Algorithm for matrix multiplication that uses Chaos Theory

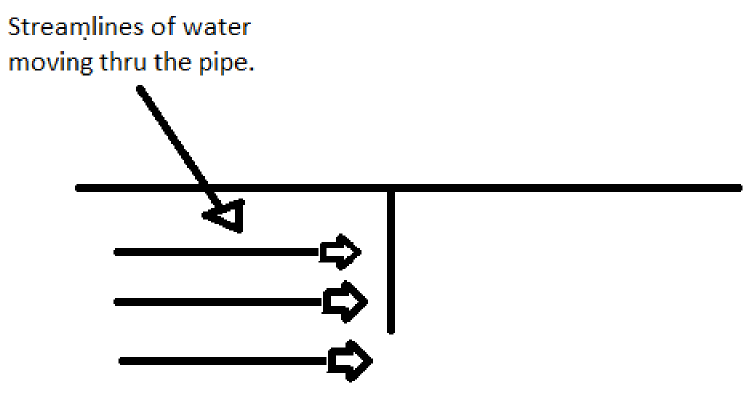
Author: PhD.Eng. Adrian Chelaru
I.N.C.A.S. Bucharest

Abstract: The purpose of this paper is to present a novel approach to the problem of matrix multiplication. This paper will attempt to prove that Chaos Theory can be use in order to obtain matrix multiplication.

Introduction: The problem of matrix multiplication has always been one of great interest for everyone who is preoccupied with solving systems of equations. From an historic perspective, matrix multiplication has appeared as a need to make a connection between the mathematics involved in solving equations and the computational power given by a computer. Without the process of matrix multiplication, a computer would not be able to solve equations in a feasible manner. A large number of methods have been developed in order to tackle the problem of matrix multiplication. The problem that arises in matrix multiplication, and the problem of upmost importance is the problem of computational time required, denoted by O [3]. For example, a computational requirement of $O(n)$, would denote an algorithm that if given n variables, would require n operations to complete. The best matrix multiplication algorithm available today can do matrix multiplication with a complexity of $O\left(n^{2.3729}\right)$.

1. Chaos Theory

In order to understand Chaos Theory, it is sufficient to imagine a flat plate placed in a pipe. The plate if fixed on one end and can rotate along a single axis.



If jets of water are thrown at the plate, it will move. When the jet hits the plate, the plate moves upwards. After the jet passes, the plate comes down to it initial position. It is true that after the jet passes, most likely the plate will exhibit some oscillations around the initial position, but because of air friction, it will eventually stabilize to its initial position. If another jet is thrown, the plate will exhibit the same motion. Now, one can ask the following question: what happens to the plate's movement if the second jet is thrown at it before it gets to stabilize? Well, in order to answer this question, another question must be answered first: how fast is the second jet thrown after the first? Or, in engineering therms, what's the periodicity of the water movement? The question can become even harder. What happens if a third jet is thrown? Or a forth one? Or an uncountable number of jets all of them being separated from each other by a certain interval of time. Well, it all comes down to the problem of time. What is the time between two separated jets? If the time is sufficiently large, the plate will exhibit a movement that is periodic. That means that it will always have the same trajectory of movement, that after a certain time, no matter how complex the movement, it will always form a pattern that will be the same. However, if the time between two individual jets becomes small enough, the plate will develop a pattern of motion that will no longer be period. The trajectories of the plate will never be the same at a predictable period of time. The movement of the plate, even though computable, will be chaotic, in the sense that it will not show a period after which the movement starts to repeat and follow the exact trajectories it did in the past. Once this has happened, the system is said to have become chaotic.

This phenomenon has first been discovered by Edward Lorenz. He was a meteorologist. While doing weather forecasts on a computer, He discovered that if the initial conditions were slightly different, the final results would show immense differences.

In the case of the plate, after the point in which the jets of water have a movement period just right in order to make the plate move in a chaotic fashion, all variations that affect the period of the jets, will produce completely different trajectories in the plate's movement. This observation, that really small variations of the initial conditions of the system produce immense variations in the final results represent the basis of Chaos Theory.

