

Recurrence Plots in Nonlinear Time Series Analysis: Free Software

Jorge Belaire-Franch*
Dulce Contreras

*Dept. of Economic Analysis
University of Valencia*

Abstract

Recurrence plots are graphical devices specially suited to detect hidden dynamical patterns and nonlinearities in data. However, there are few programs available to apply such a methodology. This paper reviews one of the best free programs to apply nonlinear time series analysis: Visual Recurrence Analysis (VRA). This program is targeted to recurrence analysis and the so-called Recurrence Quantitative Analysis (RQA, the quantitative counterpart of recurrence plots), although it includes many procedures in a friendly visual environment. Comparisons with alternative programs are performed.

1 Introduction

The fast development of computer resources available to the scientist community and the parallel growing bulk of theoretical knowledge about complex dynamics have allowed many researches to look for non-linear dynamics in data whose evolution linear ARMA models are unable to explain in a satisfactory manner. This approach arose in natural sciences (Physics, Biology,...) but quickly was adopted in Economics (e.g., Brock et al., 1988).

*Corresponding author. Department of Economic Analysis. University of Valencia. Campus dels Tarongers, Avda. dels Tarongers s/n. 46022 Valencia (Spain). Phone: 34-96-3828246, fax: 34-96-3828249, e-mail: jorge.belaire@uv.es

In recent years, the attempts of systematising the (apparently) disperse set of techniques have led to the so called Nonlinear Time Series Analysis (NLTSA, hereafter; see Kantz and Schreiber, 1997). Nevertheless, and maybe due to such a dispersion, the methods involved have not been incorporated yet into standard econometric packages, or they have been embodied very partially into more general environments (e.g., Xplore).

Those methods can be classified into metric, dynamical, and topological tools. The metric approach depends on the computation of distances on the system's attractor, and it includes Grassberger-Procaccia correlation dimension. The dynamical approach deals with computing the way nearby orbits diverge by means of estimating Lyapunov exponents. Topological methods are characterised by the study of the organisation of the strange attractor, and they include close returns plots and recurrence plots.

As an aftermath of dispersion of these methods, very few programs applying NLTSA techniques are available. In our opinion, DATAPLORE is the most complete commercial software designed to that end (more information available at <http://www.datan.de/dataplore>. But it is not the only one: Chaos Data Analyzer (<http://www.sprott.physics.wisc.edu/cda.htm>) or cspW (or the Lynux version of the program, cspX, both available at <http://www.zweb.com/apnonlin/csp.html>) are remarkable competitors.

Nevertheless, the World Wide Web provides several free alternatives to academic, non-linear oriented researchers: TISEAN¹; Nonlinear Dynamics Toolbox, NDT²; Recurrence Quantification Analysis, RQA³; or Visual Recurrence Analysis, VRA⁴, just to mention a few. All of them provide a wide range of NLTSA tools.

This paper reviews one of the best free programs available to apply non linear time series analysis, and more specifically, recurrence analysis: VRA. To that end, recurrence analysis will be reviewed in the next section; section 3 briefly compares the free programs TISEAN, NDT, VRA and RQA. Section 4 focuses on VRA, and section 5 concludes.

¹<http://www.mpipks-dresden.mpg.de/tisean>

²<http://www.physics.gatech.edu/chaos/research/NDT.html>

³<http://www.rushu.rush.edu/molbio/physiozbi.html>

⁴<http://www.pweb.netcom/eugenek/download.html>

2 Recurrence analysis

Recurrence analysis is a graphical method designed to locate hidden recurring patterns, nonstationarity and structural changes, introduced in Eckmann et al. (1987).

Suppose that information is available on a univariate time series which is part of a larger n -dimensional (maybe deterministic) model. Takens (1981) theorem shows that we can recreate a topologically equivalent picture of the original multidimensional system behaviour by using the time series of a single observable variable, by means of the method of time delays: for the scalar series x_i , we construct the embedded vectors

$$x_i^m = (x_i, x_{i+d}, x_{i+2d}, \dots, x_{i+(m-1)d})$$

where m is the embedding dimension and d is the time delay. Thus, if $m \geq 2n + 1$, a single output variable is sufficient to recreate completely the dynamics of the underlying system. However, the sequence of embedded vectors is useful only if parameters m and d are properly chosen by using appropriate methods.

Next, a symmetric matrix of distances (e.g., Euclidean distances) can be constructed by computing distances between all pairs of embedded vectors; the recurrence plot relates each distance of such a matrix to a colour (e.g., the larger is the distance, the “cooler” is the colour). Thus, the recurrence plot is a solid rectangular plot consisting of pixels whose colours correspond to the magnitude of data values in a two-dimensional array and whose coordinates correspond to the locations of the data values in the array.

It is also quite usual to establish a critical radius, ε , and to plot a point as a darkened pixel only if the corresponding distance is below or equal to ε (in fact, papers used to be published in black-and-white journals). Vectors compared with themselves necessarily compute to distances of zero, which explains the presence of the strong upward diagonal (line of identity) in all recurrence plots.

If the analysed time series is deterministic, then the recurrence plot shows short line segments parallel to the main diagonal which correspond to sequences $(i, j), (i + 1, j + 1), \dots, (i + k, j + k)$, such that $x_j^m, x_{j+1}^m, \dots, x_{j+k}^m$ are close to $x_i^m, x_{i+1}^m, \dots, x_{i+k}^m$. On the other hand, if the series is white noise, then the recurrence plot does not show any structure at all.

