CHAOS – LIMITATION OR EVEN END OF SUPPLY CHAIN MANAGEMENT

Michael Grabinski[1]

Abstract — *Proven in the early 1960s, weather forecast is not possible for an arbitrarily long period of time for principle reasons. (No supercomputer will ever be able to calculate whether it will rain in exactly 30 days or not) Supply chain management has lots of similarities to weather forecast. This paper clearly shows that supply chain management has its limitations. Starting from a certain point it is impossible to make any predictions even with the best underlying model, the biggest computer and the best software. At this point any attempts to optimize are a waste of resources. Therefore it is indispensable for every major company to check its supply chain for chaos as the phenomenon is called. This paper clearly states how such a check should be performed. In a simple example of warehouse optimization the onset of chaos is calculated mathematically.*

Keywords — *chaos, forecast, planning, SCM*

INTRODUCTION

In 1961 Edward Lorenz used a computer for weather forecast at MIT, USA. His results were puzzling. He used complex but totally deterministic numerical models. However, when changing the input variables (e.g. present temperature) very slightly (about 0.1 %) the predicted weather changed completely. Further investigations showed that the effect was not a shortcoming of Lorenz's numerical model. It turned out that every sufficiently long term weather forecast led to the same effect. Incredible small variations in parameters such as today's wind speed, temperature, or pressure have an influence on the predicted weather of say ten days ahead. Of course, it has not only an influence on the *predicted* weather. It influences the weather itself. Lorenz concluded that for long term weather prediction one has to know how the butterflies are flying today. It is known as the *butterfly wing effect*. The little eddy of a moving wing of a butterfly has a severe effect on the weather of next month. Therefore a long term weather forecast is impossible for principle reasons, because nobody knows how the butterflies will fly.

The above mentioned butterfly wing effect is due to the mathematically well known effect of chaos. It was arguably Jacques Hadamard who first wrote about the mathematically effect in the end of the 18th century. The first application to science was probably performed by Lorenz as mentioned above. Meanwhile chaos has standard applications in science. In the 1980s it became the content of standard text books (Schuster, 1984).

The connection of business and management to chaos has been established rather late (Freedman, 1992). A practical application to things like supply chain management (SCM) is even newer (Grabinski, 2004, 2007). On the other hand, the connection of chaos to management and especially SCM looks pretty obvious. The vast amount of the work in management and SCM is planning. Planning means forecasting the future business. What will be, e.g. the demand next month at a particular place of the supply chain? A supply chain manager derives it from the data of the past and maybe some fixed events of the future. In order to do so he or she uses mathematical formulas. As a simple example one may take exponential smoothening in warehousing. Though the terms and formulas are completely different from the weather forecast the principle approach is the same. Therefore one should expect the same as with weather forecast. At least long term predictions should be virtually impossible. Instead of the butterfly wing effect one might find the *forklift driver effect*: The details of the supply chain will depend heavily on the time of the breakfast brake of a particular forklift driver.

In the next section I will show the mathematical background of chaos. Besides explaining the mathematics it will show that chaos is extremely likely in SCM. Then (in the section *How to check SCM for chaos*) I will state how everyone dealing with SCM should look out for chaos. A special form of error analysis is proposed in order to detect the onset of chaos. In the last chapter (*Example of warehouse location*) a typical model for warehouse location is discussed. It is a simple example where one can prove chaos in logistics directly.

[1] Michael Grabinski, Neu-Ulm University, Wileystr. 1, 89231 Neu-Ulm, Germany; michael.grabinski@uni-neu-ulm.de

