SIMULATION OF TRANSIENT CURRENT THROUGH POLYMETHYLMATHACRYLATE THIN FILMS BASED ON A CHARGE DENSITY WAVE MODEL

by
Gökhan Şahin
B.S., Physics, Boğaziçi University, 1999
M.S., Physics, Boğaziçi University, 2001

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Physics
Boğaziçi University
2006

SIMULATION OF TRANSIENT CURRENT THROUGH POLYMETHYLMATHACRYLATE THIN FILMS BASED ON A CHARGE DENSITY WAVE MODEL

APPROVED BY:

Prof. Dr. Yani Skarlatos ....
(Thesis Supervisor)

Prof.Dr. Avadis Hacınlıyan..................
(Thesis Co-supervisor)

Prof. Dr. Ömür Akyüz ....

Prof. Dr. Metin Arık ....

Prof. Dr. Haluk Beker ....

DATE OF APPROVAL: 01.02.2006

ACKNOWLEDGEMENTS

This work is dedicated to Prof. Dr. Avadis Hacınlıyan, Prof. Dr. Yani Skarlatos, Münevver Şahin and Mustafa Şahin, without them it would not be possible at all. I owe special thanks to my thesis advisor and co-advisor, for their critical advice, support and relaxing attitude during the progress of my work. I would also like to thank H. Ahmet Yıldırım, for his cooperation and support.

I would like to express my gratitude to Prof. Dr. Ayşe Erzan; first for mentioning the model, and then for her advice and corrections throughout the preparation of the manuscripts on which this thesis is based.

I would also like to thank to my ex-dean Prof. Dr. Hilmi Demiray and my dean at present Prof. Dr. Ömer Gökay and my ex-chairman Asst. Prof. Dr. N. Ziya Perdahçı for their understanding and strong support.

Lastly I would like to thank Özgül Kurtuluş, Kaan Atak, Murat Cihan, Nurullah Beyter and Erinç Özden, for their friendship and support.

ABSTRACT

SIMULATION OF TRANSIENT CURRENT THROUGH POLYMETHYLMATHACRYLATE THIN FILMS BASED ON A CHARGE DENSITY WAVE MODEL

A piecewise continuous, time dependent diffusive coupling in a classical one dimensional randomly pinned charge density wave model has been used for the analysis of experimentally observed transient current data for polymethylmethacrylate thin films. Satisfactory agreement has confirmed that this can be a dynamical model for the behavior of the transient current. The analysis also suggests the presence of three different regimes in the decay of the transient current.

ÖZET

YÜK YOĞUNLUĞU DALGALARI MODELİYLE POLYMETİLMETAKRİLAT İNCE FİLMLERİNDE GEÇİCİ AKIMIN MODELLENMESİ

Polymetilmetakrilat ince filmlerinde deneysel olarak gözlemlenen geçici akım, rastgele çakılmış yük dalgalarında parçalı sürekli, zamana bağlı dağınık bağlaşma (diffusive coupling) kullanılarak modellenmiştir. Bu modelin geçiçi akımın davranışını acıklamak için uygun bir model olduğu yönünde doyurucu sonuçlar bulunmuştur. Analiz ayrıca akımın sönümü sırasında üç farklı rejim olduğunu göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES xi
LIST OF SYMBOLS/ABBREVIATIONS xii
1. INTRODUCTION 1
2. EXPERIMENTAL SETUP AND TIME SERIES ANALYSIS 3
2.1. Experimental Setup 3
2.2. Time Series Analysis 4
3. CONDUCTIVITY OF PMMA 9
3.1. Dependence on the Electric Field E 10
3.2. Temperature Dependence 11
3.3. Polarons, Polymers and Charge Density Waves 11
4. QUENCHED DISORDER AND CHARGE DENSITY WAVES 13
4.1. Charge Density Wave Model 14
5. SELF SIMILARITY AND DETRENDED FLUCTUATION ANALYSIS . . 17
5.1. Fractal Objects and Self-Similar Processes 17
5.2. Detrended Fluctuation Analysis (DFA) 19
5.3. Effects of Trends 21
5.3.1. Noise with Linear Trend 21
5.3.2. Noise with Sinusodial Trend 22
6. SIMULATION 28
7. CONCLUSIONS 37
APPENDIX A: Generating Correlated noise 38
APPENDIX B: LORENZ EQUATIONS 40
APPENDIX C: SOURCE CODE FOR CHARGE DENSITY WAVE FIT . . . 42
APPENDIX D: GENERATOR FOR RANDOM NUMBERS 46
APPENDIX E: DFA SOURCE CODE 52

REFERENCES 67

LIST OF FIGURES

Figure 2.1. Experimental Setup 4

Figure 2.2. A semi logarithmic graph of a typical transient current at room temperature at 10V 5

Figure 2.3. Embedding dimension versus false neighbors 7

Figure 4.1. The upper figures show the single particle energy band a) in the case when the electron density ($\rho(r)$) and the lattice are not coupled. In that case the charge density is uniform. b) When the electron and lattice are allowed to interact, the competition between the elastic and electronic energies leads to a static lattice deformation and periodically modulated charge density. The modulations have a wave length of $\lambda_c = \frac{\pi}{k_f}$ 14

Figure 5.1. A time series of current 18

Figure 5.2. Two observation windows, with time scales n_1 and n_2. The lower picture is the magnification of the smaller window with time scale n_1 where the fluctuations look similar 24

Figure 5.3. DFA analysis of correlated noise *alpha* is 0.86 25

Figure 5.4. DFA analysis of linear trend *alpha* is 2 25

Figure 5.5. DFA analysis of superposition of linear trend and correlated noise: The crossover is at log(n)=3 26

Figure 5.6. DFA analysis of a sinusoidal trend: The crossover can be clearly seen, the slope of the first line is 2 26

Figure 5.7. DFA analysis of a superposition of correlated noise with a sinusoidal trend: Three crossovers can be observed with slopes 2,almost horizontal,0.8 27

Figure 5.8. DFA analysis of correlated noise with a slope of 0.89 27

Figure 6.1. CDW fit of current at 3.25MV/m. Both the measured current(+) and simulated CDW polarization current(×) are normalized and drawn versus number of time steps (n) 29

Figure 6.2. CDW fit of current at 2.50MV/m (current (+),simulation (×)) . . 30

Figure 6.3. CDW fit of current at 2.0MV/m (current (+), simulation (×)) . . 31

Figure 6.4. Power spectrum analysis of the observed current at 2 MV/m. The slope of the straight line is equal to -0.98 32

Figure 6.5. Plot of log F(n) vs. log n for time series of the x component of Lorenz equations showing the crossover 33

Figure 6.6. log(F(n)) vs log(n) with three different scaling exponents for the measured current (+) at 3.25MV/m and the simulation (x).The inset shows the scaling exponents for the current 34

Figure 6.7. log(F(n)) vs log(n) with three different scaling exponents for the measured current (+) at 2.50MV/m and the simulation. The inset show the scaling exponents for the current 34

Figure 6.8. log(F(n)) vs log(n) with three different scaling exponents for the measured current (+) at 2.00MV/m and the simulation(x). The inset show the scaling exponents for the current 35

Figure 6.9. Autocorrelation function vs delay time with three different regions for 2.00MV/m 35

Figure 6.10. Autocorrelation function vs delay time with three different regions for 3.250MV/m 36

Figure 6.11. Autocorrelation function vs delay time with three different regions for 3.00MV/m 36

Figure B.1. The Lorenz attractor 41

Figure B.2. The time evaluation of x component of Lorenz system 41

LIST OF TABLES

Table 2.1. First local minimum of mutual information, zero crossing of autocorrelation function and maximum Lyapunov exponents 8

Table 5.1. The slopes of crossover regions mentioned above. × indicate that there is no such region. For instance, for $log(F_{\eta}(n))$ there is only one region hence $n < n_{2\times}$ and $n < n_{3\times}$ is represented by a × . . . 23

Table 6.1. The scaling exponents, the B values, and the ξ values for the three applied electric fields 33

LIST OF SYMBOLS/ABBREVIATIONS

B	Diffusive coupling
$C(l)$	Correlation function
e	Electronic charge
E	External field
E_{th}	Threshold value for the external field
F	Root mean square fluctuation
F_d	Damping force
J	Polarization current
F_l	Root mean square fluctuation of linear signal
F_η	Root mean square fluctuation of correlated noise
F_s	Root mean square fluctuation of sinosidial signal
m	Embedding dimension of the reconstructed phase space
m_e	Electron mass
n	Box size
P	Polarization
q	Wave number
$R_d(k)$	Distance in d dimensions
s	Standard Deviation
$S(k)$	Fourier transform of correlation function
T	Temperature
T_t	Transit time of the carrier front
T_p	Peierl's temperature
U	Energy density
V	Potential Difference
v_F	The Fermi velocity
$y_n(k)$	Integrated time series in box n
α	Parameter of self similarity
ϵ	Dielectric constant

λ_c	Wavelength of charge density waves
λ	Electron phonon coupling
ρ_0	One dimensional electron density
ρ_m	The effective mass density of the CDW
$\rho(\cdot)$	Charge density wave function
ϕ_i	Quenched random variable
ξ	Dimensionless constant related to E field

CDW	Charge Density Wave
DFA	Detrended Fluctuation Analysis
PMMA	Polymethylmethacrylate
PPV	polyphenylenevinylene

1. INTRODUCTION

Polymeric dielectrics have complicated structures that may contain many impurities and additives. They are also known to be very sensitive to their thermal, mechanical and electrical history [1, 2, 3, 4]. The difficulty of obtaining identical results under nearly identical conditions has been reported in the literature [2].

Polar polymers such as Polymethylmethacrylate (PMMA), are known to need very long times to reach a steady state current. From theoretical considerations [1], this time has been estimated to be approximately a hundred years for PMMA. In spite of the dependence on the history, experimentally observed reproducible chaotic behavior as reflected by a positive Liapunov Exponent in the transient current has been reported [5, 6].

The experiment, which this work is a result of, was originally designed for finding the steady state value for the current in an attempted study of conductivity mechanisms in PMMA samples. The relatively peculiar behavior of the observed transient current and the unusually long times required for reaching a possible steady state implied the alternative of chaotic behavior.

Initial studies for ascertaining possible chaotic behavior was carried out [5, 6, 7] using time series analysis. In order to furnish a theoretical interpretation to the observation of a positive maximal Lyapunov exponent, a classical one-dimensional model of randomly pinned charge density waves (CDW) is proposed [8]. The aim is to understand the experimentally observed decaying transient current as a time dependent dynamical system based on this model. There are three reasons behind this choice: i) the qualitative resemblance of the observed decaying patterns of the current, and simulations based on this model; ii) the observation of a positive Lyapunov exponent in the data which also agrees with the model; iii) the similarity of the observed electric field dependence of conductivity with this model [6, 9].

History dependence is also known to be a property of spin glasses, polymers [10] as well as processes described by a continuous time random walk [11, 12]. The proposed model inherits glassy dynamics, is akin to the short-range Ising spin glass, and can describe random walks on directed percolation clusters [9]. All of these systems show memory effects.

Another reason for the choice of the model is that it makes use of quenched randomness, which accounts for the heterogeneity of the polymer and reflects nonlinearity in the transient current.

A piecewise continuous, time dependent diffusive coupling in a classical one dimensional randomly pinned charge density wave model has been used for the analysis of experimentally observed transient current data for polymethylmethacrylate. Satisfactory agreement between the data and simulation based on the model mentioned above has confirmed that this can be a dynamical model for the behavior of the transient current. The analysis also suggests the presence of three different regimes in the decay of the transient current. The results have also been confirmed by Detrended Fluctuation Analysis on both the data and the simulation.

The outline of this work has been influenced by the the comments of an anonymous referee of the journal Phys. Rev .B. The plan is as follows: In the second chapter the experimental set up and the time series analysis is discussed. The third chapter is about the conductivity of PMMA. In the Fourth chapter detrended fluctuation analysis is introduced. Fifth chapter gives the details of simulation. The last chapter is conclusion.

2. EXPERIMENTAL SETUP AND TIME SERIES ANALYSIS

In this section, we present the details of the experimental setup for observing and recording the transient current in PMMA samples. For completeness, we then give a brief summary of the time series analysis of the relatively peculiar behavior in the transient current leading to the observation of a positive maximal Lyapunov exponent. Details of this analysis have been presented elsewhere [6, 7].

2.1. Experimental Setup

The specimens under investigation were prepared as sandwiched metal-polymer-metal structures with PMMA as the isolating layer. 300 nm thick aluminum electrodes were thermally evaporated at 10^{-6} mbar on microscope glass slides cleaned in a detergent solution. $20\mu m$ thick PMMA layers were deposited from 6% PMMA solution in toluene. Subsequently, aluminum top contacts were evaporated. The I-V measurement was performed via a programmable picoammeter/voltage source (Keithley, model 487) and a temperature controller (Lake Shore, model 300) at the Boğaziçi University Physics Department Solid State Laborotary. The picoammeter and the temperature controller were interfaced to a computer through an interface card that automated data taking, as shown in Fig.3.4.

The picoammeter model 478 used is capable of reading currents in the range 10 fA to 2 mA. It also serves as a DC voltage supply in the range $\pm 500V$. In an I-V measurement a given voltage range $V_{\min}$ to $V_{\max}$ is scanned by constant increments V at constant time intervals Δt. ΔV and Δt are controlled by the computer in the following manner:

- The computer drives the voltage supply to apply the lower voltage limit $V_{\min}$ to the sample and to wait for a prescribed time in order to let the current settle down.

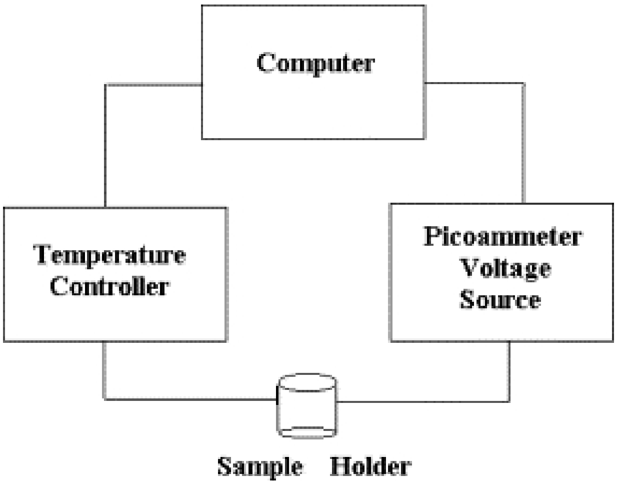

Figure 2.1. Experimental Setup

- When 9/10 of Δt elapses, the current values are averaged over very short time intervals with respect to time interval Δt.
- At the end of Δt seconds, the voltage and the corresponding averaged current value for the last one tenth of Δt are recorded in the output file. Then the voltage is incremented by ΔV.
- The procedure is applied repeatedly until the voltage reaches $V_{\max}$.

When a DC voltage is applied, the current in the system achieves a stationary state only after a definite time [13]. With the same setup, the extent of the settling time Δt is determined by applying a small voltage on the sample and then recording the time evolution of the resulting current.

The data was taken under different voltage values ranging from 10V to 80V and at a constant temperature of $295^o K$.

2.2. Time Series Analysis

The observed fluctuations that saturate in a time scale around 9000 s are displayed in the figure below. It can be seen that the transition region shows a typical pattern that can be associated with chaotic behavior.

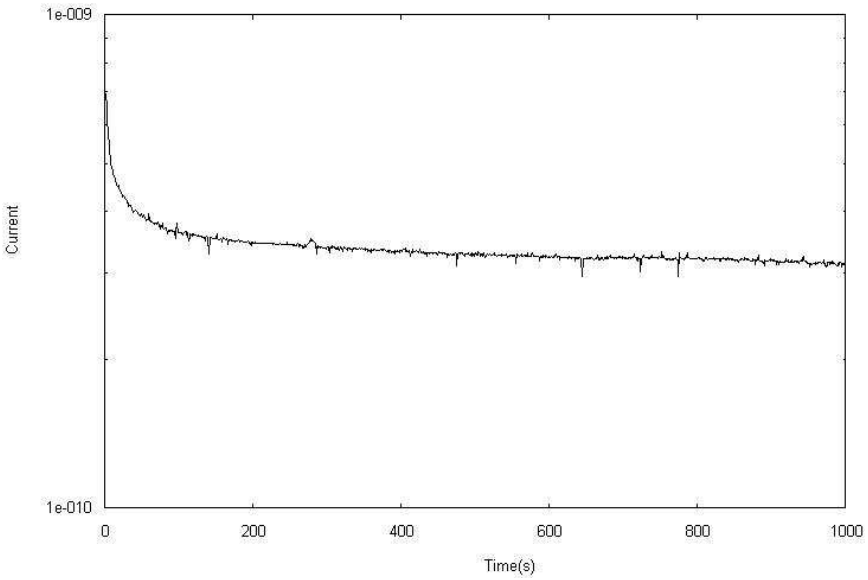

Figure 2.2. A semi logarithmic graph of a typical transient current at room temperature at 10V

Applying nonlinear time series analysis on the transient current required a phase space reconstruction and analysis by using the nonlinear time series analysis methods as described in the TISEAN software package and literature [14, 15, 16].