As an illustration, we generate 1000 observations of the (chaotic) Rossler

system, defined by the following equations:

$$\begin{aligned}\dot{x} &= -(y + z) \\ \dot{y} &= x + 0.2y \\ \dot{z} &= 0.2 - 4z + xz\end{aligned}$$

Figure 1 shows the recurrence plot from the Rossler x variable, for a time delay $d = 14$ (selected through the method of average mutual information) and an embedding dimension $m = 4$ (selected through the false nearest neighbours method). Figure 2 shows the recurrence plot from a Gaussian white noise, for $d = 1$ and $m = 12$. Both figures have been attained using RATS graphics; computations were performed using programs written by the authors in Ox 2.20 (see Doornik, 1997).

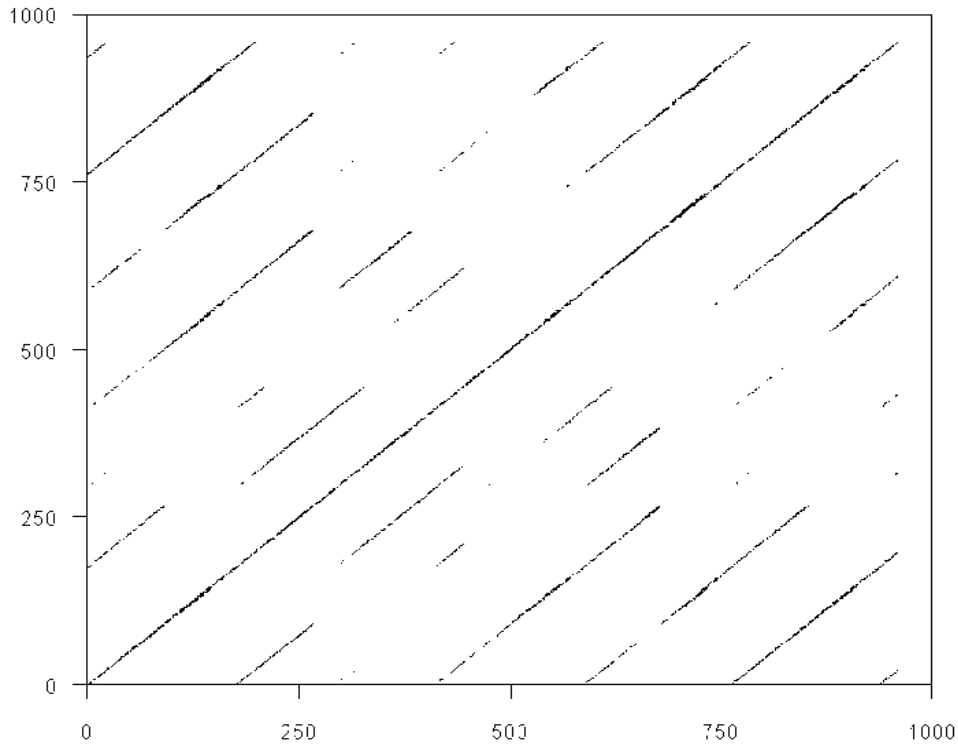


Figure 1: Recurrence plot from Rossler series

Recurrent points in Figure 1 for the Rossler x variable form distinct short diagonals parallel to, but offset from, the main diagonal. Alternatively, in

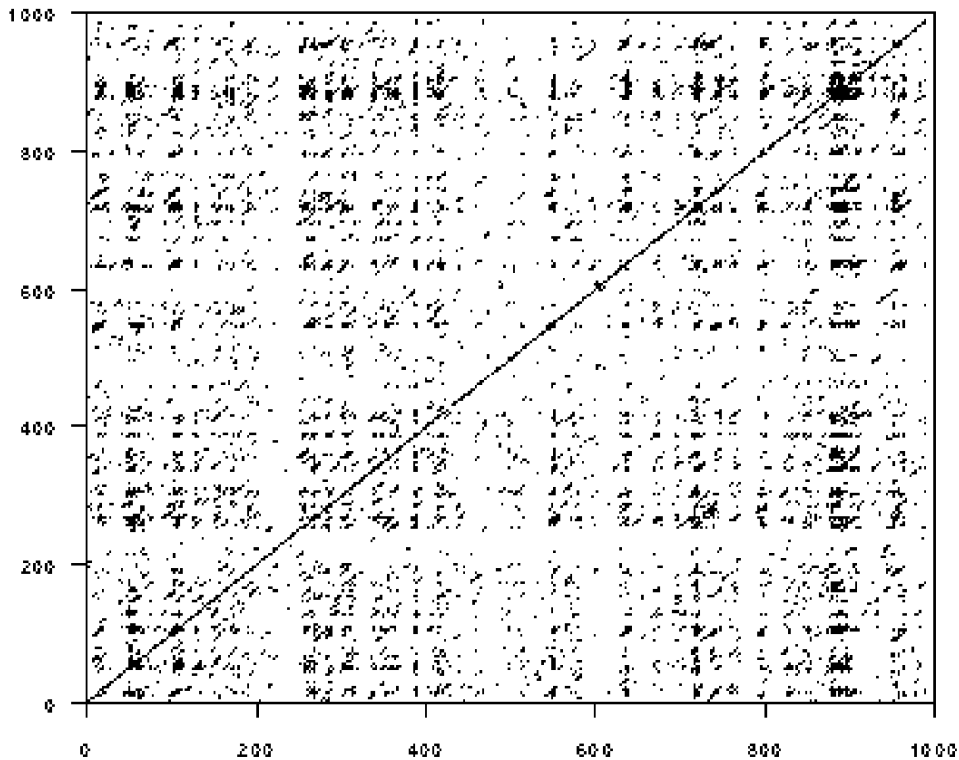


Figure 2: Recurrence plot from Gaussian noise

Figure 2, recurrence points for the white noise are simply distributed in a homogeneous random pattern, signifying that a random variable lacks of deterministic structures. The upward diagonal lines in Figure 1 result from strings of vector patterns repeating themselves multiple times down the dynamics. This type of recurrent structure indicates that the dynamics is visiting the same region of an attractor at different times; therefore, the presence of diagonal lines indicates that deterministic rules are present in the dynamics.