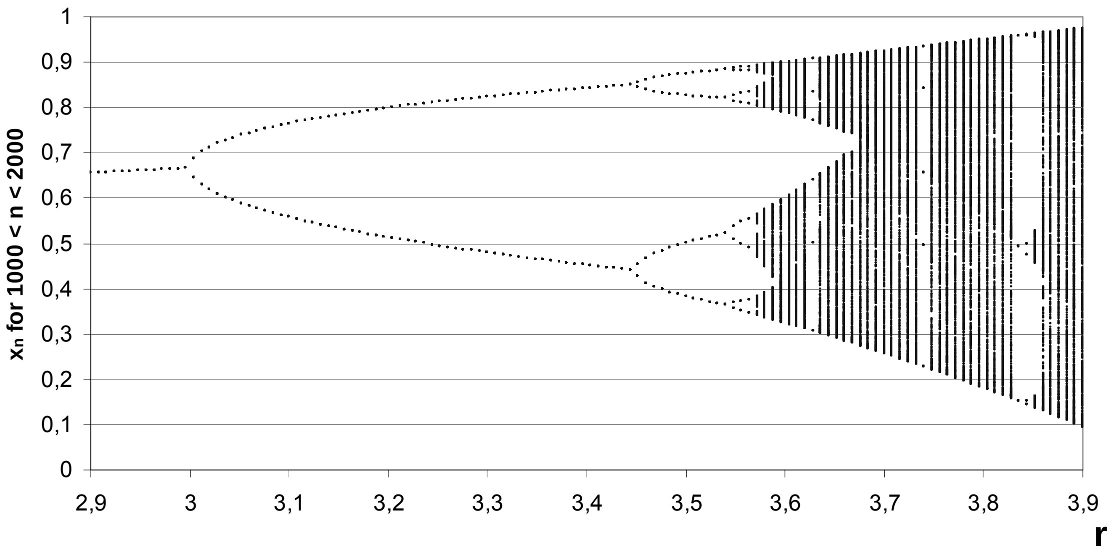
©High Speed Flow of Material, Information and Capital, Istanbul 2008
ISBN 978-605-399-070-3

MATHEMATICAL BACKGROUND

The ancient Greeks had the word Χαοσ. Originally it meant *empty space* or *void*. Later the meaning changed to *disordered mass*. Our word *gas* has its roots in the word Χαος. Gas is in fact a very disordered mass. As mentioned in the introduction the modern mathematical definition of chaos dates back to Hadamard over a hundred years ago. Rather than commenting on Hadamard's original findings I will discuss a more modern example. The logistic map (please note that it has nothing to do with logistics) is an iterative formula taking the form

$$x_{n+1} = r \cdot x_n (1 - x_n)$$

r is a parameter. Its magnitude determines whether the logistic map is chaotic or not. In order to see how it works, let us take r = 2.9 for a start. If $x_1 = 0.5$, then one will have $x_2 = 2.9 \cdot 0.5 \cdot (1 – 0.5) = 0.725$. In the same way, one will get $x_3 = 0.5781875…$, $x_4 = 0.70727147…$, and so forth. However, x_{169} is equal to just 0.65517241… All higher iterations of x (e.g., x_{1234}) will remain at this value. This phenomenon has nothing to do with chaos. Rather, 0.65517241… is a fixed point of the logistic map when r = 2.9. (Starting with $x_1 = 0.1$ instead of $x_1 = 0.5$ leads to an almost identical result, except that x_{167} already equals 0.65517241…) The first step towards chaos appears at $3 \leq r \leq 3.444$, wherein the x_i will jump between two points (For r = 3.1, for example, they are 0.558… and 0.764…). The scattering between these two points appears to be almost random. The exact way of scattering strictly depends upon the starting value of x_1. The next surprise starts at r > 3.444…. Now x_i takes four different values. In the figure below I have drawn the logistic map for $2.9 \leq r \leq 3.9$:



For r < 3 the behavior is completely non chaotic. At the end (r > 3.9) the values scatter around in a wild manner. That is the regime where chaos is completely developed. Practically speaking the result on the iteration depends on the precise initial conditions. Though the formula is deterministic in theory it leads to unpredictable results under real life conditions. Please note that objects like the logistic map are not limited to mathematicians only. In planning and forecasting, one quite often uses a formula called exponential smoothing. It takes the form

$$F_{t+1} = F_t + \alpha \cdot \left(A_t - F_t \right)$$

(F_{t+1} is the forecast for the time t +1, F_t is the old forecast, and A_t is the actual value of the last period. α is a smoothing factor.) Although the formula (map) for exponential smoothing does not show chaos, slight variations upon it may well do so. Some may object that the formula for exponential smoothing does not contain a nonlinearity, and therefore it cannot show chaotic behavior. However, extensions and alterations of the formula may contain nonlinearities.