Details of the phase space reconstruction from the scalar transient current s(k), where k means the k'th time step, follow the well known procedure. Details will only be given to the extent needed to set the notation. Time delay vectors $\vec{y}(k)$ given by

$$\vec{y}(k) = [s(k), s(k+\tau), ..., s(k+(d-1)\tau)]\, \vec{y}(k) \in R^d \tag{2.1}$$

where τ denotes the delay time and d denotes the embedding dimension are constructed. There are no clear cut rules for their determination since a limited range of data is available and noise is present. The time delay is found from [15] the first zero of the linear autocorrelation function given by

$$C(\tau) = \frac{\frac{1}{N}\sum_{m-1}^{N}\left[s(m+\tau)-\bar{s}\right]\left[s\left(m\right)-\bar{s}\right]}{\frac{1}{N}\sum_{m-1}^{N}\left[s\left(m\right)-\bar{s}\right]^{2}} \tag{2.2}$$

where

$$\bar{s} = \frac{1}{N}\sum_{m=1}^{N} s\left(m\right). \tag{2.3}$$

Another method for the determination of the delay time is to find the first minimum of the average mutual information. This can be used as if it were a nonlinear correlation function given by [17],

$$\begin{aligned} I\left(\tau\right) = I_{AB} &= \sum_{a_\imath b_j} P\left(s\left(n+\tau\right), s\left(n\right)\right) \\ &\times \log_2\left[\frac{P\left(s\left(n\right), s\left(n+\tau\right)\right)}{P\left(s\left(n+\tau\right)\right)P\left(s\left(n\right)\right)}\right]. \end{aligned} \tag{2.4}$$

Here $P\left(s\left(n\right), s\left(n+\tau\right)\right)$ is the joint probability that if at time n, $s\left(n\right)$ is measured, then, at time $n+\tau$, $s\left(n+\tau\right)$ is measured and $P(s(n))$ is the probability of measuring $s\left(n\right)$ [15, 18]. Although two widely different time intervals have been observed in two different methods the shorter one derived from the mutual information analysis has been used. The longer time reflects the overall decay and is not expected to affect the fluctuations.

The embedding dimension is determined by using the method of false nearest neighbors[19]. A typical graph is given below showing that an embedding dimension of four drastically reduces the number of nearest neighbors. In order to ascertain this embedding dimension, the Lyapunov exponent calculation has been repeated for embedding dimensions five, six and seven. There is no significant change in the result.

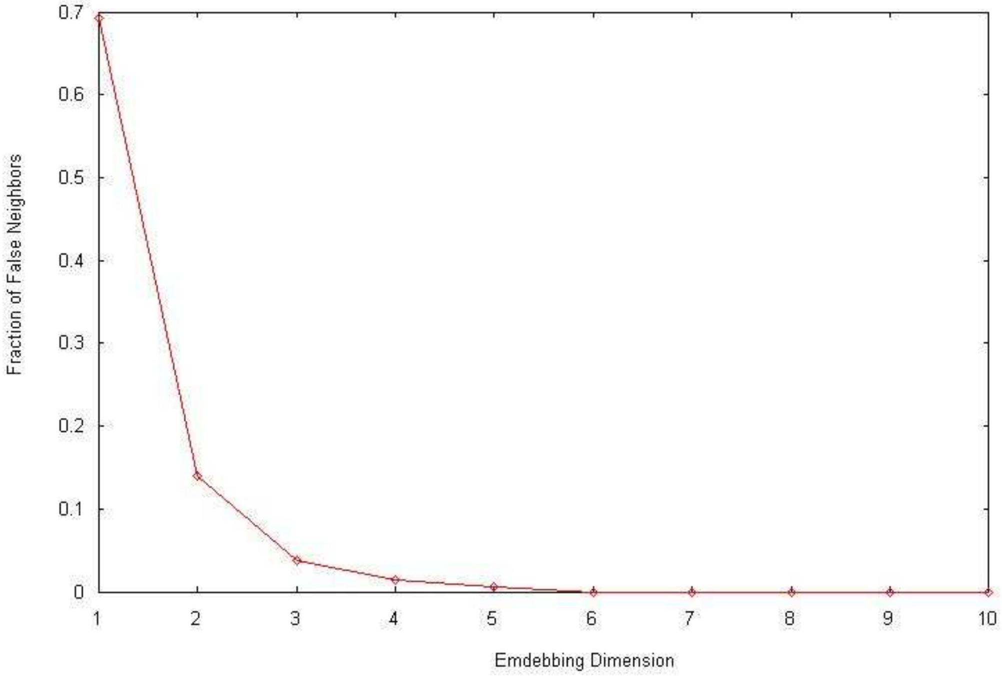

Figure 2.3. Embedding dimension versus false neighbors

Since we are interested only in detecting the presence of chaotic behavior in the transient data, the calculation is limited to the calculation of the maximal Lyapunov exponent. Possible ways of performing these calculations are given in many references including the following [20, 21, 22]. The stretching factor approach is preferred since it is expected to minimize Gaussian noise and possible truncation error effects with relatively moderate computational effort [23].

$$S(\Delta n) = \frac{1}{N}\sum_{n_0=1}^{N} \ln\left[\frac{1}{|u_n(\vec{s}_0)|}\sum_{\vec{s}\in u(\vec{s}_0)} |\vec{s}_{n_0+\Delta n} - \vec{s}_{n+\Delta n}|\right]. \tag{2.5}$$

Table 2.1. First local minimum of mutual information, zero crossing of autocorrelation function and maximum Lyapunov exponents

E Field	Mutual Inf.	Autocorrelation	Max. Lyapunov Exp.
0.50 MV/m	30s	1186s	0.0032 ± 0.0001
1.00 MV/m	30s	522s	0.0065 ± 0.0005
1.50 MV/m	60s	509s	0.0150 ± 0.0007
2.50 MV/m	380s	1246s	0.0084 ± 0.0002
2.75 MV/m	90s	549s	0.0115 ± 0.0007
3.00 MV/m	110s	638s	0.0135 ± 0.0001
3.25 MV/m	20s	1186s	0.0159 ± 0.0006
3.50 MV/m	400s	638s	0.0178 ± 0.0003
3.75 MV/m	80s	212s	0.0226 ± 0.0001
4.50 MV/m	140s	709s	0.0161 ± 0.0001

Here $\vec{s}_{n_0}$ is the embedding vector, chosen as a reference point. We select all the neighbors with distance smaller than ε, (denoted by $u_n(\vec{s}_0)$), and average over the distances of all neighbors to the reference point at time Δn. If $S(\Delta n)$ shows a linear robust increase for Δn then the slope is estimated as the maximal Lyapunov exponent. This choice of the estimator for the maximal Lyapunov exponent is favored because it does consider the fluctuations due to noise, limited data, etc. Details of the estimator can be found in [24].

Results of this analysis can be summarized as follows. The delay time analysis of the currents gave two different time scales as it was mentioned in ref [5]. The first local minimum of the mutual information function gave values ranging between 20s to 400s. But the values obtained from the first zero crossing of the autocorrelation function are in the range of 200s to 2000s. For most of the cases four seems to be a satisfactory value for the minimum embedding dimension. The calculated values of the maximal Lyapunov exponents are given in Table 2.1.

3. CONDUCTIVITY OF PMMA

Physical properties of polymeric dielectrics and particularly, their transient conductivity vary in a sensitive way depending on their thermal, mechanical and electrical history [1, 2, 3, 4]. This has been a reason why a lot of effort has been necessary in order to ascertain the chaotic behavior in a reproducible manner. The many impurities and additives that these substances contain is another factor leading to the difficulty of obtaining identical results under nearly identical conditions. This has also been reported in the literature [2].

Chaotic structures usually involve long time behavior. It is known that polar polymers such as PMMA need very long times to achieve steady state current. In ref [1], this time is estimated to be approximately a hundred years for PMMA. For this reason, the transient behavior of the current can have properties that can be considered as long time behavior.

When an electric field is applied to a dielectric, bound and free charges interact with the electric field. If the field is strong enough, creation of charge pairs by breakup of dangling bonds or trapping of free charge carriers becomes possible. This complicated sequence of interactions causes the transient current observed in measurements and one can expect several regimes. In general, after the application of voltage, the current falls off with fluctuations (polarization current), and seems to become steady only when a very long time has elapsed. Some of the causes of this behavior can be summarized as[4],

- Fast kinds of polarization, e.g. resonance and some types of dipole orientation.
- Slow types of dipole relaxation polarization.
- Flow of conduction current caused by the motion of charges injected from the electrodes or generated by thermal ionization of impurities or of the dielectric itself, or produced by photo-ionization or high-energy radiation ionization.
- Relaxation polarization caused by micro- or macro-heterogeneities of a continuous

or discrete nature.

- Trapping of charge carriers in the bulk of the dielectric.

The carrier transport in PMMA is reported to have a non-Gaussian (or dispersive) character [3] in time. The general behavior of non-Gaussian transient transport of carriers in disordered solids is summarized in the following equations [25, 26]:

$$\begin{aligned} i(t) &\propto t^{-(1-\alpha)} \qquad \text{for} \qquad t < T_t, \\ i\,(t) &\propto t^{-(1-\beta)} \qquad \text{for} \qquad t > T_t, \end{aligned} \tag{3.1}$$

where T_t is interpreted as being the transit time of the carrier front whereas α and β are parameters describing the degree of dispersion. Such a power law according to ref. [27, 28] might be caused by relaxation from non-uniform but scaling structures that can mimic fractal properties.

The conductivity of PMMA is known to depend on the electric field and temperature. The details are summarized below

3.1. Dependence on the Electric Field E

It is known that conductivity values are strongly field dependent[2]. The field dependence is mostly attributed to trapping of free charge carriers in the volume of the dielectric during their motion due to the applied field, whereas release from traps is considered to be thermally activated with a field modified activation energy [26]. These charges are believed to be trapped in some of the localization states arising from the defects in the structure of the insulator such as impurities, dopants and dangling bonds. It has been suggested that side groups may act in a way similar to doped impurities [29].

3.2. Temperature Dependence

The main relaxation processes in PMMA are α and β processes related to the cooperative dynamics (collective reorientation of the side groups with adjacent main chain segments —C—CH2—) and to the local dynamics (the reorientation of the polar ester side groups ($---COOCH_3$) by local motions around the C—C bond [1]. Both processes (hence the glass transition temperature) are known to be dependent on the film thickness, the type of substrate and temperature [30].

3.3. Polarons, Polymers and Charge Density Waves

When an insulator is placed in an electric field [31], it acquires electrostatic charges which are trapped in some of the localization states. The trapping is a consequence of the defects in the structure of the insulator such as impurities, dopants and dangling bonds. One of the models for interpreting the charge-trapping process is the polaron model [32, 33]. According to the model, when an electron enters an insulator, the medium surrounding the electron will be polarized and distorted. A localization state called polaron will be formed, which will trap the electron.

In case of conjugated polymers transport occurs by the movement of charge carriers between localized states or between polaron, or bipolaron states [34] (The term conjugated means an alternation of multiple and single bonds linking a sequence of bonded-atoms, such that there is an extended series of overlapping π orbitals. Polyphenylene and polyphenylene-based molecules are examples of conjugated polymers [35]). The electronic structure of conjugated polymers is described by ref [36] in terms of a quasi-one-dimensional model. According to which the π-electrons are coupled to distortions in the polymer backbone by the electron-phonon interaction. In the model; excitations across the p - p * band gap create the self-localized, nonlinear excitations of conducting polymers; solitons (in degenerate ground state systems); polarons; and bipolarons [37].

For instance in another polymer, poly (phenylene vinylene) (PPV) the existence of both polaron and bipolaron states are known. Both spin and charge density waves of polaron states have also been reported [38]. Although we have not seen any reports on the existence of charge density waves in PMMA, polarons have been observed and charge trapping in PMMA can be explained in terms of a polaron model [39].

4. QUENCHED DISORDER AND CHARGE DENSITY WAVES

The effect of quenched disorder on a periodic medium has been known to display many phenomena which include the nature of the depinning transition, the effect of disorder on structural properties. charge-density waves (CDW) [40] is one example of such systems. An effect of quenched disorder on a system that would normally display periodicity is pinning the system in a disordered state, thus reducing or destroying that periodicity. However, a sufficiently large driving force in a non equilibrium system can act to depin the system, and reduce the effect of quenched disorder to that of annealed disorder, that behaves in a manner analogous to thermal noise [41].

A charge density wave (CDW) is a periodic modulation of the electron density and an associated modulation of the lattice below a critical transition temperature (T_p called Peierls' temperature). Quasi one-dimensional metals are typical systems that exhibit CDW.

Due to the periodicity of the lattice deformation, the electron density will also become periodically modulated (see Figure 4.1). As a result of the modulation, a gap opens up in the single-particle excitation spectrum at the Fermi level, and a spatially periodic charge density modulation is formed with wave vector 2kF. The deformation is limited by the corresponding increase in elastic energy. The CDW state has an energy that is lower than the uniform state.

A large number of materials are known to undergo Peierls transition whereas only a small fraction of these show collective charge transport[42, 43]. Charge density waves are pinned to the underlying lattice. If the CDW's wavelength λ_c is an integral multiple of the lattice period then its energy will oscillate with a period λ_c else if the CDW's wavelength is incommensurate with the lattice period then it will be pinned by impurities and other lattice defects. In case of impurities CDW would elastically deform so as to minimize its impurity interaction energy. CDW in both commensurate

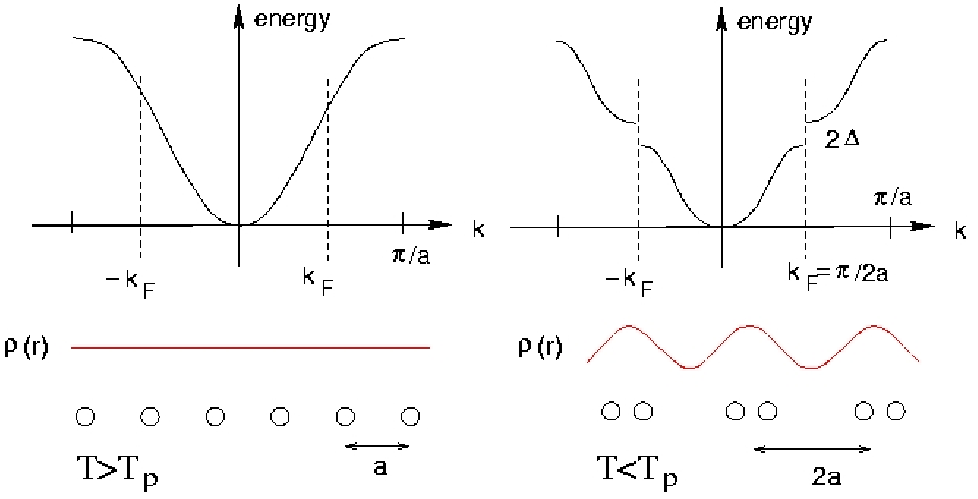

Figure 4.1. The upper figures show the single particle energy band a) in the case when the electron density ($\rho(r)$) and the lattice are not coupled. In that case the charge density is uniform. b) When the electron and lattice are allowed to interact, the competition between the elastic and electronic energies leads to a static lattice deformation and periodically modulated charge density. The modulations have a wave length of $\lambda_c = \frac{\pi}{k_f}$

and incommensurate cases would stay pinned for small electric field. However in the commensurate case if a field larger than a threshold field is applied, the CDW will start to slide, hence a collective charge transport would occur.

4.1. Charge Density Wave Model

Different models have been proposed to describe CDWs. The most widely studied model is the FLR model developed by Hidetoshi Fukuyama, Patrick Lee and T.Maurice Rice [44, 45]. In the FLR model CDW is treated as classical extended elastic medium. This medium interacts with random impurities and couples to an electric field. Another approach treats the CDW as classical particle in a periodic potential [46]. There are also semi-classical approaches [47]. None of the approaches above are complete in the sense that they can not fully interpret the experimental results [42, 48, 49].

The one dimensional model used in this work is based on the works of Pietrenero et. al. [8] and Erzan et. al. [9]. The model is a classical model of an over-damped CDW. The charge density wave is taken in the form

$$\rho\left(x\right)=e\rho_0\left[1+C\cos\left(q_0x+\phi\left(x\right)\right)\right]. \tag{4.1}$$

Then the overdamped equation of motion for the phase ϕ_i of a pinned charge density wave (CDW) is given by [8]

$$-\frac{K}{q_0^2}\frac{\partial^2\phi}{\partial x^2}-\frac{\rho_m}{q_0^2\tau_0}\sum_j\dot{\phi}\delta(x-x_j)+eCV_0\rho_0\times\sum_j\sin[q_0x_j+\phi(x)]\delta(x-x_j)+\frac{e\rho_0}{q_0}E=0 \tag{4.2}$$

where C=$\frac{N_0\Delta}{a\lambda\rho_0}$, N_0 is the constant density of states per site, a is the lattice constant, Δ the gap, λ the electron phonon coupling, l=$2\pi/q_0$ the wavelength, ρ_0 the one dimensional electron density, ρ_m the effective mass density of the charge density wave, V_0 the intensity of the interaction of the charge density wave with the local impurities whose densities are n_i. K=$\rho_0mv_f^2$ with m the free electron mass and v_f the Fermi velociy and τ is a phenomenological parameter describing the dissipation of energy from the charge density wave to the lattice. The above equation of motion can be replaced with a difference equation using finite differences and the latter can be integrated between any two consecutive impurity sites numerically.