The set of lines parallel to the main diagonal is the signature of determinism. That set, however, might not be so clear (e.g., the size of the lines being relatively short among a field of scattered recurrent points), i.e., the recurrence plot could contain subtle patterns not easily ascertained by visual inspection; in this context, Zbilut and Webber (1992) propose the so called recurrence quantification analysis (RQA). They define the following measures for diagonal segments, in order to emphasize different features of the plot:

%recurrence, %determinism, averaged length of diagonal structures, entropy and trend (see Zbilut and Webber, 1992, for more details). In addition, Webber and Zbilut (1998) introduce the concept of cross recurrence plot by which the dynamical behaviour of two time series is compared.

3 Recurrence analysis tools

All the free programs cited at the introduction apply recurrence analysis, though not all of them apply recurrence quantification analysis: that is the case for TISEAN and NDT. However, all the recurrence techniques can be implemented through RQA and VRA, although just RQA enables to using cross-recurrence quantification.

On the one hand, TISEAN, VRA and NDT are programs for general non-linear analysis. TISEAN is a set of DOS executables (though the source code is available), as a part of the so-called TISEAN project; the set of techniques is very large, but definitely it is not user-friendly. VRA and NDT work under Windows, and they are very easy to use general non-linear analysis programs; however, the set of procedures is smaller than TISEAN, and more importantly, they do not allow to vary all the control parameters of the different techniques involved. TISEAN does it, but at the cost of a greater complexity.

On the other hand, RQA, exclusively designed to apply recurrence quantification analysis, is a set of DOS executables that are very easy to use: in this case, control on the parameters of the procedures is absolute.

As it was shown above, recurrence plots are graphical devices; therefore, software should provide suitable high resolution graphical representations in order to be published somewhere. TISEAN produces a file with the values to be scattered, depending on the computed distances; thus, the file can be read into software which allows plotting of high quality graphics (e.g., RATS, Figure 3). However, it can only store the values of one half of the recurrence plot, although this is not really a serious drawback due to the fact that the representation above and below the main diagonal is symmetric. RQA output appears on the DOS screen, and it can be hardcopy-captured by any standard screen-dump utility (Figure 4), though variables to be scattered can be stored on separate files, and exported to a graphical program. NDT allows storing recurrence plots as bitmap (extension .bmp) files, but they do not look like figures journals use to publish (Figure 5). Finally, VRA

recurrence plots (Figure 6) look very much like NDT plots, and they can also be saved as bitmap files.

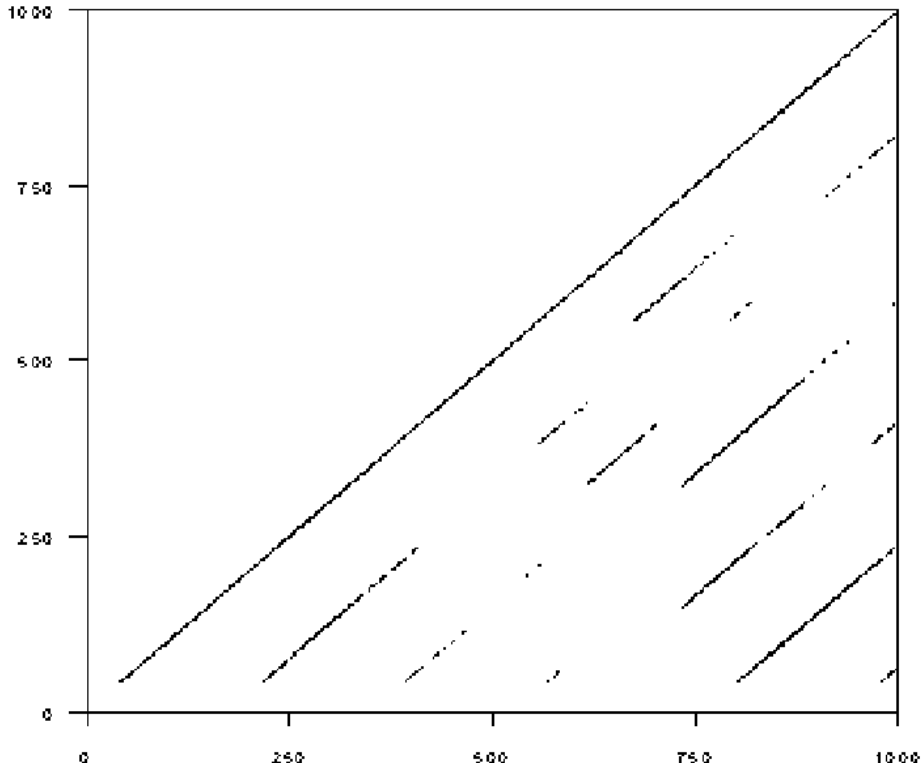


Figure 3: RATS graphic from TISEAN archive

It is not the aim of this work to compare exhaustively the mentioned programs; nevertheless, Table 1 and Table 2 summarize their main characteristics.⁵ From that comparison, it appears that VRA is one of the most complete non-commercial non-linear time series analysis software packages, and one of the easier (maybe the easiest) to use; that is why we will focus on this program in the next section.

