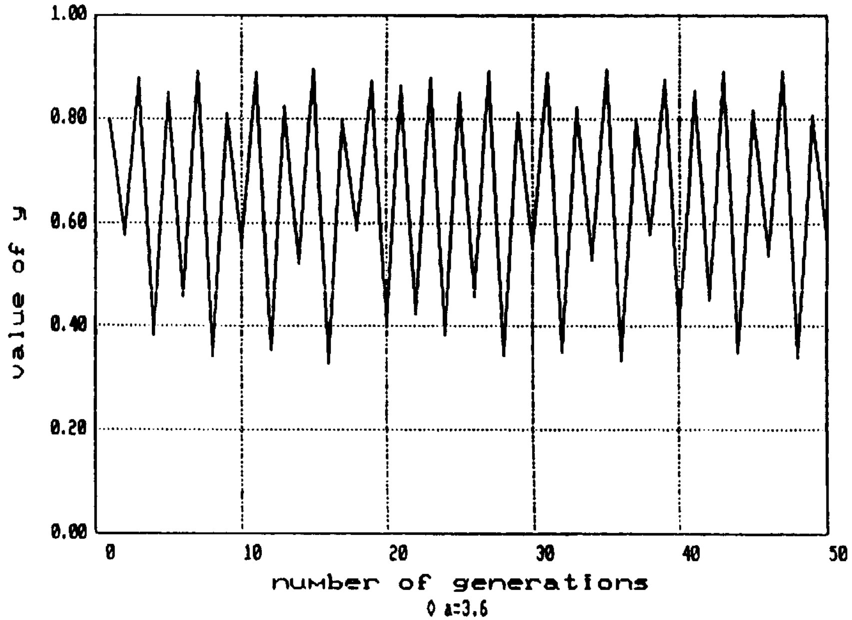


diagram 7

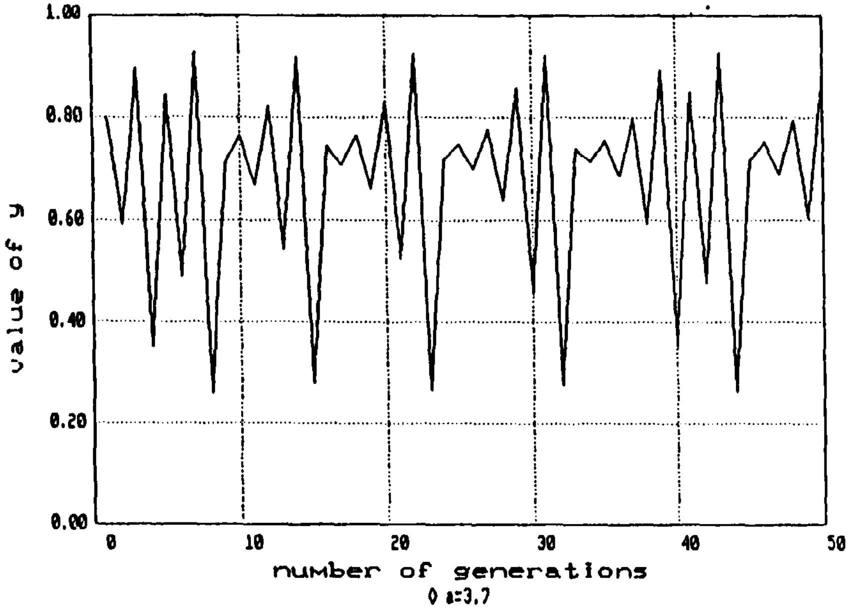


diagram 8