Introducing the dimensionless variables u=n_ix,$\xi=\frac{E}{CV_0n_iq_0}$, B=$\frac{2\pi Kn_i}{CV_0\rho_0q_0^2}$, s=$\frac{t}{\tau_0}$ where $\tau_0=\frac{2\pi\rho mn_i^2}{CV_0\rho_0q_0\tau}$ and $\psi=\frac{\phi(x)}{2\pi}$ and $Q_0=\frac{q_0}{2\pi n_i}$ the difference equation becomes ,

$$\frac{d\psi_j}{ds}=B[\frac{\psi_{j+1}-\psi_j}{r_{j+1,j}}-\frac{\psi_j-\psi_{j-1}}{r_{j,j-1}}]-\sin[2\pi(u_jQ_0+\psi_j]+\xi Q_j \tag{4.3}$$

where $Q_j=\frac{1}{2}(r_{j+1,j}+r_{j,j-1})$, $j\geq 1$. If one introduces the quenched random variable

$\phi_j = u_j Q_0$, $1 \leq \phi_j \leq 1$, and takes the intervals between the impurity sites to be uniform the equation of motion becomes

$$\frac{d\psi_j}{ds} = B[\psi_{j+1} + \psi_{j-1} - 2\psi_j] - \sin[2\pi(\phi_j + \psi_j)] + \xi \tag{4.4}$$

where the dimensionless constants ξ and B correspond to the electric field and to the diffusive coupling respectively. In terms of ψ_j the current density is given by

$$J = en_e \frac{2\pi}{q_0 \tau_0} < \frac{d\psi_j}{ds} > \tag{4.5}$$

where n_e is the bulk density and <> represents an average over all impurity sites.

The results of the model can be summarized as follows [8]:

(I) Nonlinear polarization.

(II) In the strong pinning case, The CDW splits into independent portions of different length each pinned by an impurity which corresponds mathematically to many domains.

(III) the existence of both broad and narrow band noise in the polarization current [9].

(IV) the observation of a positive maximal Lyapunov exponent on simulated time series of the current density based on the model[9].

5. SELF SIMILARITY AND DETRENDED FLUCTUATION ANALYSIS

5.1. Fractal Objects and Self-Similar Processes

A fractal object can be described as a geometrical object satisfying two criteria: self-similarity and fractional dimensionality. Self-similarity means that an object can be divided into sub-units and sub-sub-units that (statistically) resemble the structure of the whole object [50]. The second criterion for a fractal object is that it should have a fractional Hausdorff dimension. This distinguishes fractals from Euclidean objects whose dimensions are integer.

In order to extend the idea of a fractal structure to the complex temporal processes (ie time series) the following challenge has to be faced: A time series involves two different physical variables whereas for geometrical curves both axes represent the same physical variable. For example, both axes represent lengths for a fractal coastline embedded in a two dimensional plane. On the other hand, in our studies one axis represents time whereas the other represents the transient current as shown in figure 5.1.

Self-similarity of a two dimensional set is determined by taking a subset of the object and rescaling it to the same size of the original object with the same magnification factor for both its width and height. Then the statistical properties of the rescaled object with the original object are compared. In order to compare a subset of a time series with the original data set, two magnification factors (along the horizontal and vertical axes) are needed in light of the argument mentioned above [51].

In mathematical terms self similarity of time series ($x(t)$) is defined as

$$x(t) \equiv a^{\alpha} x(\frac{t}{a}) \tag{5.1}$$

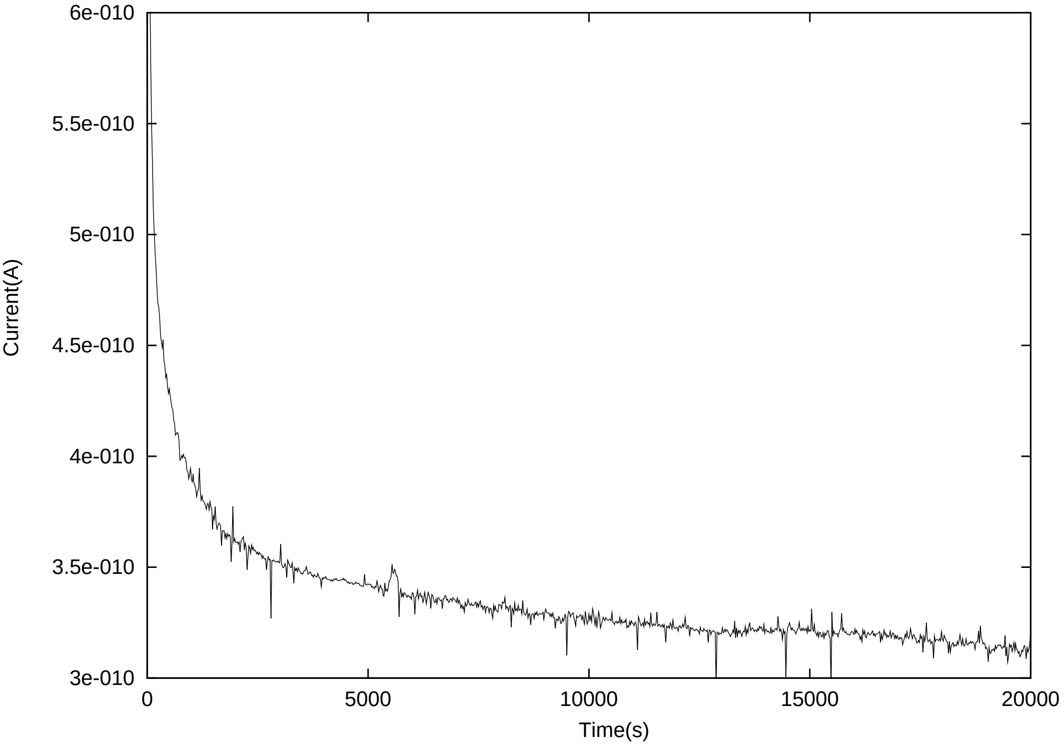

Figure 5.1. A time series of current

where $\equiv$ means that the statistical properties of $x(t)$ and $a^{\alpha}x(\frac{t}{a})$ are identical. α is called the parameter of self-similarity. By statistical identity, equality of all the moments of the probability distribution is meant. Since it is almost impossible to meet the above statistical identity criterion in real data, usually Eq. 5.1 is approximated by the means and variances (first and second moments) of the distribution functions .

In order to find α from a given time series (adopting the weak criterion of self-similarity), two magnifying factors (named mag_x and mag_t) are needed . Assume one has two observation windows with horizontal sizes n_1 and n_2 (see figure 5.2). In the x direction the magnifying factor is trivial, $mag_t = \frac{n_2}{n_1}$. In the y direction one needs to determine the vertical characteristic scales of the two observation windows. A reasonable estimate can be defined by using the standard deviations of these two observation windows, denoted as s_1 and s_2 respectively ([51]). Hence $mag_x = \frac{s_2}{s_1}$, and from Eq. 5.1,

$$\alpha = \frac{ln(mag_x)}{ln(mag_t)} = \frac{ln(s_2) - ln(s_1)}{ln(n_2) - ln(n_1)}. \tag{5.2}$$

This relation is just the slope of a line joining, (n_1,s_1) and (n_2,s_2) on a log-log plot. The calculations for determining the scaling exponent α from a time series are carried out using the following procedure:

(I) For a given size of the observation window the time series is divided into subsets of independent windows of the same size.

(II) All individual values of the standard deviation s obtained from these subsets are averaged in order to obtain a reliable estimate of the characteristic fluctuation at this window size.

(III) These calculations are repeated for many different window sizes

(IV) α is estimated by fitting a line on the log-log plot of s versus n across the relevant range of scales.

5.2. Detrended Fluctuation Analysis (DFA)

For a self-similar process with $\alpha > 0$, the fluctuations are expected to grow with the window size. Therefore, the fluctuations on large observation windows are larger than those of smaller windows. As a result, the time series is unbounded. However, most time series are bounded–they cannot have arbitrarily large amplitudes no matter how long the data set is. A solution to this problem is studying the fractal properties of the accumulated (integrated) time series, rather than those of the original signals [50, 52, 53, 54].

When a bounded time series is mapped to a self-similar process by integration, if the time series is non-stationary, integration process magnifies further the non-stationarity of the original data.

Detrended fluctuation analysis was originally introduced to overcome this problem [55, 56, 57]. DFA is reported to have advantages over conventional methods (e.g., spectral analysis and Hurst analysis). It permits the detection of intrinsic self-similarity embedded in a seemingly non stationary time series, and also avoids the spurious detection of apparent self-similarity, which may be an artifact of extrinsic trends [58, 59, 60, 61, 62]. Hence it serves as a scaling analysis method used to estimate long-range power-law correlation exponents [57, 62, 63]

The procedure for DFA is as follows (assuming that the time series is represented by $x(t)$):

(I) First the time series is integrated. This integration step maps the original time series to a self-similar process.

(II) To measure the vertical characteristic scale of the integrated time series, the integrated time series is divided into boxes of equal length, n. In each box of length n, a least squares line is fitted to the data (representing the trend in that box). The y coordinate of the straight line segments is denoted by $y_n(k)$.

(III) Next, the integrated time series, y(k), is detrended, by subtracting the local trend, $y_n(k)$, in each box.

The root-mean-square fluctuation of this integrated and detrended time series is calculated by

$$F(n) = \sqrt{\frac{1}{N}\sum_{K=1}^{N}[y(k) - y_n(k)]^2}. \tag{5.3}$$

This computation is repeated over all time scales (box sizes) to characterize the relationship between F(n), the average fluctuation, as a function of box size, n. A linear relationship on a log-log plot indicates the presence of power law scaling. Under such conditions, the fluctuations can be characterized by a scaling exponent, α, such that $F(n) \sim n^{\alpha}$. α values indicate the behavior as follows: [51]

(i) $\alpha = 0.5$, the time series are uncorrelated [64].

(ii) $0.5 < \alpha \leq 1$, indicates persistent long-range power-law correlations, the case $\alpha = 1$ corresponds to $\frac{1}{f}$ noise.

(iii) $alpha > 1$ indicates that correlations exist but cease to be a power law, $\alpha = 1.5$ indicates Brownian noise.

The slope of the line relating log F(n) to log n determines the scaling exponent (self-similarity parameter), α. A crossover in the scaling exponent, α, indicates a transition from one type to a different type of underlying correlation, due to a transition in the dynamical properties [62, 57].

5.3. Effects of Trends

When DFA analysis is applied on a signal, the scaling exponent does not always happen to be constant; crossovers often exist. Different value for the scaling exponent α is found in different ranges of scales [65, 66, 67, 68].

A crossover might happen as a result of either trends in the signal, or a change in the correlation properties of the signal. For instance, the number of particles emitted by a radiation source in an unit time has a trend of decreasing because the source becomes weaker[69, 70].

Below is a summary of the effect of different trends (e.g., polynomial, sinusoidal and power-law trends) on the scaling behavior of time series [57].

5.3.1. Noise with Linear Trend

Linear trend (with the slope A_L) is denoted by

$$u(i) = A_L i. \tag{5.4}$$

The root mean square (rms) fluctuation function for correlated noise with standard deviation one, can be aproximated by [57]

$$F_\eta(n) = b_0 n^\alpha \tag{5.5}$$

where b_0 is a parameter independent of n (box size).

The rms fluctuation function for for linear trend is

$$F_L(n) = k_0 A_L n^{\alpha_L} \tag{5.6}$$

where k_0 is a constant independent of n.

Then the correlated noise and the the linear trend are superposed and this gives rise to the rms fluctuation function denoted as $F_{\eta L}(n)$. In fig 5.3 below, fig 5.4, fig 5.5, DFA results for correlated noise, linear trend and superposition of linear trend and correlated noise respectively are shown. The correlated noise is generated using the algorithm of Makse et. all. [71] (the details of the model and the mathematica code is given in Appendix A.)

In fig 5.5 a crossover is seen at log(n)$\approx$ 3, which will be referred as n_x. For $n < n_x$ the slope is nearer to the slope of the correlated noise. For $n > n_x$ the slope is nearer to the slope of linear trend. The crossover is not due to any intrinsic change in dynamics.

5.3.2. Noise with Sinusodial Trend

It has been shown in [57] that when a sinusoidal trend $u(i) = A_\mathrm{S} \sin(2\pi i/T)$ $(i = 1, ..., N_{max})$ is taken, where A_S is the amplitude of the signal and T is the period, the rms fluctuation function $F_\mathrm{S}(n)$, has the same shape for different amplitudes and different periods. Below in fig 5.6 a typical behavior is shown with $A_\mathrm{S} = 2$ and $T = 2^{12}$. A crossover is observed at a scale $n_{2\times} \approx T$.

Table 5.1. The slopes of crossover regions mentioned above. × indicate that there is no such region. For instance, for $log(F_{\eta}(n))$ there is only one region hence $n < n_{2\times}$ and $n < n_{3\times}$ is represented by a ×

crossover regions	slope of $log(F_{\eta}(n))$	slope of $log(F_{\mathrm{S}}(n))$	slope of $log(F_{\eta\mathrm{S}}(n))$
$n < n_{1\times}$	0.89	2	2
$n < n_{2\times}$	×	horizontal line	horizontal line
$n < n_{3\times}$	×	×	0.8

For $n < n_{2\times}$, the rms fluctuation $F_{\mathrm{S}}(n)$ exhibits a scaling with an exponent equal to 2. This behavior is due to the fact that at small scales (box size n) the sinusoidal function is dominated by a linear term. When $n > n_{2\times}$, $F_{\mathrm{S}}(n)$ is a constant independent of the scale n because of the periodic property of the sinusoidal trend.

When the DFA method is applied to the a superposition of correlated noise with a sinusoidal trend, three crossovers in the rms fluctuation $F_{\eta\mathrm{S}}(n)$ at characteristic scales denoted by $n_{1\times}$, $n_{2\times}$ and $n_{3\times}$ are observed (see fig. 5.7).

The first and third crossovers at scales $n_{1\times}$ and $n_{3\times}$ seem to result from the competition between the effects on $F_{\eta\mathrm{S}}(n)$ of the sinusoidal signal and the correlated noise. For $n < n_{1\times}$ the linear part of the sinusoidal signal is effective and for $n > n_{3\times}$, the noise has the dominating effect ($F_{\eta}(n) > F_{\mathrm{S}}(n)$) (see fig 5.8 for the behavior of $F_{\eta}(n)$), and the behavior of $F_{\eta\mathrm{S}}(n)$ is very close to the behavior of $F_{\eta}(n)$. For $n_{1\times} < n < n_{2\times}$ and $n_{2\times} < n < n_{3\times}$ the sinusoidal trend dominates ($F_{\mathrm{S}}(n) > F_{\eta}(n)$), and the behavior of $F_{\eta\mathrm{S}}(n)$ is close to $F_{\mathrm{S}}(n)$. A summary of the values is presented in table 5.1.

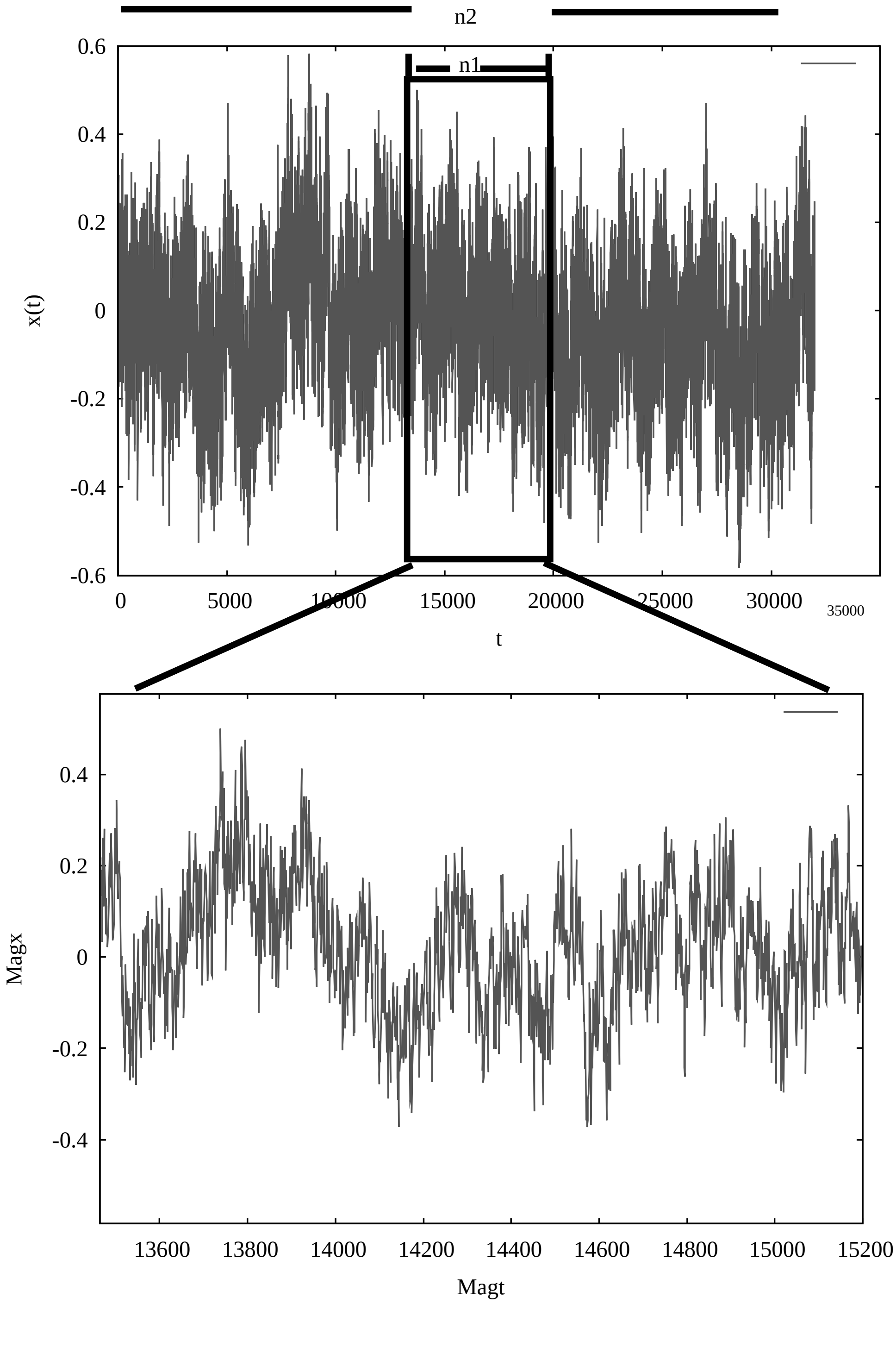

Figure 5.2. Two observation windows, with time scales n_1 and n_2. The lower picture is the magnification of the smaller window with time scale n_1 where the fluctuations look similar