⁵Speed was computed as CPU time on a P-II (233 MHz) to obtain a recurrence plot from a data set of size 2000 observations, $m = 5$, $d = 1$ and $\varepsilon = 0.5$.

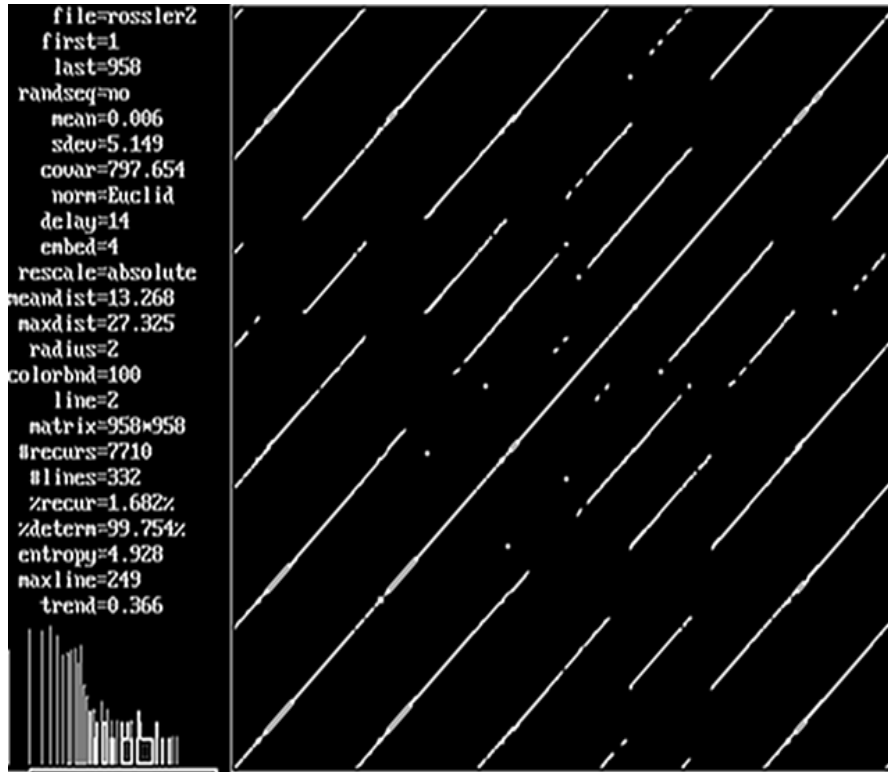


Figure 4: Graphical output from RQA DOS window

Table 1: Software comparison

Software	Save graphics	Ease of use	Control	Set of techniques	Html help system
TISEAN	No	*	****	****	Yes
NDT	Yes	****	*	***	Yes
RQA	No	***	****	*	No
VRA	Yes	****	***	****	Yes

****: excellent. ***: very good. **: good. *: weak

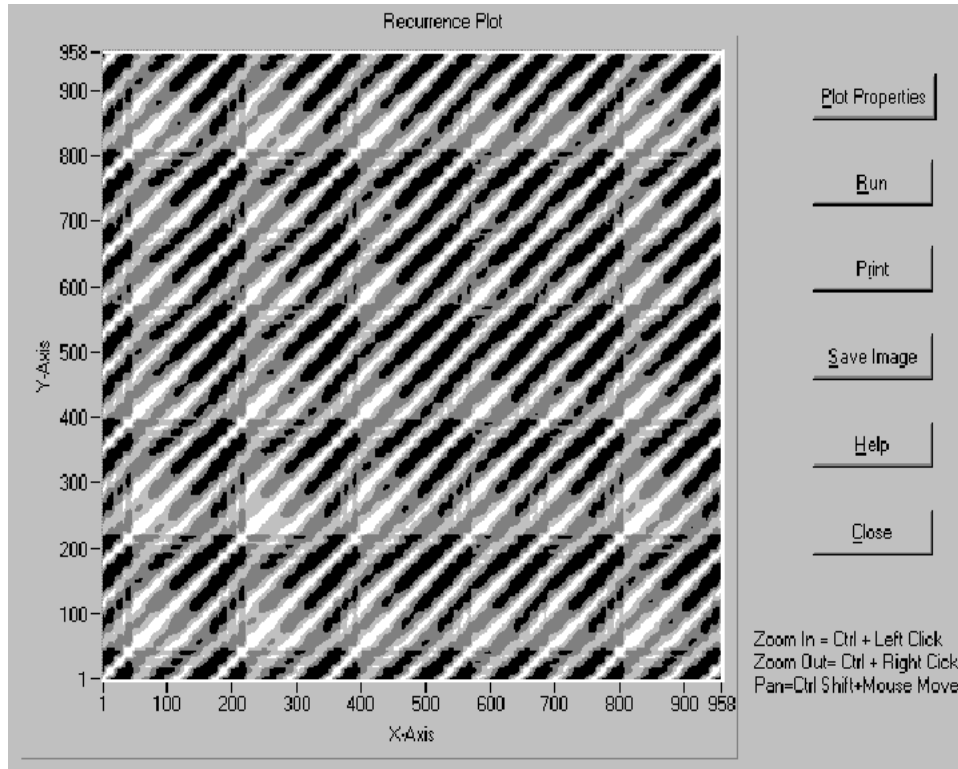


Figure 5: Graphical output from NDT

Table 2: Speed comparison

Software	Speed
TISEAN	< 2 secs.
NDT	> 2 mins.
RQA	> 2 mins.
VRA	< 1 sec.

