



# ChaosKit White Paper

Andrew N. Edmonds  
Scientio, LLC.

Friday, 16 September 2005  
Version 1

<http://www.scientio.com>  
support@scientio.com

Introduction.....1  
Temporal database.....2  
Financial time series features .....2  
Dead period handling.....2  
Embedding.....2  
Measures .....3  
Prediction.....3  
Derivation.....4

## Introduction

Chaos kit is a web class library, available in Microsoft .Net or Java versions, which permit you to add chaotic time series analysis and prediction to your systems.

Chaotic processes crop up in a large number of domains, such as medicine, biotech, mechanical engineering and finance. Whenever a system has a feedback path it is possible for chaos to arise.

Chaotic systems are ultimately unpredictable: the presence of feedback means that the number of bits required to calculate some future state will increase at least linearly with increasing time, and that the uncertainty associated with the prediction will also grow at least linearly.

The fact that you cannot determine the state of such systems an arbitrary period ahead does not mean that you can't generate practically useful predictions for some way into the future. It very much depends on the application the system and the requirements whether you can make predictions that are useful.

ChaosKit provides tools for evaluating your data to determine the degree of chaos and the limits of predictability, and for generating predictions.

To limit your risk, ChaosKit is available in developer and commercial versions. The former is much cheaper than the latter, and the user can determine if practically useful results can be generated for their particular problem with minimal expenditure, before buying a commercial license for a full application. Alternatively, depending on workload, Scientio can often help with the analysis at an hourly rate.

## **Temporal database**

At the heart of ChaosKit is a temporal database. This allows the user to enter time-tagged data values in a variety of ways. Chaos kit then acts on these stored values to calculate various measures of chaos and to generate an embedding specification.

Data values can be unordered, and ChaosKit does not require that they be regular samples in time. So for instance ChaosKit can handle financial ‘tick’ data, representing transactions and quotes that occur at arbitrary moments in time. The user can then set a sample time that they wish to use for the analysis, and ChaosKit will take care of sampling the underlying data set to create a sampled time series to use for processing. ChaosKit can optionally interpolate between stored data points where necessary.

## **Financial time series features**

Financial time series are not continuous, since markets close daily or at weekends. Nor are the data values produced truly real numbers, since all are quantised in some way or other. ChaosKit enables the user to specify trading times and the minimum quantum in which they are traded. Non trading periods are removed from the embedding process, and predictions are rounded to the nearest quantum.

## **Dead period handling**

Some time series contain dead periods, where for some reason data was not collected. The user can set a time span which is used as a dead period detector. As the internal database is scanned in the embedding process, if any period between stored data points exceed this dead period, the embedding process is stopped and restarted once the scanning point has passed the next stored point.

## **Embedding**

Embedding is the process of converting a time series into a multidimensional space through sampling. Regularities that are completely hidden in the time series, and that are not amenable to conventional analysis, such as Fourier transforms, become apparent after embedding.

The critical issue is to determine the optimum dimension of the embedding, and the optimum separation of the samples in time. At some critical dimension the internal dynamics of the series are revealed, and adding further dimensions does not add any more information. The addition of unnecessary dimensions complicates processing and predictions however, and thus there is a trade off that makes one choice of dimension optimal.

ChaosKit uses a geometric technique known as False Nearest Neighbours to determine the optimal embedding dimension, and a prediction based technique to determine the optimal sample separation.

## Measures

Once the data has been correctly embedded the Lyapunov exponent can be measured. This is a measure of predictability, based on information theory, which supplies the average number of bits of information lost each sample step away from a given point in time.

Two other measures are created that do not rely on embedding; these are the fractal dimension of the time series, and the Hurst exponent. The former, a value between 1 and 2, describes how the time series “fills the space” available to it. A time series that seldom changes or does so slowly has a fractal dimension tending to 1, whereas a very active time series tends to 2.

The Hurst exponent is a standard indicator of random behaviour. A Hurst exponent of 0.5 is associated with random behaviour, whereas values tending towards zero demonstrate so called ‘anti persistent’ behaviour of the series, and values tending towards 1.0 demonstrate persistent behaviour.

## Prediction

Assuming the results of the analysis are favourable you may want to generate predictions.

ChaosKit uses the calculated embedding specification to create an internal memory-based model of the time series. This model can be limited to use only the most recent data by setting an “attention span” parameter. Data older than this period will be ignored.

Predictions are generated in multiples of one sampling step after the last data item in the data base. It is totally permissible to interleave prediction and adding new data items to the database, so that the model continually updates in real time.

There are a variety of methods for generating models of the internal data, from Neural nets, to fuzzy rules. At Scientio we are experts in computational intelligence and have, as you can see from our web site, our own Fuzzy logic data mining system, XML Miner. Our experience, however, is that for this particular application a memory based system is best. With such a system new data items can be immediately added to the model without substantial retraining, and such models are effectively parameterless, making them simple and reliable to use.

## **Derivation**

ChaosKit is derived from the algorithms in and experienced gained in creating the Thesis "Time series prediction using Supervised Learning and tools from Chaos Theory" by the author. <http://www.scientio.com/resources/thesis.pdf> However it consists of completely new code and has the benefit of many improvements in techniques and technology since the original version. Several of the key algorithms have been modified for this version of software.

The author was contacted some months ago by the representative of a hedge fund who had built a version of the software in the thesis, despite ChaosKit being on the market. This is rather bizarre behaviour given that ChaosKit is much improved and false economy.

Using some standard metrics (COCOMO II), ChaosKit represents 12 man months of work, assuming that the algorithms were available. This would cost at least \$41,000 to commission. ChaosKit therefore represents very good value for money.