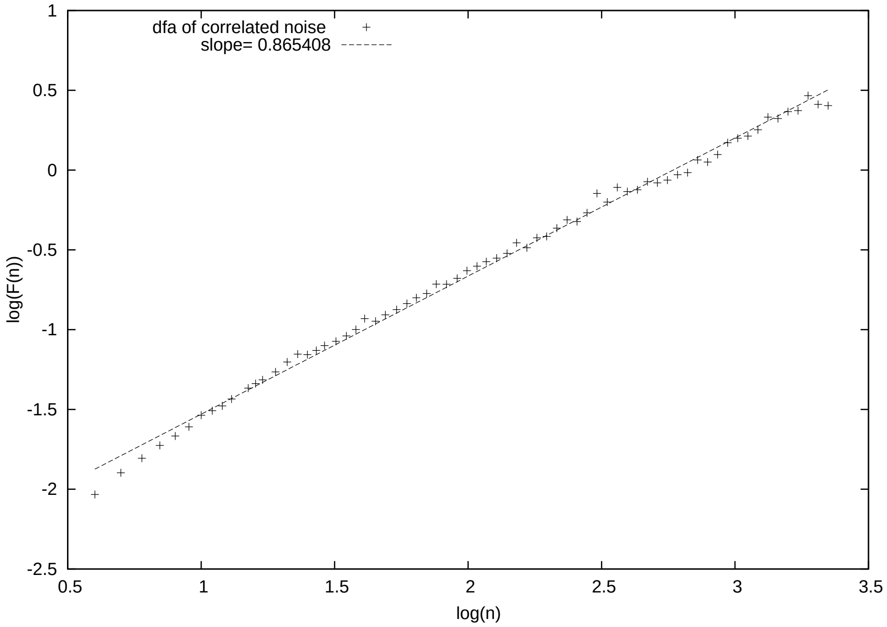

Figure 5.3. DFA analysis of correlated noise *alpha* is 0.86

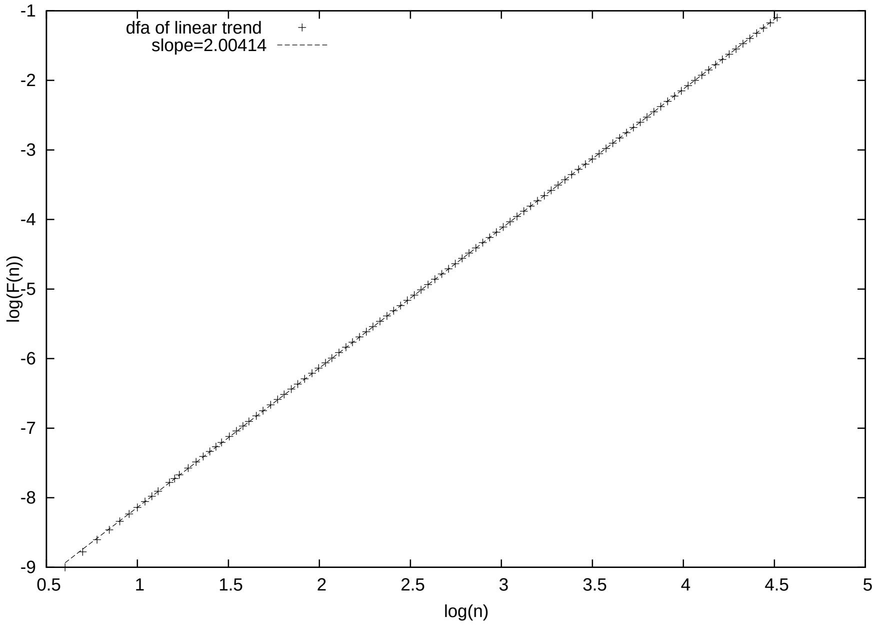

Figure 5.4. DFA analysis of linear trend *alpha* is 2

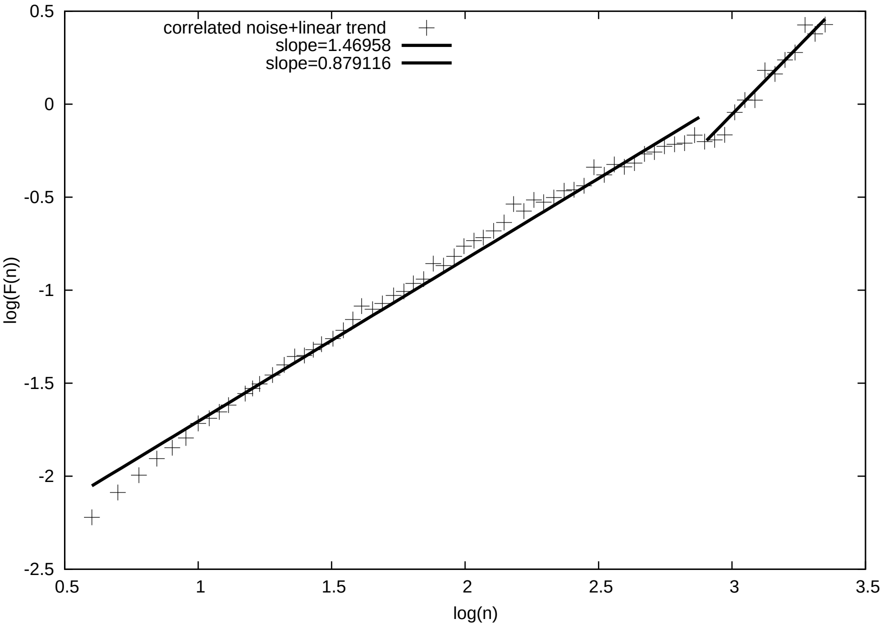

Figure 5.5. DFA analysis of superposition of linear trend and correlated noise: The crossover is at log(n)=3

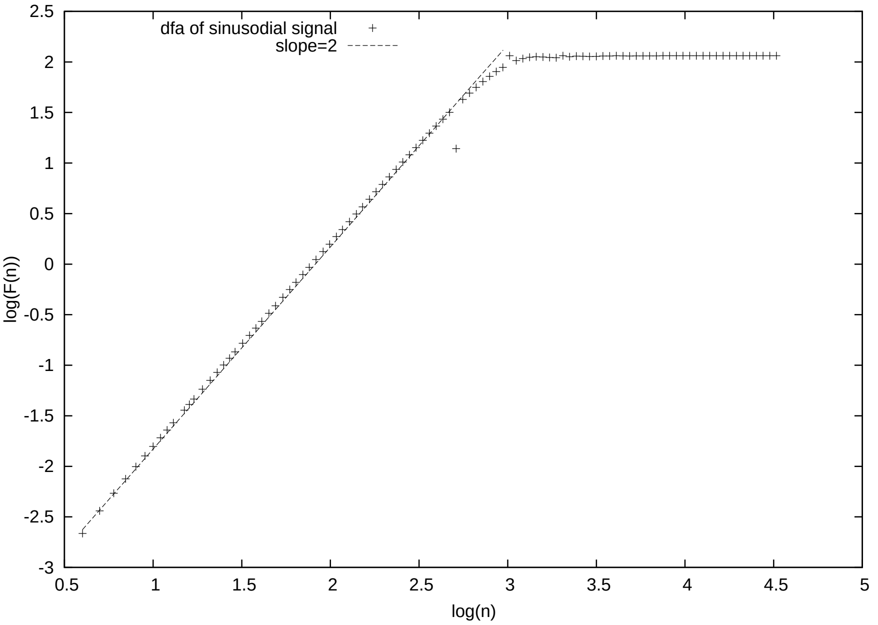

Figure 5.6. DFA analysis of a sinusoidal trend: The crossover can be clearly seen, the slope of the first line is 2

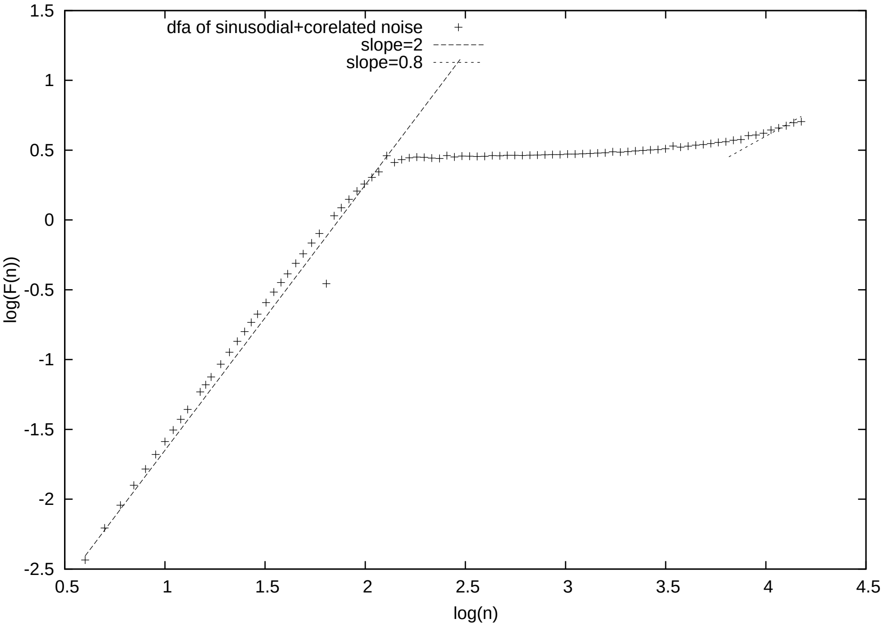

Figure 5.7. DFA analysis of a superposition of correlated noise with a sinusoidal trend: Three crossovers can be observed with slopes 2,almost horizontal,0.8

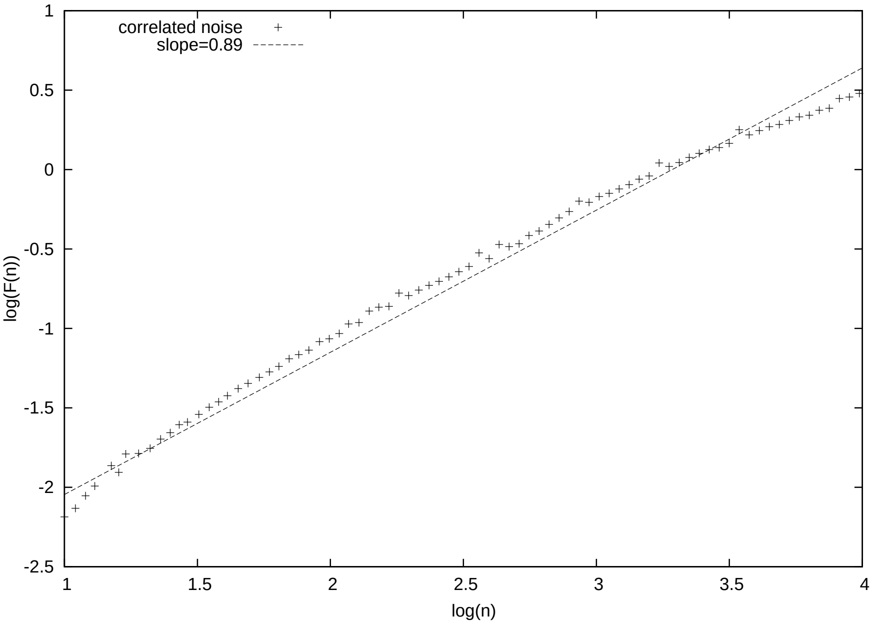

Figure 5.8. DFA analysis of correlated noise with a slope of 0.89

6. SIMULATION

The competition between the quenched random field and the elastic restoring forces invokes some very complex dynamics. Charge density waves provide one of the very few cases where it is possible to study the effects of quenched randomness due to impurities in such systems.

The recursion relation given by equation 4.4 is used as the CDW model. It is numerically simulated, and the polarization current has been computed using the values of phases obtained.

The simulated model has then been fitted to the experimentally observed transient current through PMMA. The simulation and the fitting codes are written in the Fortran programming language. The data and the simulation for the transient current are normalized between 0 and 1. There are four parameters, the number of the impurity sites, the number of time steps, the values of B and ξ that control the behavior of the CDW polarization current. Initial conditions are chosen randomly, that is, the code fills the initial values of the impurity sites with quenched random values which have values between 0 and 1. N (the number of time steps), the number of states (which is set to 10000 as in [9]) are given manually to the code. Of the two parameters, ξ represents the forcing electric field and B represents the diffusive coupling. A constant value for the former fits the experimental data satisfactorily.

For a satisfactory fit, B has to be taken as a time dependent parameter. For simplicity, the following approach is used. The data is split into time intervals of the order of the first minimum of the average mutual information.

Mutual information gives an estimate for the information connection between data values. The CERN MINUIT package [72, 73] is used to find the best fit for an overall ξ and B for each interval.

Figures 6.1, 6.2, 6.3 show the simulation results and the data for an applied voltage of 3.25MV/m, 2.50MV/m and 2.0MV/m respectively.

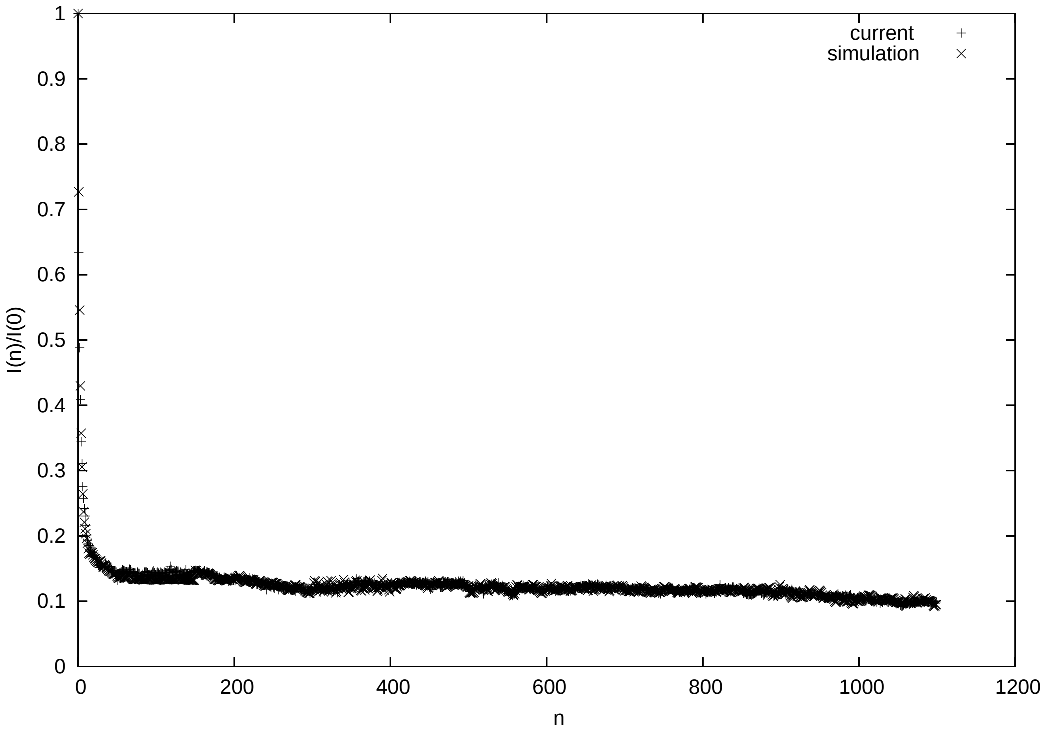

Figure 6.1. CDW fit of current at 3.25MV/m. Both the measured current(+) and simulated CDW polarization current(×) are normalized and drawn versus number of time steps (n)

The fit required three different constant values for the parameter B for different regions of time. In order to verify this observation that more than one different regions of B are indicated, detrended fluctuation analysis (DFA)is used on the observed time series, assuming that a change in the dynamics of the system would involve a change in the pinning and hence B.