2. Computers and Chaos Theory

This property of systems, that in certain given conditions They become chaotic, has been observed in all human known systems. In most application, this effect has to be avoided, because it produces undesired effect. For example, in the case of the flat plate, if this were to happen and the movement of the plate would become chaotic, the chances of the mount breaking would be really high.

This chaotic behavior has been observed in computational cases also. For, a well known study case is given by the following formula:

$x_{n+1} = r x_n (1 - x_n)$ [2]

As simple as the formula might seem, if the parameter r goes beyond the value of $r >$ 3.57 the computation of x at every time step given by the value n will give values that seem completely random. When computing such values, one would expect them to have a rather predictable variation, that would arrange themselves in a pattern that is somehow easy to approximate without having to compute the entire function step by step. But that is not the case. The values of the functions will vary so greatly, that the only way to predict the desired number of values given by the function is to actually compute all the values.

For example, if one were to run the following fortran code:

```
program matrici

implicit none
real*8 x(10), r
integer i
open(1, file='rez.txt')
x(1) = 0.1
r = 3.572
do i= 2,9
x(i) = r*x(i-1)*(1-x(i-1))
write(1,*)x(i)
enddo
```

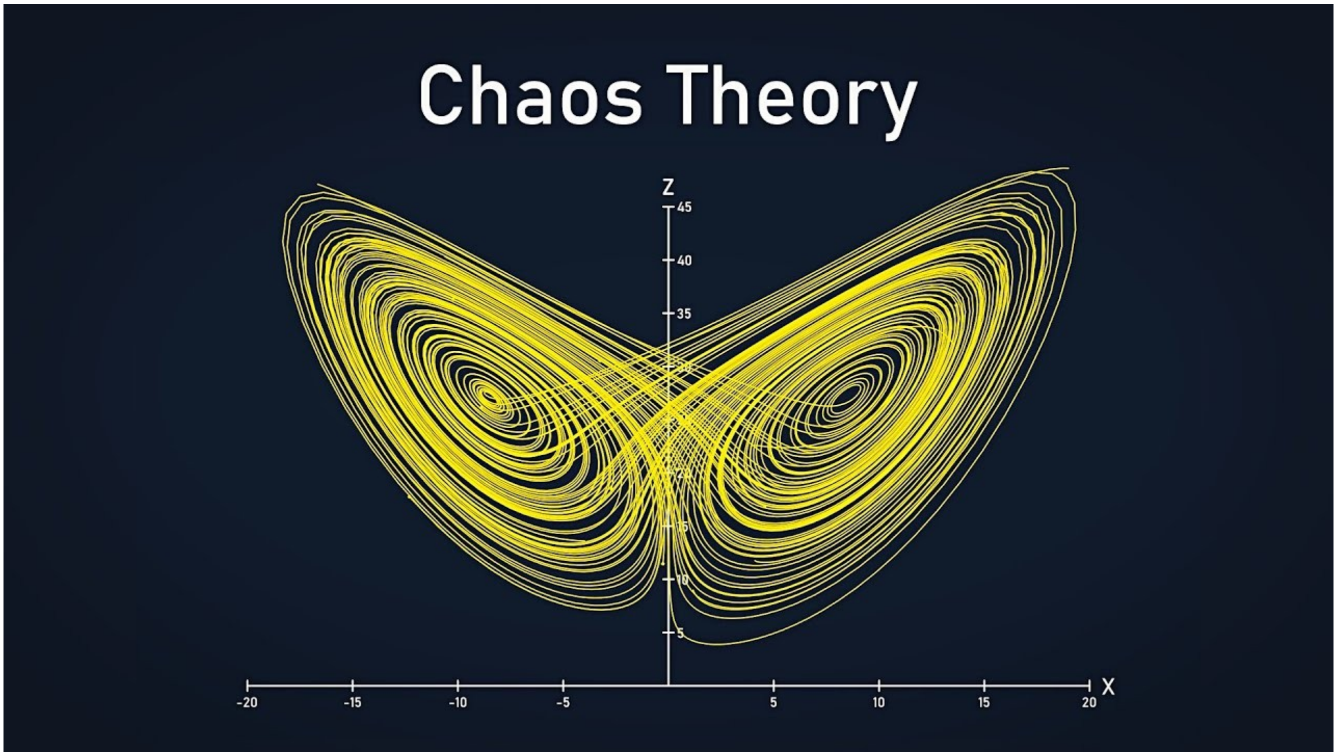
end program matrici

the results would look like this:

0.32148000666141513
0.77916255181153049
0.61462786384193679
0.84606554383731136
0.46521242330233581
0.88867725978420975
0.35337791880126634
0.81620902617094904

If one looks at the values, the results seem to be moving around a certain position, but not actually having a clear orientation.

Traditionally, when describing Chaos Theory, one would encounter the following picture [1]:



This picture shows the main characteristic Chaos Theory. When plotting the values that result from a chaotic systems, these values seem to arrange themselves around a specific point called attractor. In the picture shown, the system has two attractors around which the solutions of the system revolve around, without stabilizing themselves into the actual attractor.

In the results given by the computational function, such an attractor exists. If one were to compute multiple values for the function, the attractor can be aproximated. The difference between the function given above and the picture shown as a traditional example, consists in the fact that in the case of the function, the system evolves in 1D, unlike the picture presented that is in 2D. So, in the case of the function, the values would position themselves on a line, which wouldn’t give for much of a visual interpretation of the results. If the results of the function are plotted in 1D, one would only see many points really closed to one another and the idea of an attractor would be hard to grasp, unlike when one visualizes the results for a 2D system, that moves in the coordonates XZ, as shown in picture above.