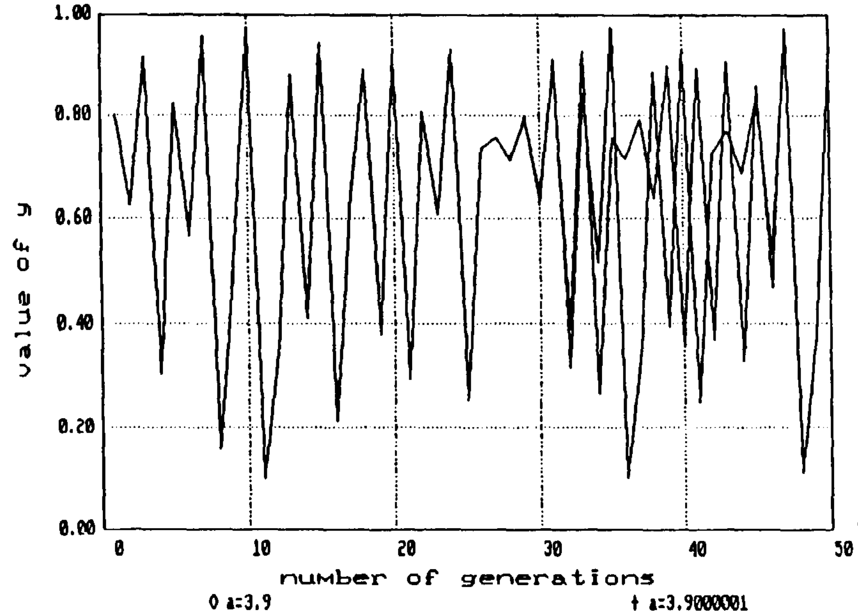


diagram 9

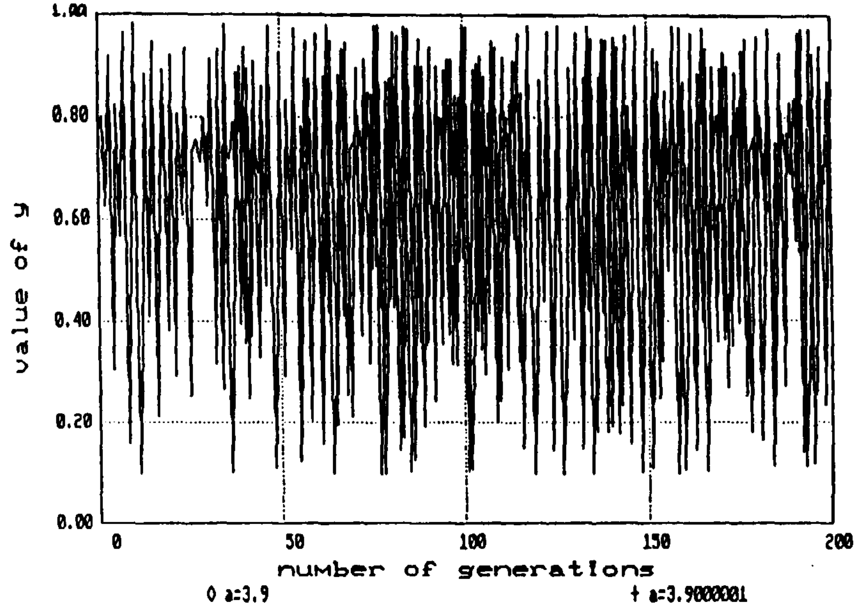
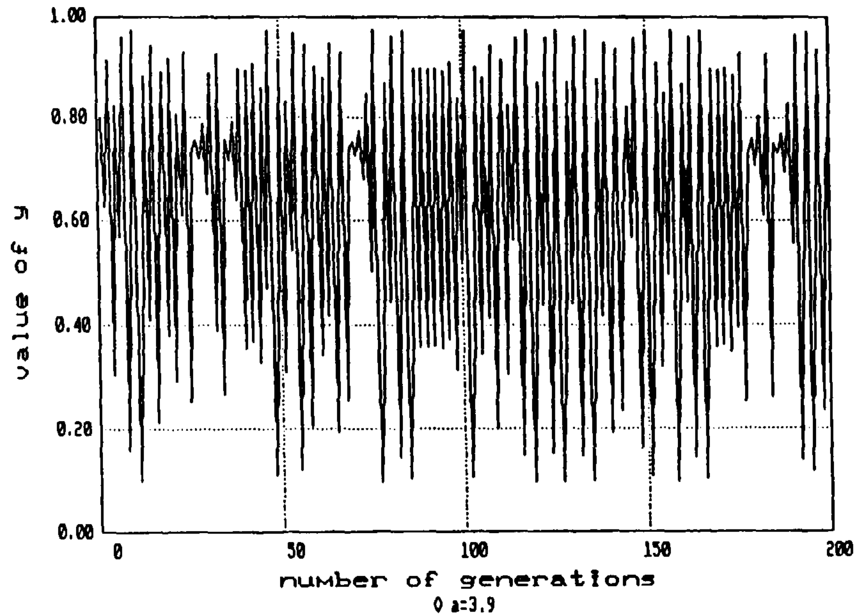