Figures 6.6, 6.7, 6.8 show log(F(n))vs log(n) for 3.25MV/m, 2.50MV/m and 2.0 MV/m respectively both for the current and the simulation. Although a precise identification of power law behavior would require more decades of data, a discontinuity in slopes clearly indicates three regions for the 3.25 MV/m data. Although a linear

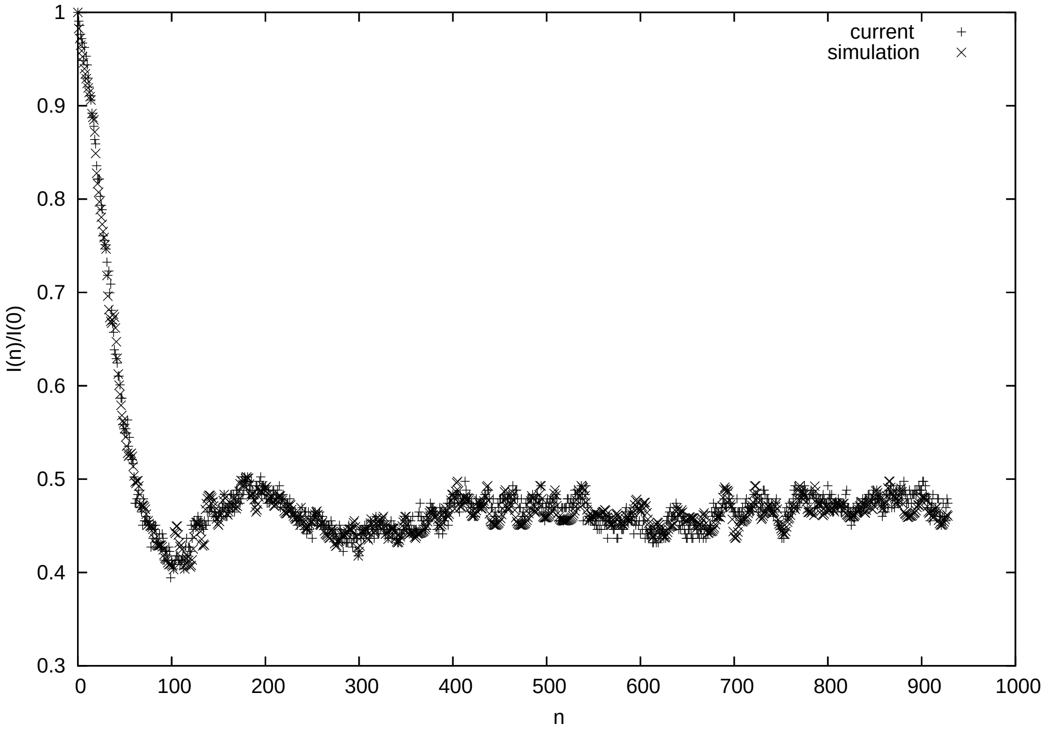

Figure 6.2. CDW fit of current at 2.50MV/m (current (+),simulation (×))

log-log plot is less clearly indicated for a large box size for the next lowest voltage of 2.50MV/m, three regions can still be identified. In this set one can observe a middle region with slope nearly equal to 2 followed by a crossover, which, according to [57] might be caused by a sinusoidal trend rather then a change in the dynamical properties of the system. The similarity between the crossover regions in the data and the simulation leads us to conclude that the crossover is more likely to result from a transition in the dynamical properties of the system. The process under study is basically diffusive and damped with chaotic fluctuations; the pinning parameter B is indirectly related to the damping because of the scaled time-like parameter in the CDW model used in this work. Moreover, in [6], this data set is reported to have the minimum maximal Lyapunov exponent with respect to other data sets, which might explain why this data set (Figure 6.2) seems to be more periodic-like than other sets. The power spectrum for the 2.0 MV/m data is shown in Figure 6.4 does not show any marked indication of periodic behavior. One can also see a crossover behavior from a region with slope near 2 similar to that in the data when DFA analysis is applied to a numerical solution of

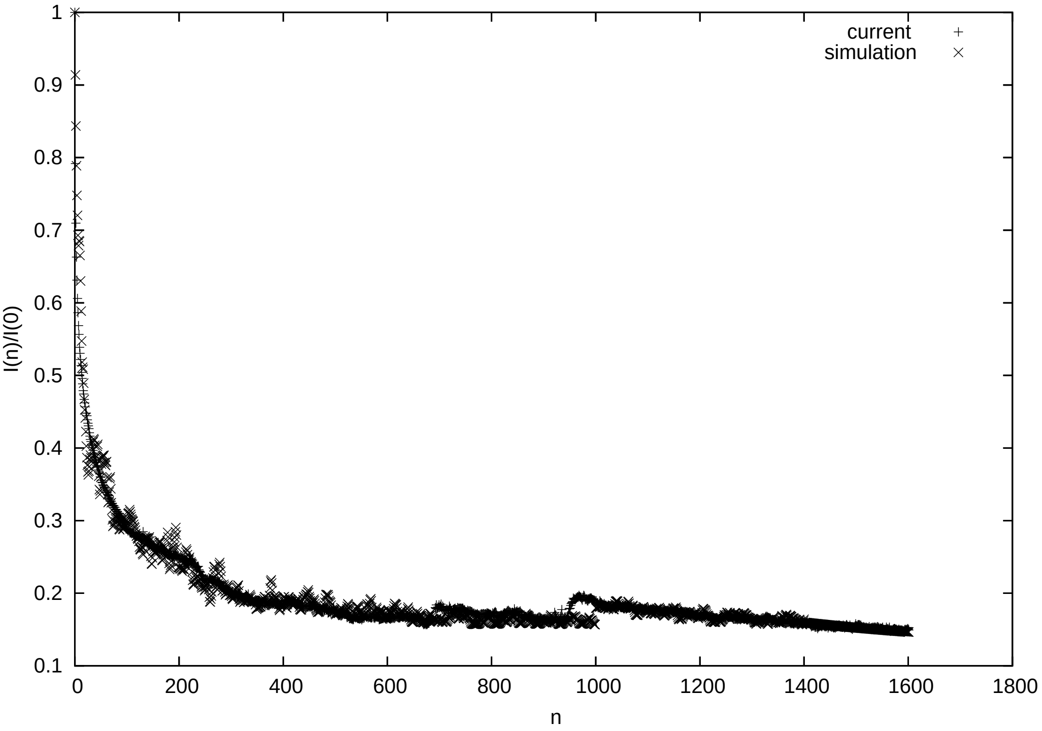

Figure 6.3. CDW fit of current at 2.0MV/m (current (+), simulation (×))

Lorenz equations (for the Lorenz Equations see Appendix). This is shown in Figure 6.5. For the lowest voltage of 2.0 MV/m the demarcation of regions is less clear but plausible.

Table 1. summarizes the results of the fit and the DFA scaling exponents. The three different regions with the corresponding values of B and the values for the scaling exponents are shown. The variation of the scaling exponent at the weakest applied electric field among regions is less pronounced. For stronger applied electric fields, the values of B for the first two regions seem to show a significantly slower variation, but attempting a single overall value deteriorates the fit. It is also observed that as the voltage increases the long range correlations (indicated by α) decrease, and the regime changes become more distinct. This behavior seems to be consistent with the strong dependence of conductivity on the electric field[2]. The field dependence is mostly attributed to trapping of free charge carriers in the volume of the dielectric during their motion due to the applied field, whereas release from traps was considered to be thermally activated with a field-modified activation energy [74]. These charges are believed to be trapped in some of the pinning states arising from the defects in the structure of the insulator such as impurities, dopants and dangling bonds. It has been suggested that side groups may act in a way similar to doped impurities [29]. The variation of B seems to be related with the competition between the mechanisms of conductivity in PMMA via dipole relaxation process, caused by heterogeneities and trapping of charge carriers [2]. It is known that dipole relaxation completes before other mechanisms [75]. This is probably related to the first crossover in the scaling exponent. The other two crossovers in α might be due to relaxation caused by heterogeneities and trapping of charge carriers.

Another observation that supports the existence of three regimes reflects itself in the autocorrelation function parallel to the results obtained in DFA analysis. In figure 6.9 a crossover behavior having what seems to be three different slopes can be seen.

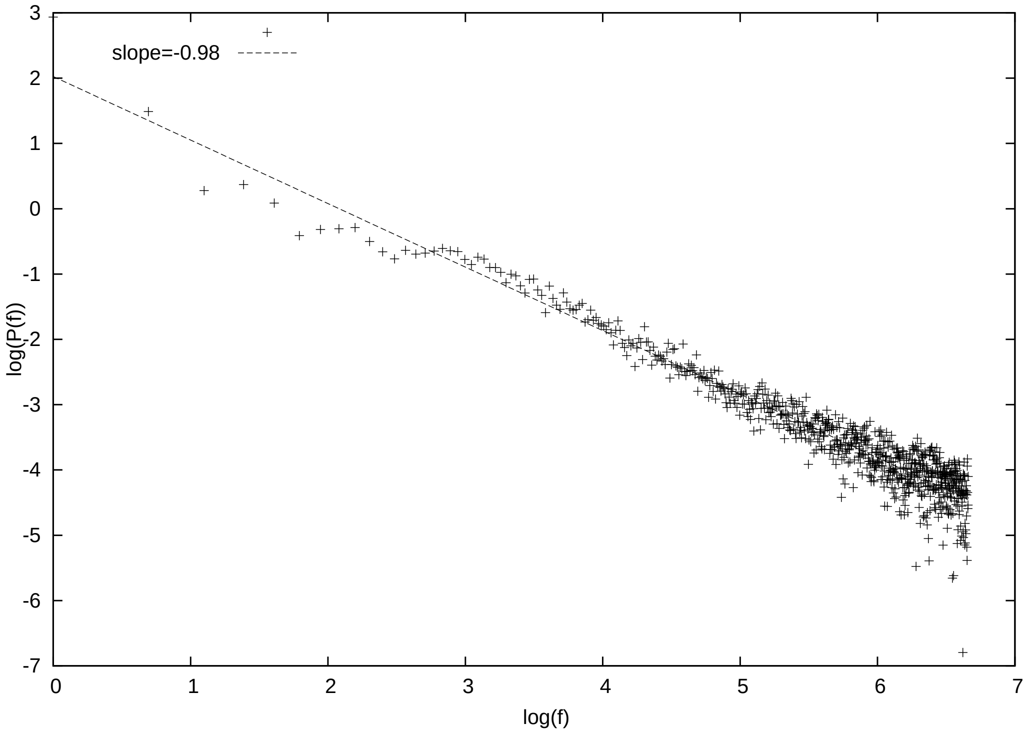

Figure 6.4. Power spectrum analysis of the observed current at 2 MV/m. The slope of the straight line is equal to -0.98

Table 6.1. The scaling exponents, the B values, and the ξ values for the three applied electric fields

	α_1	α_2	α_3	B_1	B_2	B_3	ξ
3.25MV/m	1.47	0.73	1.43	1.3±0.2	1.4±0.2	1.3±0.2	0.7
2.50 MV/m	1.45	1.98	0.5	1.3±0.1	1.4±0.1	1.7±0.1	0.53
2.0MV/m	1.69	1.47	1.3	0.8±0.05	0.7±0.05	0.8±0.05	0.62

A similar situation can also be seen in figures 6.10 and 6.11 for 3.25MV/m and 2.50 MV/m respectively.

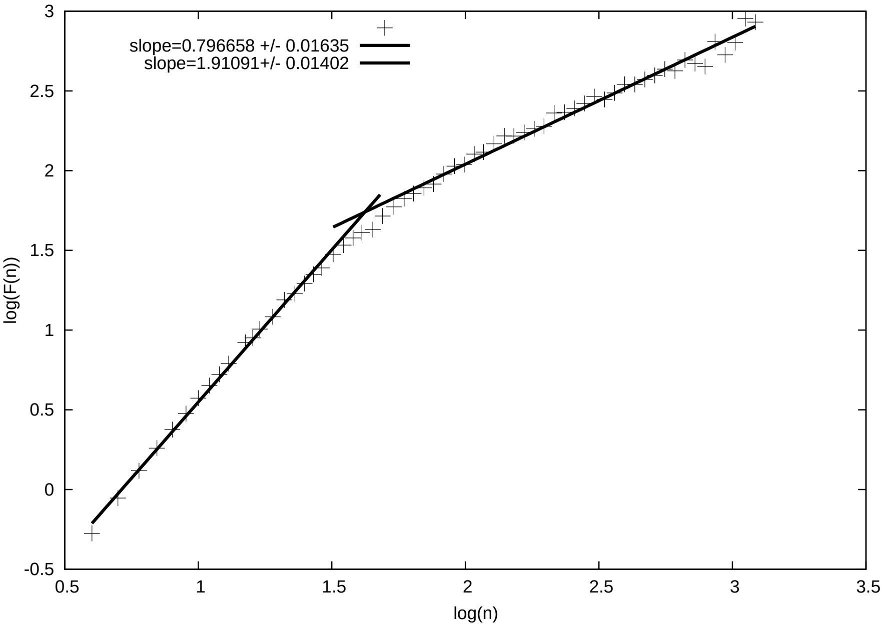

Figure 6.5. Plot of log F(n) vs. log n for time series of the x component of Lorenz equations showing the crossover

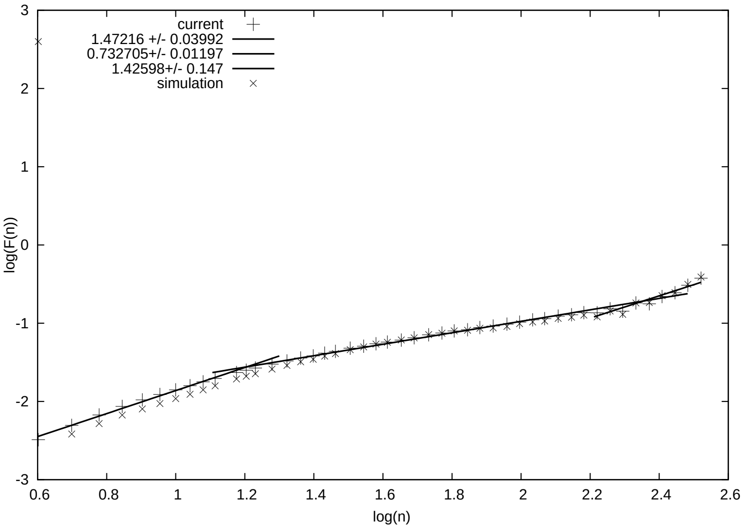

Figure 6.6. log(F(n)) vs log(n) with three different scaling exponents for the measured current (+) at 3.25MV/m and the simulation (x).The inset shows the scaling exponents for the current

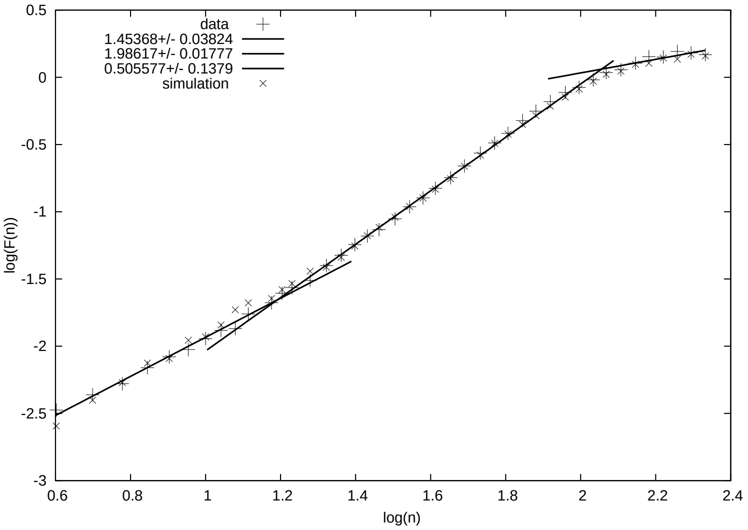

Figure 6.7. log(F(n)) vs log(n) with three different scaling exponents for the measured current (+) at 2.50MV/m and the simulation. The inset show the scaling exponents for the current

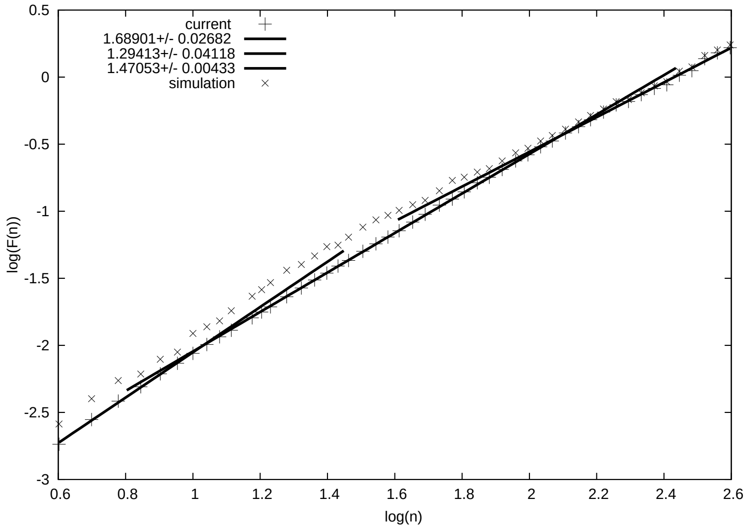

Figure 6.8. log(F(n)) vs log(n) with three different scaling exponents for the measured current (+) at 2.00MV/m and the simulation(x). The inset show the scaling exponents for the current

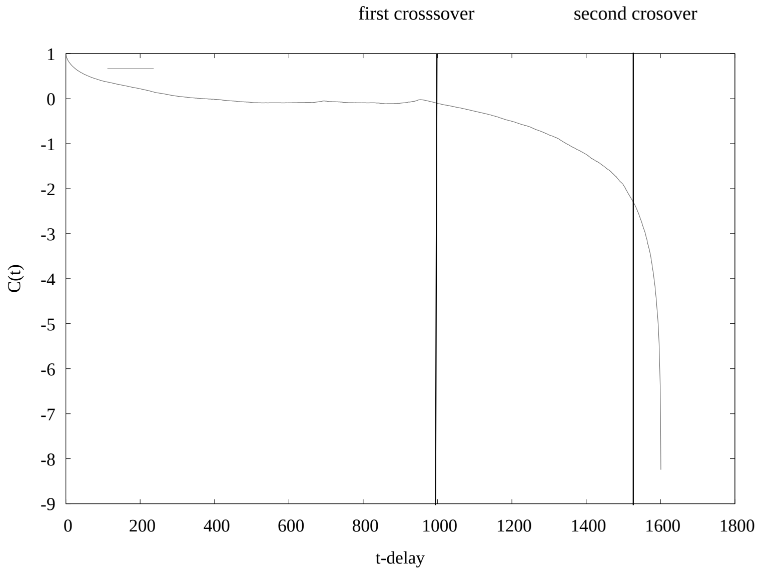

Figure 6.9. Autocorrelation function vs delay time with three different regions for 2.00MV/m