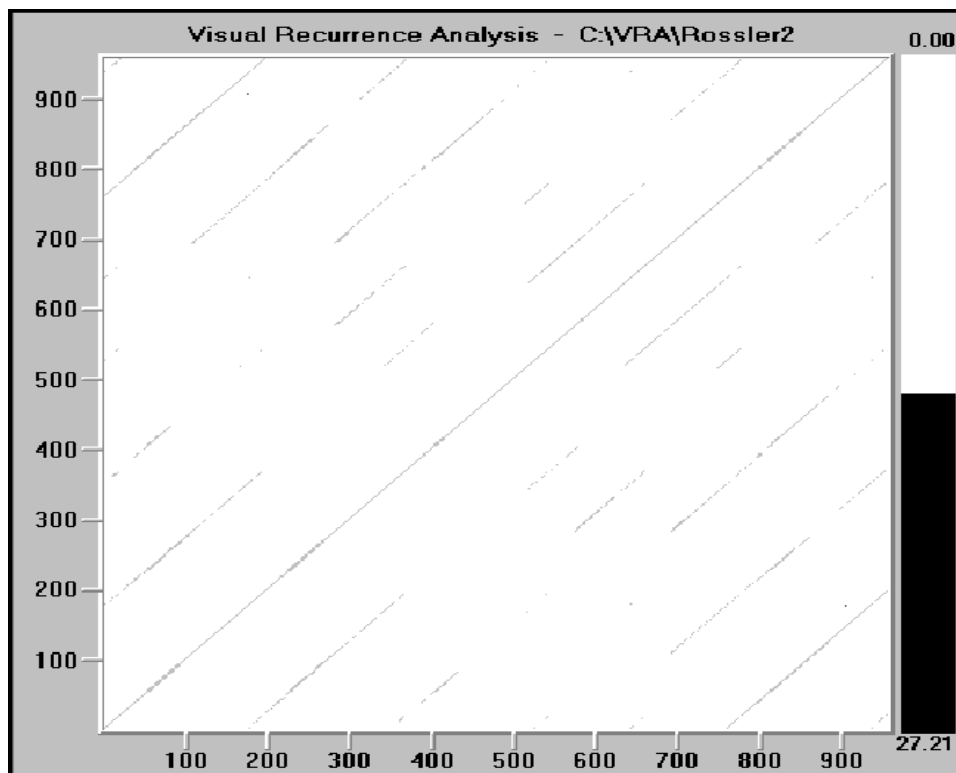


Figure 6: Graphical output from VRA

4 Visual Recurrence Analysis (VRA)

As the corresponding release notes establish, VRA is a software package written in C++ for topological analysis, qualitative and quantitative assessment, and non-parametric prediction of non-linear and chaotic time series. The program is targeted at empirical researchers using NL TSA, containing a wide range of techniques in two large sets of non standard procedures: 1) the set of quantitative recurrence analysis (the main subject of the software), and 2) the set of non-linear prediction. Both sets are menu-driven in a simple, very easy to learn, way.

VRA can be downloaded free for educational or academic research purposes. The price of the commercial version is \$199. The only difference between the freely available version and the commercial one, is the ability of the latter to save prediction values. The free version allows forecasting, but

it is not possible to save predictions into a separate file.

4.1 Installation

Installation is quite easy: download the file `vra4v2.zip` from the Web site of the writer of the program, Eugene Kononov, at the address above, unzip it into a temporary folder, run `setup.exe` from there, and follow the on-screen instructions. The version of the program runs under Windows 95, 98, 2000 and NT, and it needs 5.2 Mb of free space to be installed, and at least 8 Mb of RAM. The author also recommends a high resolution monitor and the corresponding video card in order to appreciate all the capabilities of the program.

The current version of the program is 4.2, and it will be presumably updated with additional features like cross recurrence plots, computation of Lyapunov exponents or correlation dimensions.

4.2 Starting and data manipulation

To start VRA after self-installation, users just have to double-click on the icon of the program entitled “Vra v4.2”. Next, data must be loaded from an archive, because there is no automatic loading.

Data can be read in 5 different formats: ASCII (`.dat`), comma-delimited (`.csv`), formatted text (`.prn`), sound files (`.wav`), and Excel (`.xls`). Whatever the format, scalar values must be arranged in a column, and the number of observations is limited to one million, which should be enough for most of the empirical applications in economics.

However, data management is quite restrictive: they cannot be segmented nor exported, and no transformations are allowed. Therefore, users must manipulate data with other packages (e.g., Excel), producing files which can be read by VRA (in any of the mentioned formats).

On the other hand, there is a huge set of interesting trial data files: from a uniform white noise process, to deterministic series like Rossler, Ikeda, Lorenz or Henon. The set also contains real data, like annual sun spots activity, Far-Infrared-Laser in a chaotic state, or maybe more interesting for economists, weekly closing prices of Dow Jones Industrial Average index, covering the period 01/07/1900 to 03/08/1996.

4.3 The first recurrence plot

After loading data, VRA automatically shows a first recurrence plot on the screen, using default parameters. This plot uses up to 256 colours of the so called **spectrum** scheme, an embedding dimension of one and no time delay (i.e., by default, each scalar observation is compared with the rest, without embedding); the distances are computed by the Euclidean norm, with local bounds, where the **lower bound** (minimum distance below which pixels are not coloured) is set to the smaller value, and the **upper bound** (maximum distance above which pixels are not coloured) is set to the larger value. We have to note, however, that these values are not explicitly shown but internally computed, and they can be just partially controlled, i.e., users can set a lower bound **larger** than the smaller value, but they cannot set a particular value. The parameter **bounds** (**global** or **local**) determine how the distance bounds are calculated. Jointly with **lower bound** and **upper bound**, they all form the so called **threshold corridor**.