3. Matrix multiplication using Chaos Theory

When one thinks of algorithms designed to multiply matrices, the idea of algorithm stabilization comes to mind. Since all practical algorithms used to multiply matrices are based on approximations, in order to simplify the task as much as possible, errors appear. These errors tend to add up so one must find ways to prevent this.

Unlike those algorithms, the algorithm presented in this paper does not engage into the problem of stabilization. Instead, it takes the exact opposite approach. Instead of trying to approximate the solution and stabilize it, it tries to make the solution chaotic, in order to make the values that result from the algorithm align themselves on both sides of the attractor of the given chaotic system that emerges. This method assumes that the attractor of the resulting system will be the exact solution of the process of matrix multiplication. This assumption is made on the premises that matrix multiplication is a dynamic system, so if one is dealing with a chaotic dynamic system, the attractor represents the closest thing to a stability point that one can imagine. If the dynamic system of matrix multiplication were to be exact, then basically the entire system would converge to the exact solution. Think about the plate example. After the jet passes, if the system is not chaotic, the plate returns to the initial position. The initial place represents the convergence of the system. If the system is chaotic, the plate no longer returns to the initial place, but tends to move around the attractor, that somehow represents the closest thing one can approximate to a point in which the system would become stable. In a sense, if the system could reach that point, it would have converged to the solution. But since it can't, it moves around it. The same analogy can be made in the case of matrix multiplication. The chaotic system of matrix multiplication will have an attractor, that should represent the exact solution of the matrix multiplication.

Let's take two very simple matrices as an example.

$A \cdot B = C$, all being 2x2 matrices, that we detail as:

$$\begin{bmatrix} a_1 a_2 \\ a_3 a_4 \end{bmatrix} \cdot \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix}$$

In this case, the exact value of c_1 would be:

$$c_1 = a_1 b_1 + a_2 b_3$$

For simplicity, we will only deal with the problem of determining the value of c_1.