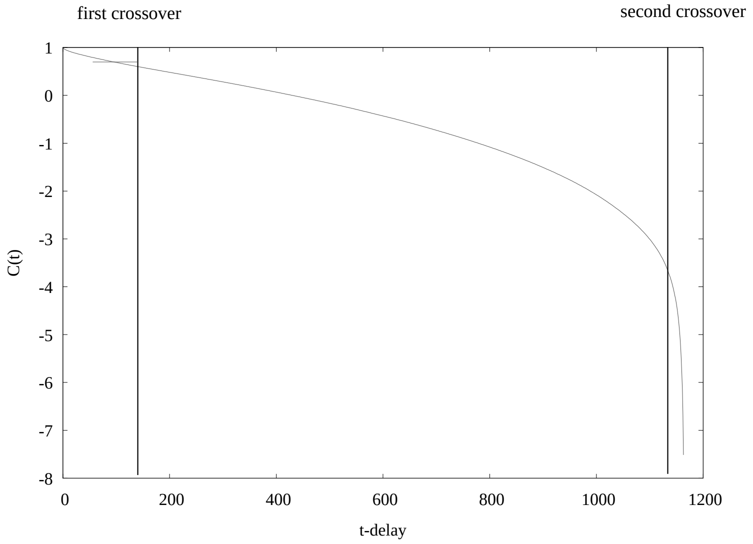

Figure 6.10. Autocorrelation function vs delay time with three different regions for 3.250MV/m

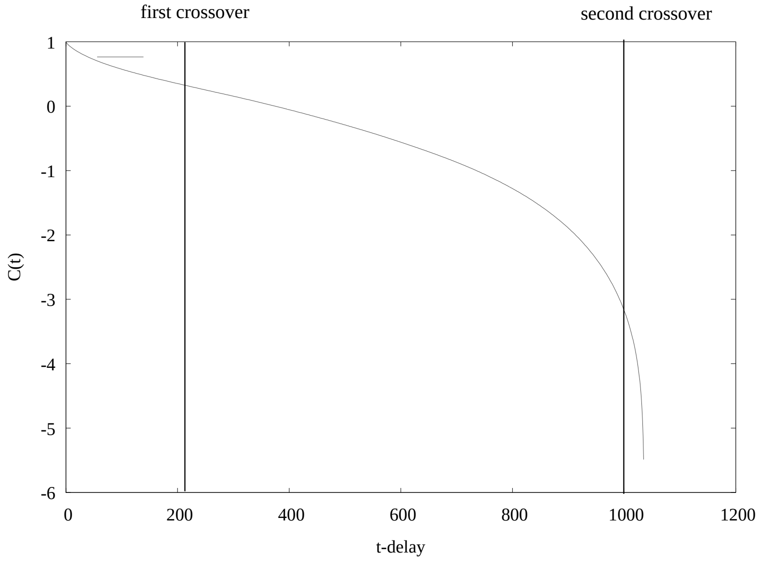

Figure 6.11. Autocorrelation function vs delay time with three different regions for 3.00MV/m

7. CONCLUSIONS

In this work, an attempt has been made to simulate the experimentally observed polarization decay current in terms of a dynamical system with a diffusive coupling that is piecewise continuous in time. The discontinuities indicate the presence of three different time dependent nonlinear conductivity mechanisms. This presence is supported by a detrended fluctuation analysis. An overall positive Lyapunov exponent had been observed and reported elsewhere [5, 6]. An attempt has been made to model the indicated chaotic behavior in terms of a model describing pinned charge density waves. Charge density waves (of polarons) have been reported in certain polymers[38], but not yet in PMMA (even though the existence of polarons is known [39]) . A satisfactory fit to the model has been obtained; however it is not absolutely clear whether this is a pure charge density wave effect, or there are further interactions which can still be modelled by the CDW model by admitting a piecewise time dependent diffusive coupling. Piecewise time dependence agrees with the DFA results. One reason for the time dependence of the diffusive coupling may be aging under the influence of applied electric field (for aging effects in PMMA see [1, 3]). The continued application of an electric field is affecting the configuration of the polymer. Thus the heavily pinned charge density wave model stands as a good candidate for modelling the transient current, in spite of the fact that charge density waves have been observedin other polymers but not PMMA.

APPENDIX A: Generating Correlated noise

The algorithm for generating correlated noise series (η_i)can be summarized as follows [76, 77]:

(I) Generate a one-dimensional sequence r_i of uncorrelated random numbers with a Gaussian distribution, and calculate the Fourier transform coefficients of the series r_q.

(II) Calculate the Fourier transform of the correlated random series (η_q) using

$$S(q) =< \eta_q \eta_{-q} > \tag{A.1}$$

$$\eta_q = \sqrt{S(q)} r_q \tag{A.2}$$

where S(q) is the Fourier transform of C(l) which is the long-range power-law correlation function of the generated series of the form

$$C(l) =< \eta_i \eta i + l >\sim l^{-\gamma} \tag{A.3}$$

and γ is the correlation exponent.

(III) Calculate the inverse Fourier transform of η_q to obtain the correlated series of η_i .

The modifications made by Makse et. al [71] is as follows:

(I) The correlation function $C(l)$ is modified such as $C(l) \equiv (1 + l^2)^{\frac{\gamma}{2}}$

(II) $C(l)$ is defined on the interval $[\frac{-L}{2}, ..., \frac{L}{2}]$

(III) It is shown that the Fourier transform of the redefined correlation function can be calculated as follows:

$$S(q) = \frac{2\pi^{\frac{1}{2}}}{\Gamma(\frac{\gamma}{2} + \frac{1}{2})} (\frac{q}{2})^{\frac{\gamma-1}{2}} \kappa_{\frac{\gamma-1}{2}}(q) \tag{A.4}$$

Below is the Mathematica source code used for the generation of the correlated noise series used in this work.

```
Needs["Statistics‘ContinuousDistributions‘"]
RandomNormal[m_,s_]:=Random[NormalDistribution[m,s]]
 Do[y[i]=RandomNormal[0,1],{i,1000}]
  Array[y, 1000]>>yout.txt
 data= Array[y, 1000]
 fdata=Abs[Fourier[data]]
 fdata>>foury.txt
 g=0.4
 0.4
sq=Array[x,1000]
sq*fdata
Abs[InverseFourier[Sqrt[sq]*fdata]]>>correlatednoise3.dat
```

APPENDIX B: LORENZ EQUATIONS

The Lorenz system is one of the most widely used examples for chaotic behavior. It is represented by the following equations:

$$\dot{x} = \sigma(y - x) \tag{B.1}$$

$$\dot{y} = rx - y - xz \tag{B.2}$$

$$\dot{z} = xy - bz \tag{B.3}$$

with the parameters r, σ and b. With $\sigma = 10$ and $b = \frac{8}{3}$, the evolution of the Lorenz sysytem with respect to variation in r can be summarized as follows:

(I) For $r < 1$, the system has only one attracting fixed point. It is the origin $c_0 = (0, 0, 0)$ of the state space.

(II) At $r = 1$ the fixed point c_0 loses its stability. c_0 becomes a repellor and the system gets two new fixed points, $c_+ = (\sqrt{(b(r-1))}, \sqrt{(b(r-1)), r-1})$ and $c_+ = (-\sqrt{(b(r-1))}, -\sqrt{(b(r-1)), r-1})$.

(III) Above $r = 24.06$ the system becomes a strange attractor (see fig. B.1) .

The time series of the x component for the above mentioned values of σ, b and $r = 25$ is illustrated in figB.2.

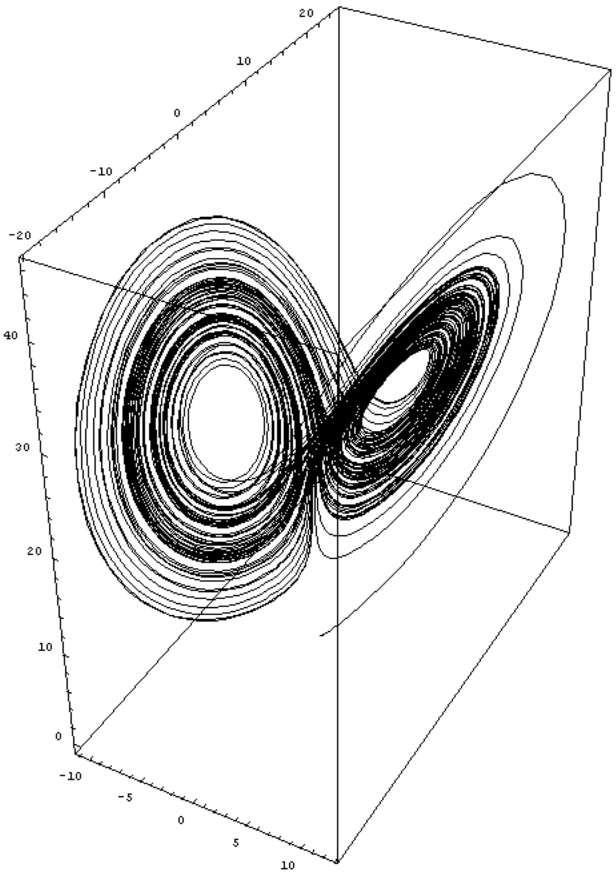

Figure B.1. The Lorenz attractor

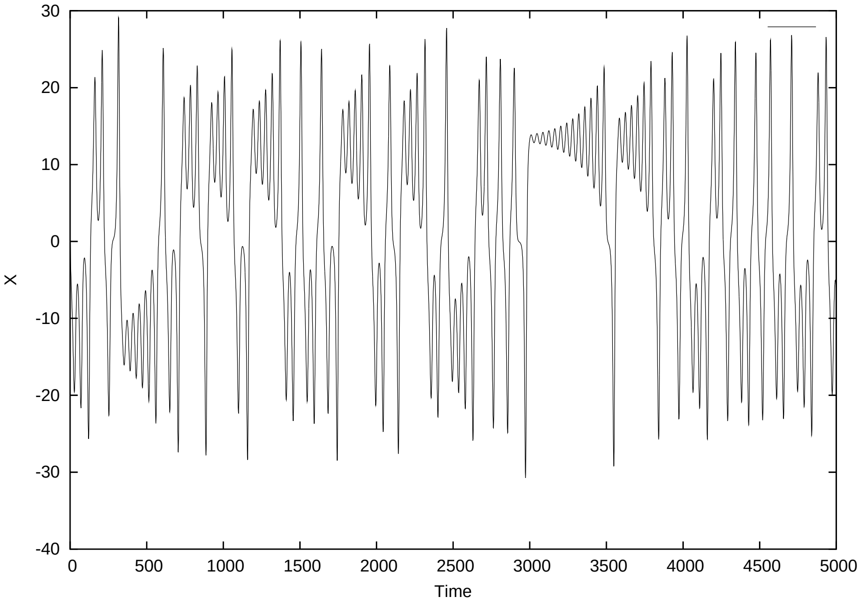

Figure B.2. The time evaluation of x component of Lorenz system

APPENDIX C: SOURCE CODE FOR CHARGE DENSITY WAVE FIT

The code first fills the array representing the impurity sites randomly. Then calculates the current as explained in the section charge density wave function.

```
     function rn(i)
          use Ziggurat

    implicit real*8(a-h,o-z)
c   T his assures that each MINUIT variation gets the same random
numbers
    common /rndms/norder,irnd,yarr(20100)
    if(irnd. eq.0) then
      do i=1,20100
      yarr(i)=uni()
      enddo

    irnd=1
    endif
    if(norder.eq.0) norder=1
        rn=yarr(norder)
    norder=norder+1
      return
    end

 Subroutine Fitdata(x,curr,nstart,NTIME,xmaxbin,xminbin,DT,iflag)
 \\C This Sub Routine calculates simulated time development
 use Ziggurat
      implicit real*8(a-h,o-z)
      implicit integer*4(i-n)
      COMMON/STATPS/ nstat, npos, ntime1
    common /rndms/norder,irnd,yarr(20100)
    parameter(NPOSM=10008,NTIMEM=3008,NSTATM=1)
    dimension Y(NPOSM,NTIMEM),current(NTIMEM),H(NPOSM),
    YPRIME(NPOSM,NT1IMEM)


      dimension X(*),CURR(*)

      TWOPI=8.0D0*ATAN(1.0d0)
    B=X(1)
    E=X(2)
      DT=2.0d-1
    write(*,*)xmaxbin,xminbin
c   write(*,*)'%%%',x(1),x(2)
c   open(unit=10, file='simcur.dat', status='unknown')

c     set the initial conditions
    norder=0
           do i=1,npos
             y(i,1)=rn(1)
           enddo

           do i=1,npos
        H(I)=rn(1)
           enddo

c    calculate YPRIME(NPOS,1) c   write(*,*)'+',ntime,npos

                 DO J=2,NTIME+7
                     DO I=1,NPOS
```

```
c   each impurity site is assumed to be randomly away
Y(I,J+1)=(1.0D0-2.0d0*B*DT)*Y(I,J)+DT*B*(Y(I-1,J)+Y(I+1,J))
+E*DT-.DT*DSIN(TWOPI*(H(I)+Y(I,J)))
                      ENDDO
                      ENDDO

                  DO J=1,NTIME+7
                     DO I=1,NPOS
c  each impurity site is assumed to be randomly away
YPRIME(I,J)=B*(Y(I+1,J)+Y(I-1,J)-2*Y(I,J))-DSIN(MOD(TWOPI*(H(I)
+Y(I,J)),TWOPI))+E



                      ENDDO
                      ENDDO
c  write(,*)"current (j)=a/npos"
    xmax=-1000.0
    xmin=1000.0
             DO J=1,NTIME+7
                A=0
             DO I=1,NPOS
                  A=A+YPRIME(I,J)
              enddo

             current(J)=A/NPOS
c  if(current(J).le.xmin)xmin=current(J) c
if(current(J).ge.xmax)xmax=current(J)

c  write(*,*) current(J)
             enddo
c  write(*,*)'ebebe'
    xmax=-1000.0
    xmin=1000.0

c  write(*,*)'ebebe2',xmin,xmax
             DO im=nstart,NTIME
c   CURR(I)=X(3)*current(i)


c   CURR(I)=(current(i+6)-xmin)/(xmax-xmin)


        curr(im)=current(im+6)
c   write(*,*)'!!',im+6,xmax,xmin,current(im+6)-xmin


      ENDDO
    do ime=nstart,ntime
    if(curr(ime).le.xmin)xmin=curr(ime)
     if(curr(ime).ge.xmax)xmax=curr(ime)
    end do
    do imec=nstart,ntime
    s=0
    s=curr(imec)

 curr(imec)=((xmaxbin-xminbin)/(xmax-xmin))*(s-xmin)+xminbin
    end do

     write (*,*) 'getting out of fit'
    RETURN
      end
```

APPENDIX D: GENERATOR FOR RANDOM NUMBERS

This random number generator is used in order to obtain uniformly distributed random numbers between 0 and 1. The reasons as to why this code is used are, first it is free software and second most of the random number generators created after give reference to this software.

```
! Marsaglia \& Tsang generator for random normals \& random
exponentials. ! Translated from C by Alan Miller
(amiller@bigpond.net.au)

! Marsaglia, G. \& Tsang, W.W. (2000) ‘The ziggurat method for
generating ! random variables’, J. Statist. Software, v5(8).

! This is an electronic journal which can be downloaded from: !
http://www.jstatsoft.org/v05/i08

! N.B. It is assumed that all integers are 32-bit. ! N.B. The
value of M2 has been halved to compensate for the lack of !
unsigned integers in Fortran.

! Latest version - 1 January 2001
  MODULE Ziggurat
   IMPLICIT NONE

   PRIVATE

INTEGER,  PARAMETER  ::  DP=SELECTED_REAL_KIND( 12, 60 ) REAL(DP),
PARAMETER  ::  m1=2147483648.0_DP,   m2=2147483648.0_DP,     \&
                            half=0.5_DP
```

```
REAL(DP)             ::  dn=3.442619855899_DP,
tn=3.442619855899_DP,    \&
 vn=0.00991256303526217_DP,\&
q, de=7.697117470131487_DP, \& te=7.697117470131487_DP, \&
\\
ve=0.003949659822581572_DP INTEGER,  SAVE       ::  iz, jz,
jsr=123456789, kn(0:127), \&
ke(0:255), hz REAL(DP), SAVE       ::
wn(0:127), fn(0:127), we(0:255), fe(0:255) LOGICAL,  SAVE       ::
initialized=.FALSE.

PUBLIC  :: zigset, shr3, uni, rnor, rexp


CONTAINS



SUBROUTINE zigset( jsrseed )

   INTEGER, INTENT(IN)  :: jsrseed

   INTEGER  :: i

   !  Set the seed
   jsr = jsrseed

   !  Tables for RNOR
   q = vn*EXP(half*dn*dn)
   kn(0) = (dn/q)*m1
   kn(1) = 0
   wn(0) = q/m1
   wn(127) = dn/m1
   fn(0) = 1.0_DP
   fn(127) = EXP( -half*dn*dn )
   DO  i = 126, 1, -1
      dn = SQRT( -2.0_DP * LOG( vn/dn + EXP( -half*dn*dn ) ) )
      kn(i+1) = (dn/tn)*m1
      tn = dn
      fn(i) = EXP(-half*dn*dn)
      wn(i) = dn/m1
   END DO

   !  Tables for REXP
   q = ve*EXP( de )
   ke(0) = (de/q)*m2
   ke(1) = 0
   we(0) = q/m2
   we(255) = de/m2
   fe(0) = 1.0_DP
   fe(255) = EXP( -de )
   DO  i = 254, 1, -1
      de = -LOG( ve/de + EXP( -de ) )
      ke(i+1) = m2 * (de/te)
      te = de
      fe(i) = EXP( -de )
      we(i) = de/m2
   END DO
   initialized = .TRUE.
   RETURN
END SUBROUTINE zigset



!  Generate random 32-bit integers FUNCTION shr3( ) RESULT( ival )
```

```
   INTEGER  ::  ival

   jz = jsr
   jsr = IEOR( jsr, ISHFT( jsr,  13 ) )
   jsr = IEOR( jsr, ISHFT( jsr, -17 ) )
   jsr = IEOR( jsr, ISHFT( jsr,   5 ) )
   ival = jz + jsr
   RETURN
END FUNCTION shr3



!  Generate uniformly distributed random numbers FUNCTION uni( )
RESULT( fn_val )
   REAL(DP)  ::  fn_val

   fn_val = half + 0.2328306e-9_DP * shr3( )
   RETURN
END FUNCTION uni



!  Generate random normals FUNCTION rnor( ) RESULT( fn_val )
   REAL(DP)             ::  fn_val

   REAL(DP), PARAMETER  ::  r = 3.442620_DP
   REAL(DP)             ::  x, y

   IF( .NOT. initialized ) CALL zigset( jsr )
   hz = shr3( )
   iz = IAND( hz, 127 )
   IF( ABS( hz ) < kn(iz) ) THEN
```

```
      fn_val = hz * wn(iz)
   ELSE
      DO
         IF( iz == 0 ) THEN
            DO
               x = -0.2904764_DP* LOG( uni( ) )
               y = -LOG( uni( ) )
               IF( y+y >= x*x ) EXIT
            END DO
            fn_val = r+x
            IF( hz <= 0 ) fn_val = -fn_val
            RETURN
         END IF
         x = hz * wn(iz)
IF( fn(iz) + uni( )*(fn(iz-1)-fn(iz)) < EXP(-half*x*x) ) THEN
            fn_val = x
            RETURN
         END IF
         hz = shr3( )
         iz = IAND( hz, 127 )
         IF( ABS( hz ) < kn(iz) ) THEN
            fn_val = hz * wn(iz)
            RETURN
         END IF
      END DO
   END IF
   RETURN
END FUNCTION rnor



!  Generate random exponentials FUNCTION rexp( ) RESULT( fn_val )
```

```
   REAL(DP)  ::  fn_val

   REAL(DP)  ::  x

   IF( .NOT. initialized ) CALL Zigset( jsr )
   jz = shr3( )
   iz = IAND( jz, 255 )
   IF( ABS( jz ) < ke(iz) ) THEN
      fn_val = ABS(jz) * we(iz)
      RETURN
   END IF
   DO
      IF( iz == 0 ) THEN
         fn_val = 7.69711 - LOG( uni( ) )
         RETURN
      END IF
      x = ABS( jz ) * we(iz)
      IF( fe(iz) + uni( )*(fe(iz-1) - fe(iz)) < EXP( -x ) ) THEN
         fn_val = x
         RETURN
      END IF
      jz = shr3( )
      iz = IAND( jz, 255 )
      IF( ABS( jz ) < ke(iz) ) THEN
         fn_val = ABS( jz ) * we(iz)
         RETURN
      END IF
   END DO
   RETURN
END FUNCTION rexp

END MODULE ziggurat
```