Users can choose between 14 colour schemes; Figure 7 shows the first recurrence plot of VRA when loading a uniform white noise series (in the file `Noise.dat`), and colour scheme is changed to **Grey**.

The embedding dimension, the time delay, the method of computing distances (Euclidean, maximum norm or minimum norm), the maximum number of colours to be displayed (from 2 to 256), and the range of distances to be mapped with different colours (through the parameter **Mapping**), can also be changed by the users in order to reveal hidden patterns of data.

4.4 The time delay embedding

The first recurrence plot that VRA shows can be a beautiful picture, but absolutely uninformative. Deep insight is needed into the dynamical structure of the possible multidimensional system that generated the data. To that end, analysts must choose a suitable embedding dimension and an adequate time delay.

To choose the appropriate time delay, users can compute the average mutual information function, as an alternative to the classical autocorrelation function; the latter detects linear correlations, but the former is useful to detect both linear and non-linear correlations. The time delay should be chosen such that the elements in embedding vectors are no longer correlated, thus subsequent analysis would reveal spatial or geometrical struc-

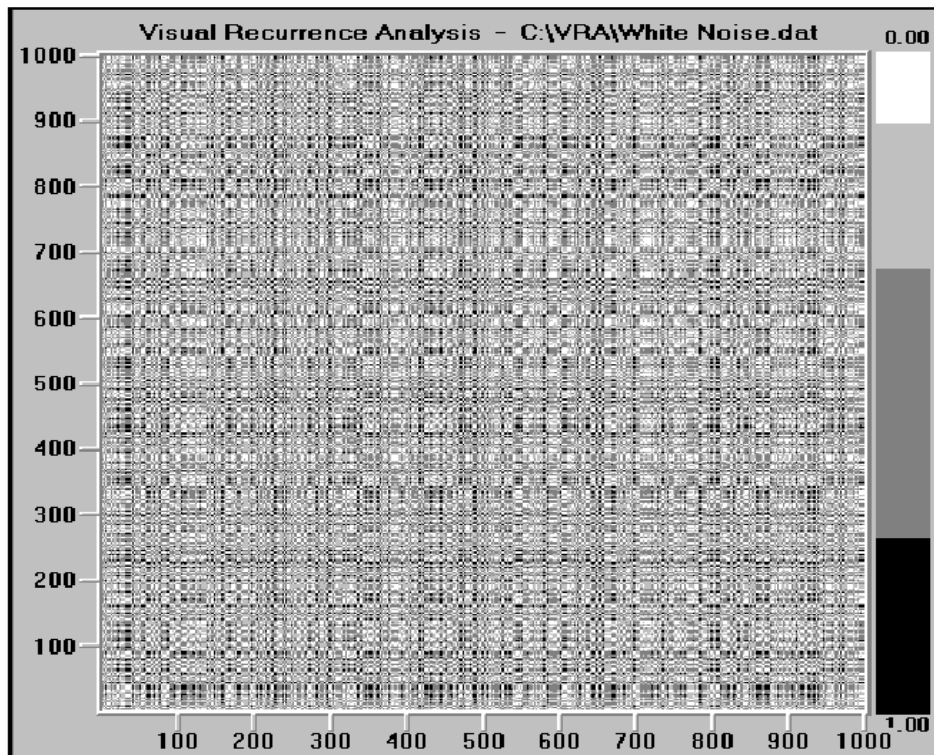


Figure 7: VRA first recurrence plot. White noise. Grey scheme

tures. In VRA, mutual information is easy to implement: in the main menu, choose the options **Analysis**, **General Nonlinear Analysis**, and then **Mutual Information**. Next, one can save the resulting graph. Figure 8 shows the procedure applied on the white noise series: the optimal delay corresponds to the first minimum in the function, in this case, at lag 1 (as it was expected).

Once the optimal delay is chosen, one could compare different recurrence plots, one per embedding parameter, as VRA does this at a very low computational cost (the plot appears almost immediately). However, one can also use a procedure called “false nearest neighbours method”, as VRA applies it very easily. Users have to choose the options **Analysis**, **General Nonlinear Analysis**, and then **False Nearest Neighbours**. Then, at the corresponding window, one can choose the minimum embedding dimension and the maximum one (along with a parameter which represents a trade-off