ISBN 978-605-399-070-3

In maps such as the logistic map, proving chaos is not just done numerically like in the plot of the figure above. There are more rigorous definitions. One is the so-called Lyapunov exponent. For its definition, take an initial value x_0 and its (arbitrarily small) variation ε leading to an initial value between x_0 and $x_0 + \varepsilon$. Taking a map of the general form

$$x_{n+1} = f(x_n)$$

leads after N iterations to a value for x_N between

$$f^N(x_0) \qquad \text{and} \qquad f^N(x_0 + \varepsilon)$$

The difference of these two values may be defined as

$$f^N(x_0 + \varepsilon) - f^N(x_0) \equiv \varepsilon \cdot e^{N\lambda(x_0)}$$

where λ is at this stage just a parameter. Dividing both sides by ε and taking the limit $\varepsilon \to 0$ will create a differential quotient. Taking the logarithm of it and the limit $N \to \infty$ leads to the final definition of the Lyapunov exponent

$$\lambda(x_0) = \lim_{N \to \infty} \frac{1}{N} \log \left| \frac{df^N(x_0)}{dx_0} \right|$$

Chaos is given if $\lambda > 0$. From this definition, one sees that chaos means that an initial arbitrarily small disturbance grows exponentially with a positive exponent. This is a reasonable definition. It is in accordance with the sloppy definition given in the beginning, where I essentially said that chaos is a situation where arbitrarily small causes have big effects in the end.

To summarize I can say that chaos lures in many mathematical equations. One way to prove it is to calculate the Lyapunov exponent. (However, there are other ways (Schuster, 1984)) A necessary ingredient for chaos is a nonlinearity. (In the logistic map x_{n+1} depends on x_n^2, e.g.) Of course, not every non linear equation will lead to chaos.

HOW TO CHECK SCM FOR CHAOS

From the previous section about mathematics one may wonder what chaos has to do with SCM. However, in supply chain management one has to calculate e.g. amounts and delivery dates. In doing so many coupled mathematical equations are used. (One of them may be the formula for exponential smoothing stated above) The particular equations are normally pretty simple. They will rarely show chaotic behavior. However, long supply chains will produce a very complex array of equations. Normally they are not displayed on paper. They are hidden in complex ERP systems, sometimes coupled over several companies. Though SCM provides a complex system of equations, it is not enough to create chaos in the mathematical sense. A nonlinearity is an indispensable ingredient. Many business managers might wonder where the nonlinearities in business are coming from. Indeed quadrates and cubes are pretty rare in normal business life. However there are two sources of nonlinearities in SCM. Firstly, cost optimization deals often with nonlinear equations. Interest is an archetypal example for it. Secondly, *if-then-decisions* are nonlinearities. The first source of nonlinearities in SCM is presently considered rarely. People still try to minimize stock or reduce times in order to lower cost indirectly. However, considering costs directly is indispensable. It will be the way of the future. The second source of nonlinearities (if-then-decision) is present in abundance in SCM. Examples such like *if the delivery comes until … then we can transport by train, else we have to use a truck* are probably well known. Like here they are mostly formulated in prose rather than equations. Nevertheless they are a nonlinearity in a mathematical sense. Translated into an equation they are even non-analytic functions. Therefore the standard way of proving chaos mathematically (positive Lyapunov exponent) is not possible, because a differentiation is impossible. However other methods to prove chaos such like power spectrum analysis (Schuster, 1984) can be applied. In doing so one can show that even a single if-then-decision will lead (partly) to chaos. This is easy to see even without using sophisticated mathematics. If one takes the example above literally a delivery a nanosecond earlier or later will lead to a completely changed supply chain (transport by truck or train). In other words an arbitrarily small change in the initial conditions will lead to a macroscopic (big) effect. This is the very definition of chaos.

From this it is clear that chaos might lure in every long or complex supply chain. To prove it in a mathematical sense will be rarely possible. Because there is no standard supply chain network, it is impossible to make such statements as: "Chaos starts if x number of vendors are included in the network." Just as in big projects, the onset of chaos depends on the particular case. To probe whether chaos exists or not is, however, not so complicated as some might think. In project management or in supply chain management one normally has software that makes the calculations and predictions. Such software is analogous to a mathematical formula, except that it is in a black box. Finding numerical evidence for chaos in it is straightforward. Normally, one enters many initial data, such as delivery times or workloads, then the software produces a result for something like a finishing date or total cost. Instead of inserting single numbers for the initial conditions one should insert distributions. Typically, one may choose a Gaussian distribution for the initial conditions. Please note that all initial conditions must vary independently. All parameters such as production cost or storage capacity are considered elements of the initial condition. In doing so one can easily create over 1,000 independently varying initial conditions. Maybe one needs something bigger than an ordinary PC to perform the calculation in a reasonable time. One will end up with a final result showing some distribution too. (Instead of, e.g., a finishing date of October 31, 2007 one will have a finishing date between October 15 and November 15, 2007.) There are two possible outcomes:

- The result shows essentially the same distribution as did the initial conditions.
- Even for very small variations in the initial conditions, the result has a completely different distribution, and in the extreme case a random one.

In the first case, one has proven that no chaos is present. Besides being sure, such result is valuable for other reasons. For the width of the distribution of the initial conditions, one should take the typical margin of error in such numbers. (Please note that every number has a margin of error) Then the width of the distribution of the final result is its typical margin of error. Quite often, one finds that this margin of error is too large to allow drawing any conclusion from the final result. As an example, let us suppose the total delivery time in an industry is between 20 and 30 days. In our own company, it is 26 days. A simulation of a completely new supply chain network gives a result for a new delivery time that is between 19 and 32 days. Obviously, such information is of no use whatsoever, although the system is not chaotic.

In the case of the second bullet point above, the distribution of the final result is changed substantially. This indicates chaotic behavior. In this case, any further use of the system is utter nonsense. Here one may be able to prove chaos exactly. If the variation of the initial conditions becomes arbitrarily small and the final result nevertheless shows a random distribution (sometimes known as "white noise"), then chaos is fully developed. It is an alternative but identical definition to the definition of chaos by the Lyapunov exponent being bigger than zero.

I have just shown how to spot chaos in systems used in business. One additional important point is that if an IT system shows chaotic behavior there are two possible reasons for it:

- Reality is chaotic here.
- Only the underlying model is chaotic.

The first case is rather a hopeless one. It is a situation like the long-term weather forecast. In the second case, the reality is nonchaotic but the model used makes the reality appear chaotic. All models used in IT systems are more or less accurate descriptions of the reality, and, in this instance, it is possible to bring the model closer to reality in order to be nonchaotic. Although it is better to encounter the second case, because it at least allows the potential for a solution to be found, it can be next to impossible to distinguish between these two cases.

EXAMPLE OF WAREHOUSE LOCATIONS

And now I come to an example from the business world through which it is easy to see how chaos develops. It is the problem of warehouse location. There are two competing costs that determine warehouse locations. One is the transport cost. If it were the only cost, then there should be a warehouse (and production site) at every individual client. Transport cost would be zero in this case. However, the costs of warehouses are extremely high. They are minimized if a single, central warehouse is built. One may expect to find an optimal solution (minimum total cost) by having "some" warehouses at particular locations. Obviously, the result will depend on the cost data for transport and warehousing as well as the locations and sizes of the customers. Standard software is available to solve problems like these. In order to see how chaos develops here, I will not use such tools. I will create an admittedly oversimplified situation. Then one can solve the problem easily and see how chaos comes into play.

In my model, there are just two customers: C_1 and C_2. Their distance is d. C_1 and C_2 consume goods at the rates c_1 and c_2, respectively. There are only two possible solutions imaginable: two warehouses or one warehouse. In the case of two warehouses, each warehouse is onsite at one customer and this leads to zero transport cost. In the case of one warehouse, its location should be at the bigger customer. This minimizes transport costs (as only the smaller amount of goods must be transported). The cost of a warehouse is independent of its location. Let us assume that C_1 is the bigger customer ($c_1 > c_2$). Furthermore, the fixed cost of a warehouse would be f_W. Its variable storage cost is proportional to the consumption rates (c_1 or c_2 or both) with a specific storage cost rate α. The transport costs are proportional to the amount delivered and distance with a specific transport cost rate of β. The total costs for arrangements with one (C_{one}) or two (C_{two}) warehouses are

$$C_{one} = f_W + \alpha \cdot (c_1 + c_2) + \beta \cdot d \cdot c_2$$

$$C_{two} = 2 \cdot f_W + \alpha \cdot (c_1 + c_2)$$

If C_2 were the bigger customer ($c_1 < c_2$), then the result would read as

$$C_{one} = f_W + \alpha \cdot (c_1 + c_2) + \beta \cdot d \cdot c_1$$

$$C_{two} = 2 \cdot f_W + \alpha \cdot (c_1 + c_2)$$

The decision whether there should be one or two warehouses is based upon minimum cost. In the first case ($c_1 > c_2$), we have