As one may know, not all matrices are created equally. For example, if the value of b_1 is much greater than the values of all other numbers that form the matrix B, then all other values except for b_1 can be neglected in the process of matrix multiplication. In practice, many matrices that are of interest for computations are form of band matrices, that only have values on the main diagonal, the rest of the matrix being formed of zeros. This is also a desired thing for many algorithms that do matrix multiplication. If a matrix can be diagonalized then the computation requirements of the matrix multiplication algorithm drop. This is the case for example in all CFD solvers.

Unfortunately, for a random case that does not result from a specific physical process, most matrix are not band matrices.

Let's now try and determine a way thru which the process of matrix multiplication can be made to fit Chaos Theory.

One approach would be this:

Can We say that the matrix A, multiplied by the number b_1 is an approximation to the matrix multiplication process? Well, if b_1 were to be much bigger than all other values that make up the matrix B, then one could say that. The question then becomes, is b_1 bigger than all other values that make up B? There is no reason to assume that it is. But the main question is: Can We say that $A \cdot b_1$ is an approximation of $A \cdot B$? Well… sure, why not? It is an approximation. A very bad approximation, but still an approximation.

What about $A \cdot b_2$? Is that also an approximation to $A \cdot B$? Yes, it is.

Does this apply to all other values from the matrix B? Of course!

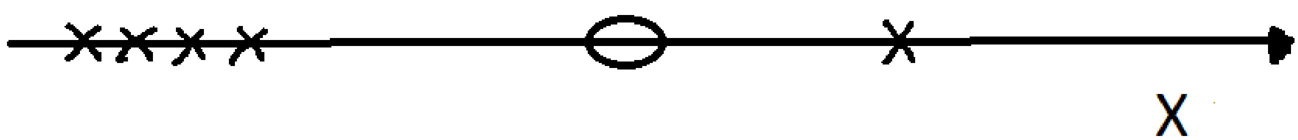
And now the question becomes:

If we do such approximations of the matrix multiplication of $A \cdot B$ and also make sure to make this process chaotic, will these approximations line up around an attractor?

Well… if the system is chaotic and these are indeed approximations of the $A \cdot B$ matrix multiplication, then the answer must be yes. The approximations, as bad as They might be, will line up around the attractor.

So, how do we determine the solution of the matrix multiplication?

Let's look at a diagram that represents this chaotic process.



As one can see, if the process is chaotic, the theory says that the values resulting in the process will be found on both sides of the attractor (because this is a 1D case, such as in the function previously presented). And if the attractor is the exact solution of the matrix multiplication, then once one finds solutions on both sides of the attractor, one can simply apply the interval splitting algorithm in order to determine the value of the attractor, and thus the value of the matrix multiplication process.

The author of this article has done such an algorithm in practice. What He has managed to observe was that for the cases He tested, the values of the function once it goes on the other side of the attractor are very high in comparison to the previous value. The algorithm will not be presented in this paper, since it requires additional work in order to determine the tweaks needed to be done in order to make it robust enough for real life computational cases.

One observation has to be made. The algorithm that was previously described doesn't actually take into account the real magnitude of the matrices A and B. Nor should it. It is unimportant how big the matrices are. So, in theory, even for matrices that have billions of elements, the complexity of determining the final result should not be larger than $m \cdot O(n)$, where m is a number given by the number of operations required to determine each value from the matrix C. In order to compute this m, one must add up the operations required by the approximations made and the operations required by the splitting of intervals. For trivial cases like the one presented, where A and B are 2 by 2 matrices, this algorithm is slower that the simple Gauss method. But for matrices that are billions times billions it should in theory be much much faster, since the computational time is linear.

Conclusions:

The paper presents an algorithm based on Chaos Theory that could theoretically optimize all scientific processes that involve matrix multiplication. Additional work must be undertaken in order to assure the robustness of the algorithm.

References:

1. https://en.wikipedia.org/wiki/Chaos_theory
2. https://en.wikipedia.org/wiki/Chaos_computing
3. https://en.wikipedia.org/wiki/Computational_complexity_of_matrix_multiplication