APPENDIX E: DFA SOURCE CODE

file: dfa.c J. Mietus, C-K Peng, and G. Moody 8 February 2001 Last revised: 25 January 2005 v4.9

dfa: Detrended Fluctuation Analysis (translated from C-K Peng's Fortran code)

Copyright (C) 2001-2005 Joe Mietus, C-K Peng, and George B. Moody

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

You may contact the authors by e-mail (peng@physionet.org) or postal mail (Beth Israel Deaconess Medical Center, Room KS-B26, 330 Brookline Ave., Boston, MA 02215 USA). For updates to this software, please visit PhysioNet (
http://www.physionet.org/
).

This method was first proposed in: Peng C-K, Buldyrev SV, Havlin S, Simons M, Stanley HE, Goldberger AL. Mosaic organization of DNA nucleotides. Phys Rev E 1994;49:1685-1689. [
 Available on-line at
http://prola.aps.org/abstract/PRE/v49/i2/p1685_1
]

A detailed description of the algorithm and its application to physiologic signals can be found in: Peng C-K, Havlin S, Stanley HE, Goldberger AL. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. Chaos 1995;5:82-87. [Abstract online at
http://www.ncbi.nlm.nih.gov/entrez/-

```
query.fcgi?cmd=Retrieve&db=PubMed&list_uids=11538314&dopt=Abstract
```

]

If you use this program in support of published research, please include a citation of at least one of the two references above, as well as the standard citation for PhysioNet: Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. PhysioBank, PhysioToolkit, and Physionet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [

```
Circulation Electronic Pages;
  http://circ.ahajournals.org/cgi/content/full/101/23/e215
```

]; 2000 (June 13).

```
#include <stdio.h> #include <stdlib.h> #include <math.h>

#define SWAP(a,b) {temp = (a); (a) = (b); (b) = temp;}

/* Function prototypes. */ long input(void); int rscale(long
minbox, long maxbox, double boxratio); void dfa(double *seq, long
npts, int nfit, long *rs, int nr, int sw); void setup(void); void
cleanup(void); void help(void); double polyfit(double **x, double
*y, long ndat, int nfit); void error(char error_text[]); double
*vector(long nl, long nh); int *ivector(long nl, long nh); long
*lvector(long nl, long nh); double **matrix(long nrl, long nrh,
long ncl, long nch); void free_vector(double *v, long nl, long
nh); void free_ivector(int *v, long nl, long nh); void
free_lvector(long *v, long nl, long nh); void free_matrix(double
**m, long nrl, long nrh, long ncl, long nch);

/* Global variables. */ char *pname;    /* this program's name
(for use in error messages) */ double *seq;    /* input data
buffer; allocated and filled by input() */ long *rs;   /* box size
array; allocated and filled by rscale() */ double *mse;    /*
```

```
fluctuation array; allocated by setup(), filled by dfa() */ int
iflag = 1;  /* integrate the input data if non-zero */ int nfit =
2;   /* order of the regression fit, plus 1 */ int nr;     /*
number of box sizes */

main(int argc, char **argv) {
  int i, sw = 0;
  long minbox = 0L, maxbox = 0L, npts, temp;

  /* Read and interpret the command line. */
  pname = argv[0];
  for (i = 1; i < argc && *argv[i] == ’-’; i++) {
   switch(argv[i][1]) {
    case ’d’:   /* set nfit (the order of the regression fit) */
   if ((nfit = atoi(argv[++i])+1) < 2)
     error("order must be greater than 0");
   break;
  case ’i’:   /* input data are already integrated */
   iflag = 0; break;
  case ’l’:   /* set minbox (the minimum box size) */
   minbox = atol(argv[++i]); break;
  case ’u’:   /* set maxbox (the maximum box size) */
   maxbox = atol(argv[++i]); break;
  case ’s’:   /* enable sliding window mode */
   sw = 1; break;
    case ’h’:   /* print usage information and quit */
    default:
   help();
   exit(1);
   }
  }
```

```
    /* Allocate and fill the input data array seq[]. */
    npts = input();

    /* Set minimum and maximum box sizes. */
    if (minbox < 2*nfit) minbox = 2*nfit;
    if (maxbox == 0 || maxbox > npts/4) maxbox = npts/4;
    if (minbox > maxbox) {
    SWAP(minbox, maxbox);
    if (minbox < 2*nfit) minbox = 2*nfit;
    }

/* Allocate and fill the box size array rs[]. rscale's third
argument specifies that the ratio between successive box
sizes is
2^(1/8). */
    nr = rscale(minbox, maxbox, pow(2.0, 1.0/8.0));

 /* Allocate memory for dfa() and the functions it calls. */
    setup();
 /* Measure the fluctuations of the detrended input data at
 each box size using the DFA algorithm; fill mse[] with these
  results. */
    dfa(seq, npts, nfit, rs, nr, sw);

    /* Output the results. */
    for (i = 1; i <= nr; i++)
    printf("%g %g\n", log10((double)rs[i]), log10(mse[i])/2.0);

    /* Release allocated memory. */
    cleanup();
    exit(0);
}
```

```
/* Read input data, allocating and filling seq[], integrating if
iflag != 0. Following the convention used for other arrays in this
program, seq[0] is unused, and the first point is stored in
seq[1]. The return value is the number of points read.

This function allows the input buffer to grow as large as
necessary, up to the available memory (assuming that a long int is
large enough to address any memory location). Note that the
integration is done using double precision arithmetic to avoid
complete loss of precision when the integrated data reach large
amplitudes.  */
long input() {
    long maxdat = 0L, npts = 0L;
    double y, yp = 0.0;

    while (scanf("%lf", &y) == 1) {
        if (++npts >= maxdat) {
        double *s;

        maxdat += 50000;
/* allow the input buffer to grow (the increment is arbitrary) */
        if ((s = realloc(seq, maxdat * sizeof(double))) == NULL) {
        fprintf(stderr,
"\%s: insufficient memory, truncating input at row \%d\n",
              pname, npts);
            break;
        }
        seq = s;
    }
    seq[npts] = iflag ? (yp += y) : y;
    }
    if (npts < 1) error("no data read");
    return (npts);
}

int rslen;  /* length of rs[] */

/* rscale() allocates and fills rs[], the array of box sizes used
by dfa() below. The box sizes range from (exactly) minbox to
(approximately) maxbox, and are arranged in a geometric series
such that the ratio between consecutive box sizes is
(approximately) boxratio. The return value is the number of box
sizes in rs[]. */
int rscale(long minbox, long maxbox, double
boxratio) {
    int ir, n;
    long rw;

 /* Determine how many scales are needed. */
    rslen = log10(maxbox / (double)minbox) / log10(boxratio) + 1.5;
 /* Thanks to Peter Domitrovich for pointing out that a previous
 version of the above calculation undercounted the number of
 scales in some  situations. */
    rs = lvector(1, rslen);
for (ir = 1, n = 2, rs[1] = minbox; n <= rslen && rs[n-1] <
maxbox; ir++)
      if ((rw = minbox * pow(boxratio, ir) + 0.5) > rs[n-1])
            rs[n++] = rw;
    if (rs[--n] > maxbox) --n;
    return (n);
}
```

```
double **x; /* matrix of abscissas and their powers, for
polyfit(). */

/* Detrended fluctuation analysis// seq: input data array// npts:
number of input points//nfit: order of detrending (2: linear, 3:
quadratic, etc.)// rs: array of box sizes (uniformly distributed
on log scale)//nr: number of entries in rs[] and mse[]// sw: mode
(0: non-overlapping windows, 1: sliding window)//

This function returns the mean squared fluctuations in mse[]. */

void dfa(double *seq, long npts, int nfit, long *rs, int nr, int
sw) {
    long i, boxsize, inc, j;
    double stat;

for (i = 1; i <= nr; i++) {
  boxsize = rs[i];
 if (sw) { inc = 1; stat = (int)(npts - boxsize + 1) * boxsize; }
else { inc = boxsize; stat = (int)(npts / boxsize) * boxsize; }
 for (mse[i] = 0.0, j = 0; j <= npts - boxsize; j += inc)
  mse[i] += polyfit(x, seq + j, boxsize, nfit);
        mse[i] /= stat;
    }
}

/* workspace for polyfit() */
 double *beta, **covar, **covar0; int
*indxc, *indxr, *ipiv;

/* This function allocates workspace for dfa() and polyfit(), and
sets x[i][j] = i**(j-1), in preparation for polyfit(). */
```

```
void
setup() {
    long i;
    int j, k;

    beta = vector(1, nfit);
    covar = matrix(1, nfit, 1, nfit);
    covar0 = matrix(1, nfit, 1, nfit);
    indxc = ivector(1, nfit);
    indxr = ivector(1, nfit);
    ipiv = ivector(1, nfit);
    mse = vector(1, nr);
    x = matrix(1, rs[nr], 1, nfit);
    for (i = 1; i <= rs[nr]; i++) {
    x[i][1] = 1.0;
    x[i][2] = i;
    for (j = 3; j <= nfit; j++)
        x[i][j] = x[i][j-1] * i;
    }
}

/* This function frees all memory previously allocated by this
program. */
void cleanup() {
    free_matrix(x, 1, rs[nr], 1, nfit);
    free_vector(mse, 1, nr);
    free_ivector(ipiv, 1, nfit);
    free_ivector(indxr, 1, nfit);
    free_ivector(indxc, 1, nfit);
    free_matrix(covar0, 1, nfit, 1, nfit);
    free_matrix(covar, 1, nfit, 1, nfit);
    free_vector(beta, 1, nfit);
    free_lvector(rs, 1, rslen);
/* allocated by rscale() */
    free(seq);
/* allocated by input() */ }

static char *help_strings[] = { "usage: \%s [OPTIONS ...]\n",\\
"where OPTIONS may include:",\\ " -d K  detrend using a polynomial
of degree K",\\ " default: K=1 -- linear detrending)", " -h print
this usage summary",\\ " -i input series is already integrated",\\
" -l MINBOX smallest box width (default: 2K+2)",\\ " -s sliding
window DFA",\\ " -u MAXBOX largest box width (default: NPTS/4)",\\
"The standard input should contain one column of data in text
format.",\\ "The standard output is two columns: log(n) and log(F)
[base 10 logarithms],",\\ "where n is the box size and F is the
root mean square fluctuation.", NULL };

void help(void) {
    int i;

    (void)fprintf(stderr, help_strings[0], pname);
    for (i = 1; help_strings[i] != NULL; i++)
    (void)fprintf(stderr, "%s\n", help_strings[i]);
}

/* polyfit() is based on lfit() and gaussj() from Numerical
Recipes in C(Press, Teukolsky, Vetterling, and Flannery; Cambridge
U. Press, 1992). It fits a polynomial of degree (nfit-1) to a set
of boxsize points given by x[1...boxsize][2] and y[1...boxsize].
The return value is the sum of the squared errors (chisq) between
the (x,y) pairs and the fitted polynomial.
*/ double
polyfit(double **x, double *y, long boxsize, int nfit) {
    int icol, irow, j, k;
    double big, chisq, pivinv, temp;
    long i;
    static long pboxsize = 0L;

/* This block sets up the covariance matrix.  Provided that
boxsize never decreases (which is true in this case), covar0 can
be calculated incrementally from the previous value. */
    if (pboxsize != boxsize) {
    /* this will be false most of the time */
    if (pboxsize > boxsize)
    /* this should never happen */
        pboxsize = 0L;
    if (pboxsize == 0L)
     /* this should be true the first time only */
        for (j = 1; j <= nfit; j++)
        for (k = 1; k <= nfit; k++)
            covar0[j][k] = 0.0;
    for (i = pboxsize+1; i <= boxsize; i++)
        for (j = 1; j <= nfit; j++)
        for (k = 1, temp = x[i][j]; k <= j; k++)
            covar0[j][k] += temp * x[i][k];
    for (j = 2; j <= nfit; j++)
        for (k = 1; k < j; k++)
        covar0[k][j] = covar0[j][k];
    pboxsize = boxsize;
    }
    for (j = 1; j <= nfit; j++) {
    beta[j] = ipiv[j] = 0;
    for (k = 1; k <= nfit; k++)
        covar[j][k] = covar0[j][k];
    }
    for (i = 1; i <= boxsize; i++) {
    beta[1] += (temp = y[i]);
    beta[2] += temp * i;
    }
    if (nfit > 2)
    for (i = 1; i <= boxsize; i++)
        for (j = 3, temp = y[i]; j <= nfit; j++)
        beta[j] += temp * x[i][j];
    for (i = 1; i <= nfit; i++) {
    big = 0.0;
    for (j = 1; j <= nfit; j++)
        if (ipiv[j] != 1)
        for (k = 1; k <= nfit; k++) {
            if (ipiv[k] == 0) {
            if ((temp = covar[j][k]) >= big ||
                (temp = -temp) >= big) {
                big = temp;
                irow = j;
                icol = k;
            }
            }
            else if (ipiv[k] > 1)
            error("singular matrix");
        }
    ++(ipiv[icol]);
if (irow != icol) { for (j = 1; j <= nfit; j++)
SWAP(covar[irow][j], covar[icol][j]); SWAP(beta[irow],
beta[icol]);
    }
    indxr[i] = irow;
    indxc[i] = icol;
    if (covar[icol][icol] == 0.0) error("singular matrix");
    pivinv = 1.0 / covar[icol][icol];
    covar[icol][icol] = 1.0;
    for (j = 1; j <= nfit; j++) covar[icol][j] *= pivinv;
    beta[icol] *= pivinv;
    for (j = 1; j <= nfit; j++)
        if (j != icol) {
        temp = covar[j][icol];
        covar[j][icol] = 0.0;
        for (k = 1; k <= nfit; k++)
        covar[j][k] -= covar[icol][k]*temp;
        beta[j] -= beta[icol] * temp;
        }
    }
    chisq = 0.0;
    if (nfit <= 2)
    for (i = 1; i <= boxsize; i++) {
        temp = beta[1] + beta[2] * i - y[i];
        chisq += temp * temp;
    }
    else
    for (i = 1; i <= boxsize; i++) {
        temp = beta[1] + beta[2] * i - y[i];
        for (j = 3; j <= nfit; j++) temp += beta[j] * x[i][j];
        chisq += temp * temp;
    }
    return (chisq);
}

/* The functions below are based on those of the same names in
Numerical Recipes (see above). */
void error(char error_text[]) {
 fprintf(stderr, "%s: %s\n", pname, error_text);
```

```
    exit(1);
}

double *vector(long nl, long nh) /* allocate a double vector with
subscript range v[nl..nh] */
 {
double *v = (double *)malloc((size_t)((nh-nl+2) *
sizeof(double)));
if (v == NULL) error("allocation failure in
vector()");
    return (v-nl+1);
}

int *ivector(long nl, long nh) /* allocate an int vector with
subscript range v[nl..nh] */ { int *v =
(int*)malloc((size_t)((nh-nl+2) * sizeof(int)));
    if (v == NULL) error("allocation failure in ivector()");
    return (v-nl+1);
}

long *lvector(long nl, long nh) /* allocate a long int vector with
subscript range v[nl..nh] */
 {
    long *v = (long *)malloc((size_t)((nh-nl+2) * sizeof(long)));
    if (v == NULL) error("allocation failure in lvector()");
    return (v-nl+1);
}

double **matrix(long nrl, long nrh, long ncl, long nch) /*
allocate a double matrix with subscript range
m[nrl..nrh][ncl..nch]
*/ {
    long i, nrow = nrh-nrl+1, ncol = nch-ncl+1;
    double **m;

/* allocate pointers to rows */
 m = (double **)
malloc((size_t)((nrow+1) * sizeof(double*))); if (!m)
error("allocation failure 1 in matrix()");
    m += 1;
    m -= nrl;

    /* allocate rows and set pointers to them */
m[nrl] = (double *) malloc((size_t)((nrow*ncol+1) *
sizeof(double))); if (!m[nrl]) error("allocation failure 2 in
matrix()");
    m[nrl] += 1;
    m[nrl] -= ncl;

    for (i = nrl+1; i <= nrh; i++) m[i] = m[i-1]+ncol;

/* return pointer to array of pointers to rows */
    return (m);
}

void free_vector(double *v, long nl, long nh)\\
 /* free a double vector allocated with vector() */
 \\ {
    free(v+nl-1);
}

void free_ivector(int *v, long nl, long nh) \\
/* free an int vector allocated with ivector() */ {
    free(v+nl-1);
```

```
}

void free_lvector(long *v, long nl, long nh) \\
/* free a long int vector allocated with lvector() */ \\
{
    free(v+nl-1);
}

void free_matrix(double **m, long nrl, long nrh, long ncl, long
nch) \\
 /* free a double matrix allocated by matrix() */
 \\ {
    free(m[nrl]+ncl-1);
    free(m+nrl-1);
}
```

REFERENCES

1. Mazur, K., "More data about dielectric and electret properties of poly(methyl methacrylate)", *Journal of Physics D*, Vol. 30, pp. 1383-1398, 1997.