diagram 10



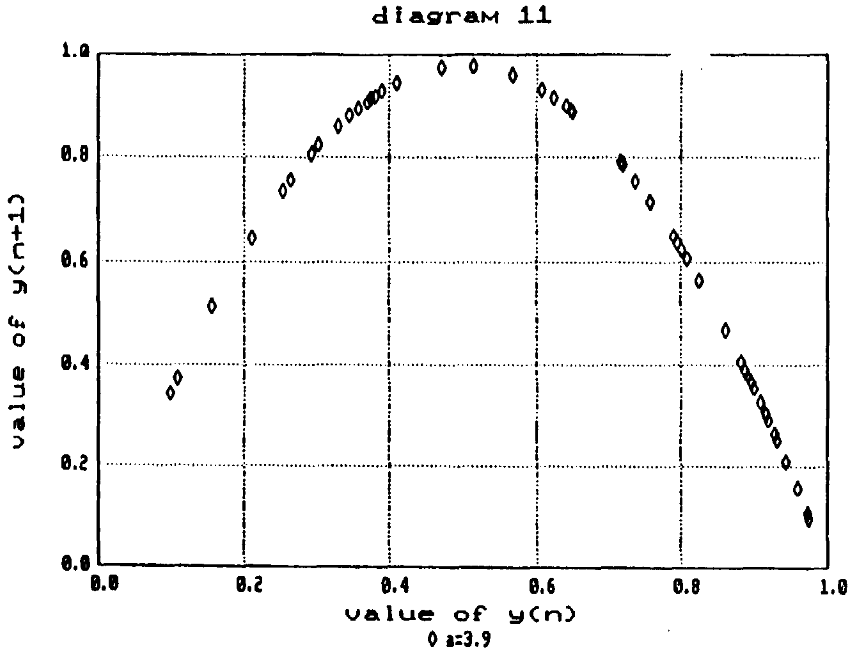


diagram 11
value of y(n+1)
value of y(n)
◊ a=3.9

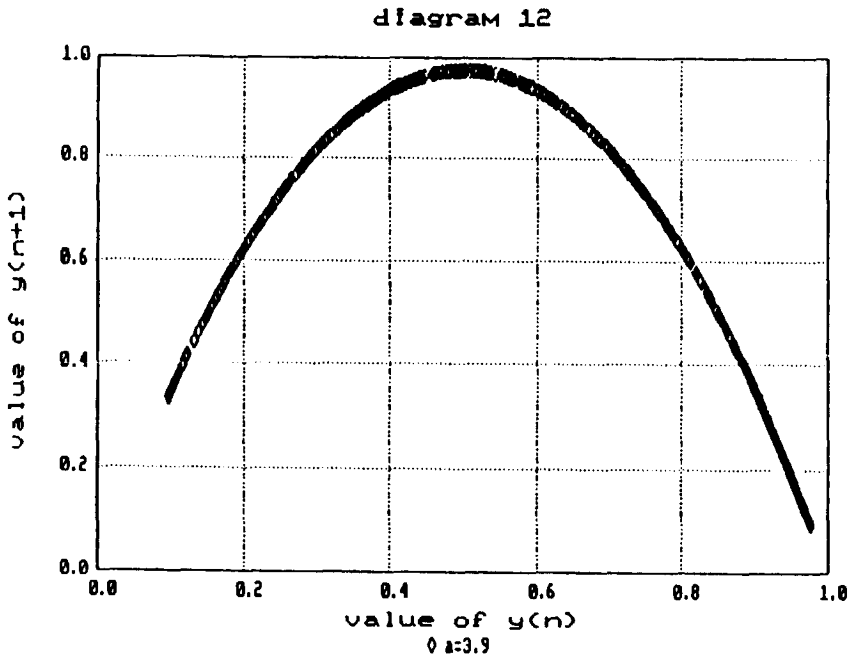


diagram 12
value of y(n+1)
value of y(n)
◊ a=3.9

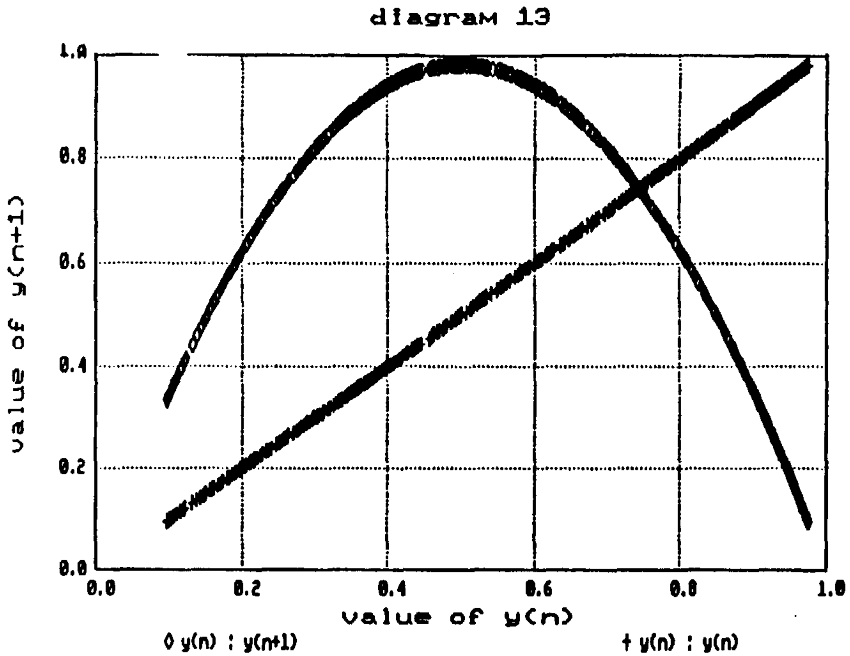


diagram 13
value of y(n+1)
value of y(n)
◊ y(n) : y(n+1)
+ y(n) : y(n)

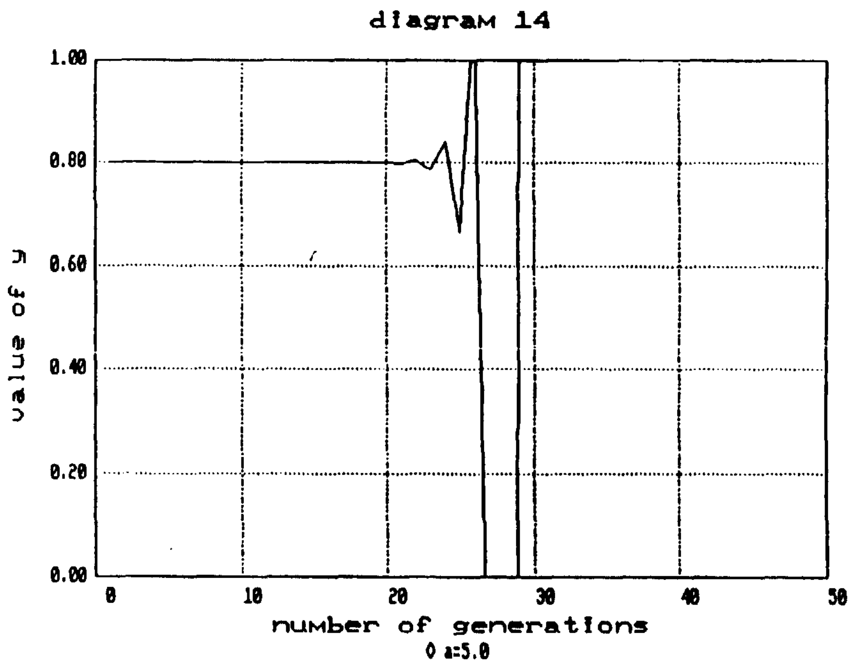


diagram 14
value of y
number of generations
◊ a=5.0

1.1 Chaos in WASSP - R. Marshall, Sheffield City Council

M. Osborne - Hydraulics Research Ltd.

Regarding your comparisons between the results produced by the spreadsheet and the simple Basic program, this is probably because the spreadsheet is working in double-precision mode (i.e. to 14 decimal places) while the Basic program is working in single precision (i.e. to 7 decimal places). All Wallingford software works in single precision. Regarding differences between machines and compilers, all should give 7 figure accuracy. However, we found tanks to be unstable in WASSP V.6 on VAX machines, and put this down to inherent inaccuracy in the hardware. Currently, we find SPIDA to be unstable on PCs, yet stable on SUN workstations. This is probably because the more complex mathematical functions are calculated in different ways.

R.M: The spreadsheet was tested on two Apricots, and COMPAQ 286 and 386 machines, yet gave identical results. Therefore I assumed that the problem was in the software. Different spreadsheets did give different results.

D. Balmforth - Sheffield City Polytechnic

We carried out experiments on WASSP V.6 using single and double precision modes of calculation. When the models were unstable, vastly differing results were obtained. I would make the following comments:

a) It is very important to assess which data has significant effect and this can only be determined by sensitivity checks.

b) There are essentially two types of instability: induced and inherent. The former is due to the algorithms used in the program, the latter due to actual physical phenomenon - it is important to understand that instability predicted by WASSP may actually exist!