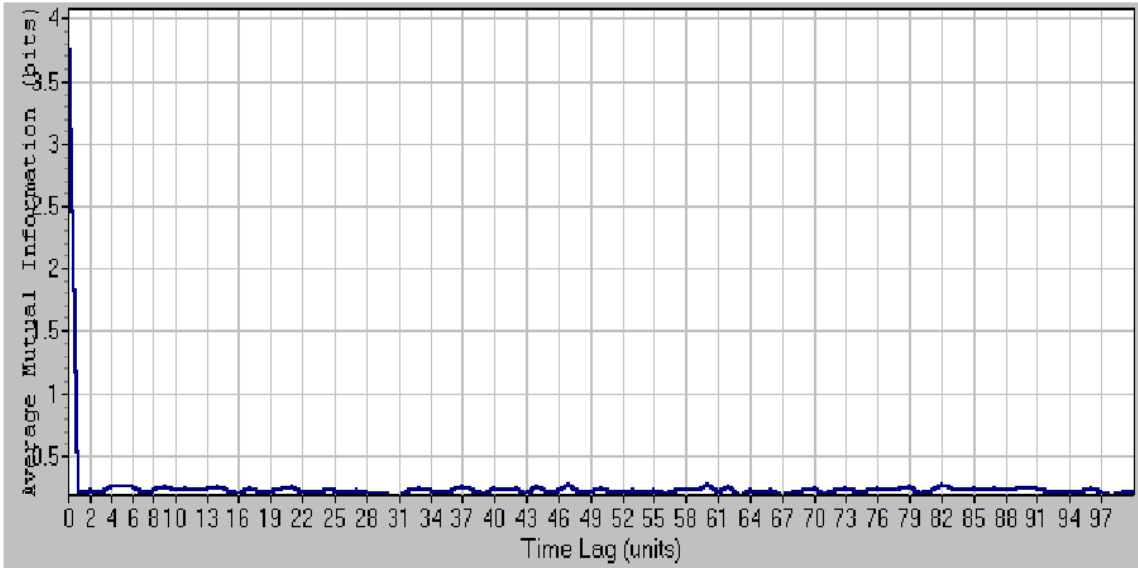


Figure 8: Average mutual information window. White noise

between speed and accuracy), and setting the optimal time delay previously calculated by the mutual information. Figure 9 shows the saved chart for the white noise series: the minimum rate of false nearest neighbours is achieved at an embedding dimension of 10.

Finally, we return to the main window of the program, select the new parameters values $m = 10$, and $d = 1$, and press the button **Apply changes**. The new recurrence plot will be shown.

4.5 Recurrence Quantification Analysis (RQA)

In addition to recurrence plots, VRA implements the recurrence quantification analysis of Zbilut and Webber. Users just have to choose **Analysis**, **Recurrence Plot Analysis**, and then **RQA Measurements**. The next window allows users to select: the embedding dimension, the time delay, the method to compute distances, the method to rescale distances, the radius (or critical distance; however, the selection ranges from **smaller** to **larger**, without specifying magnitudes), and the number of minimum consecutive points to consider as a signal of determinism (the minimum, of course, is two points). Moreover, VRA can compute recurrence magnitudes on an “epoch

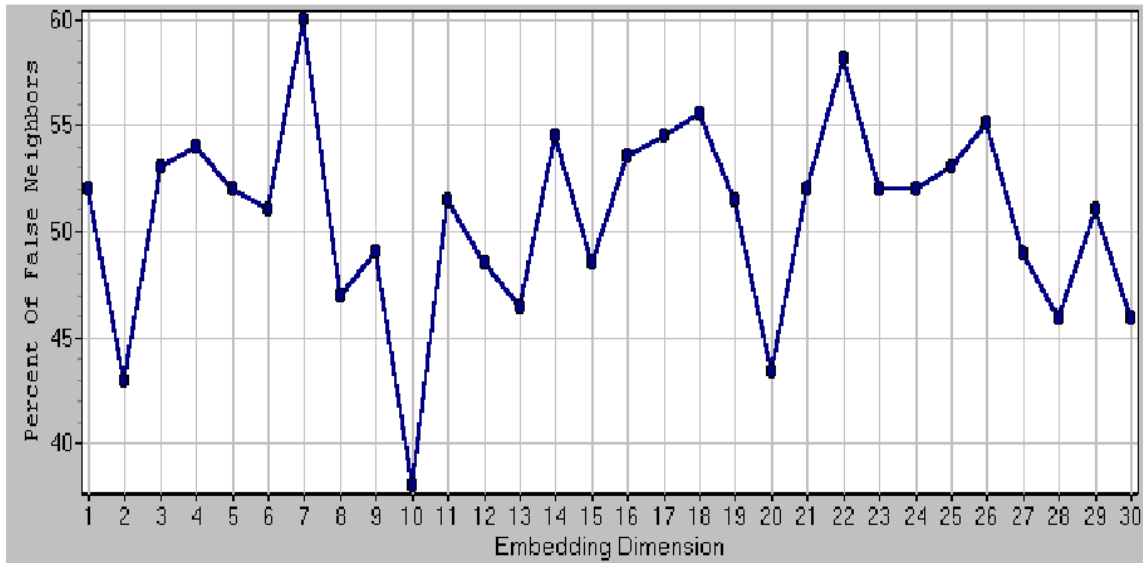


Figure 9: False nearest neighbours window. White noise

by epoch” basis, choosing the size of each epoch, the start and the end of the first epoch, the data shift and the number of epochs (depending on the choice of these parameters, users can analyze overlapped or non-overlapped periods, or just the whole period).

The program computes the variables: *Mean* (mean of input points), *StDev* (standard deviation of input points), *MeanDist* (mean of rescaled distances), *Recurrence* (%recurrence), *Determinism* (%determinism), *Ratio* (ratio %determinism to %recurrence), *Entropy*, *MaxLine* (longest diagonal line segment), and *Trend*. Users can see graphical representations of each variable as a function of the epochs, but selecting **All**, the numerical values of all the variables just computed are shown. Moreover, each graph can be saved as a bitmap file, and the numerical results can be stored into an Excel spreadsheet.

4.6 Predicting with VRA

VRA provides an important module on non-parametric forecasting, using local models by fitting a low order polynomial which maps k nearest neighbours of onto their next values, to use this map to predict future values.

In VRA, such a model can be constructed (choosing **Analysis**, and then **Time Series Prediction**) from a range of classes: nearest neighbour, locally constant, kernel regression, locally linear, locally weighted linear, and radial basis models.