$c_2 > \dfrac{f_W}{\beta \cdot d}$ for two warehouses (one at each customer)

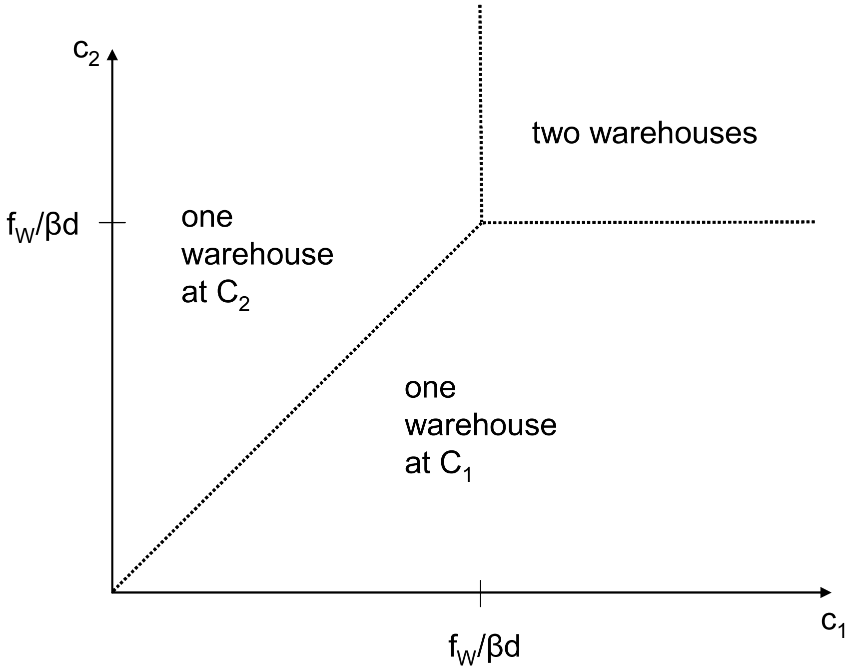
$c_2 < \dfrac{f_W}{\beta \cdot d}$ for one warehouse at C_1

In the second case ($c_1 < c_2$) we have

$c_1 > \dfrac{f_W}{\beta \cdot d}$ for two warehouses (one at each customer)

$c_1 < \dfrac{f_W}{\beta \cdot d}$ for one warehouse at C_2

The result can be displayed graphically in the figure below (Grabinski, 2007). For very big consumption rates c_1 and c_2, one should have two warehouses. For smaller consumption rates, there should be one warehouse at C_1 or C_2. As displayed in the figure below, the result depends upon the customers' consumption rates c_1 and c_2. However, if the consumption rates have values arbitrarily near to the dotted lines an arbitrarily small change in consumption may produce a big change (e.g., the move of a warehouse from its being at C_1 to C_2). So, the dotted lines denote the chaotic regime in this setup.



Of course, more realistic situations involving several clients can be easily constructed. The method of their solving will be identical. In most cases, however, only a numerical solution is possible. One may expect there to exist certain complicated areas of chaotic behavior rather than just dotted lines. Further research is necessary here. It will be an ideal playground for a PhD student.

Before closing, I want to say a few words about the software tools available for solving this class of problems. They can of course be used to spot chaos here, and one should scrutinize whether they show chaotic behavior or not. Most of them do not, however, calculate the number of warehouses. One must first enter the number of warehouses, and then the software calculates their locations. In doing so, one avoids the chaotic situation stated above and one will probably find no chaos at all.

REFERENCES

Schuster, Heinz Georg, 1984, “Deterministic chaos,” Weinheim

Freedman, D. H., 1992, “Is Management Still a Science”, Harvard Business Review

Grabinski, Michael, 2004, “Is there chaos in Management or just chaotic management?,” Complex Systems, Intelligence and Modern Technology Applications, Paris

Grabinski, Michael, 2007, “Management Methods and Tools,” Gabler