2. Adamec, V. and J. H. Calderwood, "Electrical conduction and polarisation phenomena in polymers at low fields", *Journal of Physics D*, Vol. 11, pp. 781-800, 1978.

3. Boettcher S. and M. Paczuski, "Aging in a Model of Self-Organized Criticality", *Physical Review Letters*, Vol. 79, pp. 889-892, 1997.

4. Bellon, L., S. Ciliberto and C. Laroche, "Memory in the aging of a polymer glass", *Europhysics Letters*, Vol. 51, pp. 551-556, 2000.

5. Hacinliyan, A. Y. Skarlatos, G. Sahin and G. Akin, "Signals of chaotic behavior in PMMA", *Chaos, Solitons and Fractals*, Vol. 17, pp. 575-585, 2003.

6. Hacinliyan, A., Y. Skarlatos, H. A. Yildirim and G. Sahin, "Characterization of Chaocity in The Transient Current Through PMMA Thin Films", *Fractals*, accepted for publication.

7. Yildirim, H. A., "Chaotic Structures in Conductivity Mechanisms", Boğaziçi University, Istanbul, 2006.

8. Pietronero, L. and S. Strassler, "Nonlinear conductivity and noise spectrum of a pinned charge-density wave", *Phys. Rev. B*, Vol. 28, pp. 5863-5865, 1983.

9. Erzan, A., E. Veermans, R. Heijung and L. Pietronero, "Glassy dynamics of pinned charge density waves", *Phys. Rev. B*, Vol. 41, pp. 11522-11528, 1990.

10. Allegrini, P., G. Aquino, P. Grigolini, L. Palatella, A. Rosa, B. J. West , "Correlation function and generalized master equation of arbitrary age", *Phys.*

Rev. E, Vol. 71, pp. 066109 1-12, 2005.

11. Barkai, E., "Aging in Subdiffusion Generated by a Deterministic Dynamical System", *Phys. Rev. Lett.*, Vol. 90, pp. 104101-104110, 2003.

12. Allegrini, P., G. Aquino, P. Grigolini, L. Palatella, and A. Rosa, *Phys. Rev. E*, "Generalized master equation via aging continuous-time random walks", Vol. 68, pp. 56123-56133, 2003.

13. Efimenko, K., V. Rybka, V. Svorcik and W. Hnatowicz, "Mechanism of conductivity in metalpolymermetal structures", *Journal of Applied Physics*, Vol. 68, pp. 479-482, 1999.

14. Hegger, R., H. Kantz and T. Schreiber, "Practical implementation of nonlinear time series methods: The TISEAN package", *CHAOS*, Vol. 94, pp. 413-435, 1999.

15. Abarbanel, H. D. I., "Analysis of Observed Chaotic Data", Springer Verlag, New York, 1996.

16. Abarbanel, H. D. I., R. Brown, J. J. Sidorowich and L. S. Tsimring, "The analysis of observed chaotic data in physical systems", *Rev. Mod. Phys.*, Vol. 65, pp. 1331-1392, 1993.

17. Fraser, A. M. and H. L. Swinney, "Independent coordinates for strange attractors from mutual information", *Phys. Rev. A*, Vol. 33, pp. 1134-1140, 1986.

18. Kantz, H. and T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge University Press, 1997.

19. Kennel, M. B., R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction", *Phys. Rev. A*, Vol. 45, pp.3403-3411, 1992.

20. Wolf, A., J. B. Swift, H. L. Swinney and J. A. Vastano, "Determining Lyapunov exponents from a time series", *Physica D*, Vol. 16, pp. 285-317, 1985.

21. Abarbanel, H. D. I., R. Brown, and M. B. Kennel, "Variation of Lyapunov exponents on a strange attractor", *Journal of Nonlinear Sci.*, Vol. 2, pp. 175-199, 1991.

22. Sano, M. and Y. Sawada, "Measurement of the Lyapunov spectrum from a chaotic time series", *Phys. Rev. Lett.*, Vol. 55, pp. 1082-1085, 1985.

23. Eckmann, J. P., S. O. Kamphorst, D. Ruelle, and S. Ciliberto, "Liapunov exponents from time series", *Phys. Rev. A*, Vol. 34, pp. 4971-4979, 1986.

24. Kantz, H., "A robust method to estimate the maximal Lyapunov exponent of a time series", *Phys. Rev. A*, Vol. 185, pp. 77-87, 1994.

25. Das-Gupta, D. K., "Proc. 4th Int. Conf. on Conduction and Breakdown in Solid Dielectrics", *Sestri Levante (Piscataway, NJ: IEEE)*, 1992.

26. Scher, H. and E. W. Montroll, "Anamolous transit-time disperion in solids ", *Physical Review B*, Vol. 12, pp. 2455-2477, 1975.

27. Nigmatullin, R. R., "The realization of the generalized transfer equation in a medium with fractal geometry", *Phys. Stat. Sol. (B)*, Vol. 133, pp. 425-430, 1986.

28. Nigmatullin, R. R., L. A. Dissado and N. N. Soutougin, "Fractal pore model for Archie's law in sedimentary rocks", *J. Phys. D*, Vol. 25, pp.32-37, 1992.

29. Heuer, A. and P. Neu, "Tunneling dynamics of side chains and defects in proteins, polymer glasses, and OH-doped network glasses", *J. Chem. Phys.*, Vol. 107, pp. 8686-8720, 1997.

30. Wubbenhorst, M., C. A. Murray and J. R. Dutcher, "Dielectric relaxations in

ultrathin isotactic PMMA films and PS-PMMA-PS trilayer films", *Eur. Phys. J. E*, Vol. 12, pp. 109 - 112, 2003.

31. Valdmanis, J. A., G. Mourou and G. W. Gabel , "Picosecond electro-optic sampling system", *Appl. Phys. Lett.*, Vol. 41, pp. 211-212, 1982

32. Le Gressus, C., F. Valai, M. Gautier, J. P. Durnand and H. Okuzunu , "Dielectric polarisation relaxation measurement by means of a scanning electron microscope technique",*Journal of Applied Physics*, Vol. 74, pp. 1250-1255, 1993.

33. Gong, H., K. M. Chooi and C. K. Ong, "Charging of deformed semicrystalline polymers observed with a scanning electron microscope", *EEE Transactions on Dielectrics and Electrical Insulation*, Vol. 2, pp. 1123-1131, 1995.

34. Rubner, M. F. and G. J Ashwell, "Conjugated polymeric conductors in Molecular Electronics", John Wiley and Sons Inc., New York, 1992.

35. Ellenbogen, J. C. and J. C. Love, "Architectures for molecular electronic computers: 1. Logic structures and an adder built from molecular electronic diodes", The MITRE Corporation McLean, Virginia, 1999.

36. Su, W. P., J. R. Schrieffer, and A. J. Heeger, "Solitons in Polyacetylene", *Phys. Rev. Lett.*, Vol. 42, pp. 1698-1701, 1979.

37. Heeger, A. J., S. Kivelson, J. R. Schrieffer, and W. P. Su, "Solitons in conducting polymers", *Rev. Mod. Phys.*, Vol. 60, pp. 781-850, 1988.

38. Yao, K. L., S. E. Han, L. Zhao, "The polaron and bipolaron states of poly(phenylene vinylene)",*The Journal of Chemical Physics*, Vol. 114, pp.6437-6442, 2001.

39. Song, Z. G., H. Gong and C. K. Ong , "The trapping and distribution of charge in polarized polymethylmethacrylate under electron-beam irradiation", *J. Phys. D*, Vol. 30, pp. 1561-1565, 1997.

40. Gruner, G., "The dynamics of charge-density waves", *Rev. Mod. Phys.*, Vol. 60, pp. 1129-1181, 1988.

41. Karttunen, M., M. Haataja, K. R. Elder, and M. Grant, "Defects, order, and hysteresis in driven charge-density wave", *Phys. Rev. Lett.*, Vol. 83, pp. 3518-3521, 1999.

42. Thorne, R. E, "Charge-Density-Wave Conductors", *Physics Today*, pp. 42-47, 1996.

43. Brown, S. and G. Gruner, "Charge and Spin Density Waves", *Scientific American*, pp. 50-58, 1994.

44. Fukuyama, H. and P. A. Lee, "Dynamics of the charge-density wave: Impurity pinning in a single chain", *Phys. Rev. B*, Vol. 17, pp. 535541, 1978.

45. Lee, P. A. and T. M. Rice, "Electric field depinning of charge density waves", *Phys. Rev. B*, Vol. 19, pp. 39703980, 1979.

46. Gruner, G., A. Zawadowski, and P. M. Chaikin, "Nonlinear Conductivity and Noise due to Charge-Density-Wave Depinning in $NbSe_3$", *Phys. Rev. Lett.*, Vol. 46, pp. 511-515, 1981.

47. Bardeen, J., "Theory of Non-Ohmic Conduction from Charge-Density Waves in $NbSe_3$", *Phys. Rev. Lett.*, Vol. 42, pp. 1498-1500, 1979.

48. Gruner, G., A. Zettl, W. G. Clark, and J.Bardeen, "Field and frequency dependence of charge-density-wave conduction in $NbSe_3$", *Phys. Rev. B.*, Vol. 24, pp. 7247-7257, 1981.

49. Monceau, P., J. Richard and M. Renard ,"Charge-density-wave motion in $NbSe_3$.I. Studies of the differential resistance dV/dI", *Phys. Rev. B*, vol. 25, pp. 931-947, 1982.

50. Feder, J.,*Fractals*, Plenum Press, New York, 1988.

51. Peng, C. K., J. M. Hausdorff and A. L. Goldberger, ”Fractal mechanisms in neural control: Human heartbeat and gait dynamics in health and disease. In: Walleczek J”, ed. Self-Organized Biological Dynamics and Nonlinear Control. Cambridge: Cambridge University Press, 2000.

52. Kolmogorov, K. N., ”Local structure of turbulence in fluid for very large Reynolds numbers” , *Transl. in Turbulence*, Vol. 434, pp. 151-155, 1961.

53. Hurst, H. E., ”Long-term storage capacity of reservoirs”, *Trans. Am. Soc. Civil Engineers*, Vol. 116, pp. 770-799, 1951.

54. Beran, J., ”Statistics for Long-Memory Processes”, Chapman and Hall, New York 1994.

55. Peng, C. K., S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley and A. L. Goldberger, ”Mosaic organization of DNA nucleotides”, *Phys. Rev. E.*, Vol. 49, pp. 1685-1689, 1994.

56. Peng, C. K., S. Havlin, H. E. Stanley, A. L. Goldberger, ”Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series”, *Chaos*, Vol. 5, pp. 82-87, 1995.

57. Chen, Z., P. C. Ivanov , K. Hu, H. E. Stanley, ”Effect of Trends on Detrended Fluctuation Analysis”, *Phys. Rev. E.*, Vol. 65, pp. 011114 1-19, 2002.

58. Buldyrev, S. V., A. L. Goldberger , S. Havlin, C. K. Peng , H. E. Stanley and M. Simons, ”Fractal landscapes and molecular evolution: modeling the myosin heavy chain gene family”, *Biophys. J.*, Vol.65, pp.2673-2679, 1993.

59. Ossadnik, S. M., S. V. Buldyrev, A. L. Goldberger, S. Havlin, R. N. Mantegna, C. K. Peng, M .Simons, H.E Stanley, ”Correlation approach to identify coding regions in DNA sequences”, *Biophys. J.*, Vol. 67, pp. 64-70, 1994.

60. Hausdorff, J. M., C. K. Peng, Z. Ladin, J. Y. Wei, A. L. Goldberger, "Is walking a random walk? Evidence for long-range correlations in the stride interval of human gait", *J. Appl. Physiol.*, Vol. 78, pp. 349-358, 1995.

61. Hausdorff, J. M., P. Purdon, C. K. Peng, Z. Ladin, J. Y. Wei, A. L. Goldberger, "Fractal dynamics of human gait: stability of long-range correlations in stride interval fluctuations", *J. Appl. Physiol.*, Vol. 80, pp. 1448-1457, 1996.

62. Hu, K., P.Ch. Ivanov, Z. Chen, P. Carpena, H. E. Stanley, "Effect of nonstationarities on detrended fluctuation analysis", *Phys. Rev. E.*, Vol. 64, pp. 011114-15, 2001.

63. Goldberger, A. L., L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J.E. Mietus, G. B. Moody, C. K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23): e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]; 2000 (June 13).

64. Montroll, E. W. and M. F. Shlesinger, "The wonderful world of random walks. In: Nonequilibrium Phenomena II. From Stochastics to Hydrodynamics", Amsterdam: North-Holland,pp. 1-121, 1984.

65. Iyengar, N., C. K. Peng, R. Morin, A. L. Goldberger, and L. A. Lipsitz, "Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics", *Am. J. Physiol.*, Vol. 217, pp. R1078-R1084, 1996.

66. Ivanov, P. C., A. Bunde, L. A. Nunes Amaral, S. Havlin, J. Fritsch-Yelle, R. M. Baevsky, H. E. Stanley, and A.L. Goldberger, "Sleep-wake differences in scaling behavior of the human heartbeat: Analysis of terrestrial and long-term space flight data", *Europhys. Lett.*, Vol. 48, pp. 594-600, 1999.

67. Ivanov, P. C., L. A. Nunes Amaral, A. L. Goldberger, and H. E. Stanley,

"Stochastic feedback and the regulation of biological rhythms", *Europhys. Lett.*, Vol. 43, pp. 363-368, 1998.

68. Kantelhardt, J. W., E. Koscielny-Bunde, H. H. A. Rego, S. Havlin, and A. Bunde, "Detecting long-range correlations with detrended fluctuation analysis", *Physica A*, Vol. 294,pp. 441-452, 2001.

69. Makarenko, N., L. M. Karimova, B. I. Demchenko and M. M. Novak, "Analysis of Terrestrial Radioactive Contamination" , *Fractals*, Vol. 6, pp. 359-364, 1998.

70. Viswanathan, G. M., S. V. Buldyrev, E. K. Garger, V. A. Kashpur, L. S. Lucena, A. Shlyakhter, H. E. Stanley, and J. Tschiersch, "Quantifying nonstationary radioactivity concentration fluctuations near Chernobyl: A complete statistical description", *Phys. Rev. E*, Vol. 62, 43894392, 2000.

71. Makse, H. A., S. Havlin, M. Schwartz, and H. E. Stanley, "Method for generating long-range correlations for large systems", Phys. Rev. E., Vol. 53, pp. 5445-5449, 2001.

72. James, F. and M. Roos, "CERN library program D506", *Comp. Phys. Comm.*, Vol. 10, pp. 343-365, 1975.

73. Marsaglia, G. and W. W. Tsang, "The ziggurat method for generating random variables", *J. Statist. Software*, Vol.5, pp.1-7, 2000.

74. Lewis, T. J., "Charge transport, charge injection and breakdown in polymeric insulators", *J. Phys. D*, Vol. 23, pp. 1469-1478, 1990.

75. Sorensan, J., *Engineering Tools for Internal Charging, European Space Agency* ,DERA/CIS/CIS2/CR000277 issue 1.0, 2000.

76. Peng, C. K., S. Havlin, M. Schwartz, H. E. Stanley, "Directed-polymer and ballistic-deposition growth with correlated noise", *Phys. Rev. A*, Vol. 44, pp.

2239-2242, 1991.

77. Prakash, S., S. Havlin, M. Schwartz, H. E. Stanley, "Structural and dynamical properties of long-range correlated percolation", *Phys. Rev. A*, Vol. 46, pp.1724-1727, 1992.