To generate predictions, users must choose some control parameters: embedding dimension, time delay, the predictor (options are the methods cited at the preceding paragraph), kernel (Epanechnikov, Gaussian, bisquare, tricube, exponential, inverse, triangular, and uniform), RBF (just for radial basis functions; the options are: linear, cubic, thin plate spline, Gaussian, and multiquadric), distance (Euclidean, Manhattan, distance, maximum norm, distance by cosine, and distance by correlation), type (one step, or multi-step forecasting), train (observations included in the “training set”), predict (observations to be predicted), and neighbours (in the VRA help system, this is called “bandwith”; it controls the size of the neighbourhood, i.e., the number of neighbours used to predict). Finally, the **single neighbor from each orbit** option can be enabled to predict using just one nearest neighbour from nearby orbits. After prediction, a plot shows the actual and predicted values, jointly with the normalized prediction error (this can be excluded from the graph by disabling the option **show normalized error on chart**), and the magnitudes RMSE (root of the mean squared error) and normalized error (mean squared error normalized by the mean squared error of the trivial predictor: the unconditional mean in multi-step forecasting, or the random walk predictor in the one-step ahead predictor).

Figure 10 shows a saved chart from the prediction of the last 100 values of a Rossler x series using an embedding dimension of 3, a delay time equal to 10, 1 step ahead prediction with a locally weighted linear predictor, an Epanechnikov kernel, and 6 nearest neighbours whose distances have been computed by the Euclidean norm.

5 Conclusions

Non-linear time series analysis can be a cumbersome task, especially for researchers who are reluctant to programming their own procedures. Moreover, the techniques of NLTSA have not been added to standard econometric and statistical packages, and specific commercial software can be a costly choice.

However, free software, available through the World Wide Web, provides an attractive alternative for implementing non-standard procedures at low

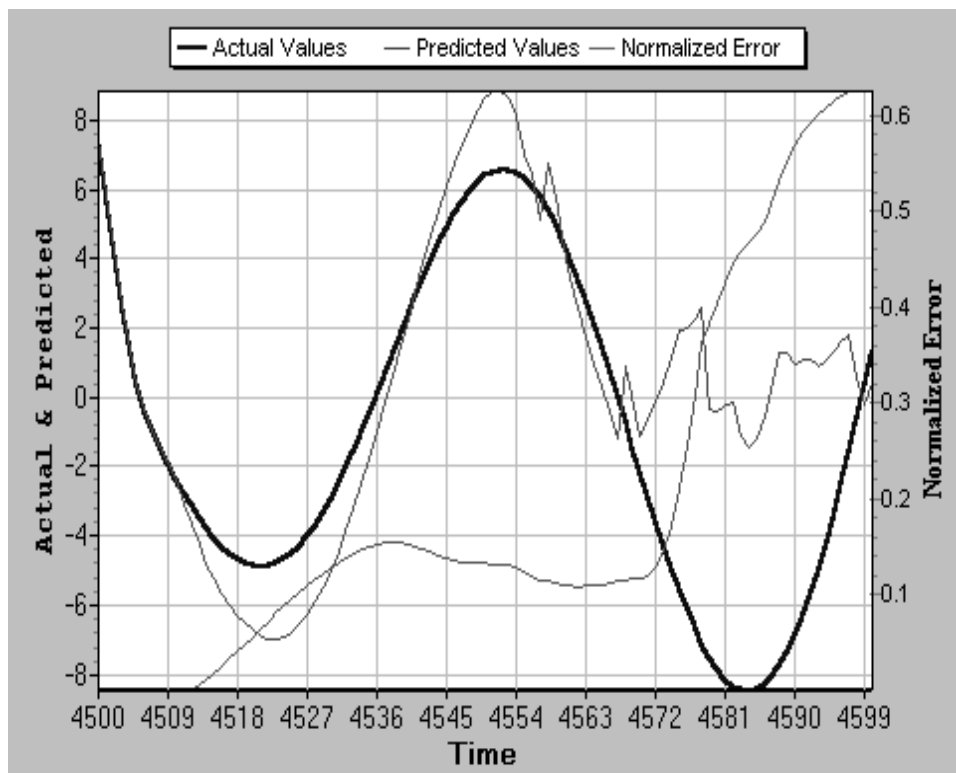


Figure 10: Non-parametric prediction window

cost. Among them, VRA stands out as easier and more user friendly than its competitors. It works under Windows in a menu-driven style, and it includes a wide range of recurrence analysis techniques jointly with a powerful prediction toolbox and a complete help system.

The main drawbacks of this non-commercial version are: graphics are not journals style (although they are very clear and informative), control on some parameters is not absolute (e.g., the so-called radius), it does not allow to saving prediction values, and it does not allow either series transformation nor selecting segments of data.

Finally, we hope that in future versions these drawbacks will be overcome, and that promising additional features will be added.

ACKNOWLEDGEMENTS

We wish to thank Lorena Tordera-Lledó and two anonymous referees for helpful comments and suggestions on earlier drafts.

References

Brock, W.A., and C. Sayers (1988): Is the Business Cycle Characterized by Deterministic Chaos?. *Journal of Monetary Economics*, 22 (1): 71-90.

Doornik, J.A. (1996): *Ox. An Object-Oriented Matrix Programming Language*. International Thomson Business Press.

Eckmann, J.P., S.O. Kampschort, and D. Ruelle (1987): Recurrence Plots of Dynamical Systems. *Europhysics Letters*, 4: 973-977.

Kantz, H., and T. Schreiber (1997): *Nonlinear Time Series Analysis*. Cambridge Nonlinear Science Series 7.

Takens, F. (1981): Detecting Strange Attractors in Turbulence. In *Dynamical Systems and Turbulence*. Rand, D., L. Young (eds). Berlin: Springer.

Zbilut, J.P., and C.L. Webber (1992): Embeddings and Delays as Derived from Quantification of Recurrence Plots. *Physics Letters A*, 171: 199-203.

Zbilut, J.P., and C.L. Webber (1998): Detecting Deterministic Signals in Exceptionally Noisy Environments Using Cross-Recurrence Quantification. *Physics Letters A*, 246: 122-128.