

Copyright © 1990, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**BIVARIATE TIME SERIES ANALYSIS  
OF SIMULATED ANNEALING DATA**

by

Gregory Sorkin

Memorandum No. UCB/ERL M90/6

10 January 1990

CONFIDENTIAL

**BIVARIATE TIME SERIES ANALYSIS  
OF SIMULATED ANNEALING DATA**

by

Gregory Sorkin

Memorandum No. UCB/ERL M90/6

10 January 1990

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering  
University of California, Berkeley  
94720

7766 PAGE

**BIVARIATE TIME SERIES ANALYSIS  
OF SIMULATED ANNEALING DATA**

by

**Gregory Sorkin**

**Memorandum No. UCB/ERL M90/6**

**10 January 1990**

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering  
University of California, Berkeley  
94720

It can easily be shown that this process is equivalent to a random walk on a graph containing self-loops, with random selection of an edge out of a node in proportion to the weight of that edge, and with the edge weights determined by the temperature  $T$ .

A typical application is a placement problem, say that of placing 100 circuits on a  $10 \times 10$  grid. Here a state is a permutation of the numbers from 1 to 100, representing a placement. A move could consist of interchanging any two circuits. Thus each state would have  $C_2^{100} = 4950$  neighbors. We also define the distance between two states as the length of a shortest path connecting them.

While annealing works well on a wide variety of practical problems, it cannot work well on arbitrary graphs, nor on any graph with an arbitrary energy function. The goal of our research has been to characterize the energy landscape itself (the graph and energy function) and the behavior of the annealing algorithm. Because the energy landscape itself is terrifically complex, we have focused on the energy trajectory, *i.e.* the sequence of energies following each move accepted.

## 2 Background

We chose a simplified version of circuit placement for experimental analysis. The problems considered consist of unit-square circuits, with  $n$  circuits to be placed on a grid of size  $1 \times n$  ('one-dimensional placement') or size  $\sqrt{n} \times \sqrt{n}$  ('two-dimensional placement'). A netlist defines which circuits are to be connected, and the cost of a placement is the total Manhattan length of wire required to connect the circuits.

In previous, univariate time series analysis of the energy data against time measured by number of moves accepted, we considered a variety of netlists, both 1- and 2- dimensional energy functions, and a variety of temperatures. Due to the ever-increasing complexity of the analysis, in this work we restrict ourselves to a single choice: The netlist came from a real sample of VLSI logic, the energy chosen was that for 1-dimensional placement, and the temperature used was 10. The length of the series was 11,000.

The choices of netlist and cost function were relatively arbitrary, since from the univariate analysis we inferred no qualitative difference between examples. The choice of temperature is somewhat more important. Typical energy changes are in the thousands and the minimum nonzero energy change possible is 1. At  $T = \infty$  every move is accepted, so the number of attempts is identically 1. At  $T \approx 1$  the energy is almost constant. Thus 10 was chosen as an intermediate value of temperature where the process is most interesting.

# Bivariate Time Series Analysis of Simulated Annealing Data

Gregory Sorkin

January 1990

## 1 Introduction

Simulated annealing (SA) is a recent technique for finding good solutions to a wide variety of combinatorial optimization problems. Given is a graph with an energy  $E$  assigned to each node. In simulated annealing parlance, the nodes are called ‘states’, the arcs represent ‘moves’ from one state to a neighboring state, and the energy is sometimes called ‘cost’. The simulated algorithm then proceeds as follows:

```
state = random initial state
repeat (until done) {
    T = new temperature
    repeat (until inner-loop criterion) {
        newstate = random neighbor(state)
        ΔE = E(newstate) - E(state)
        If  $\Delta E \leq 0$ , state = newstate.
        Else state = newstate with probability  $e^{-\Delta E/T}$ 
    }
}
```

The temperature  $T$  is a parameter of the algorithm. It is always decreased gradually as the annealing proceeds, but the optimal control of  $T$  is not understood.

For a substantial discussion of the analysis of the energy after each acceptance as a univariate time series, see [3]. The goal of that work was largely to see if the energy time series was consistent with the energy landscape having a random fractal structure, as we have conjectured that such structure is typical of real-world problems and is the reason why simulated annealing is successful for these problems [2, 1]. The univariate analysis revealed that the energies were well modeled by a first-order autoregressive with coefficient slightly less than 1. This was interpreted to be consistent with a landscape where the mean square energy difference between two points divided by the distance between them has a distribution independent of the distance, a sort of high-dimensional fractional Brownian motion.

We now go on to analyze the energy after each accepted move and the number of attempts it took for that move to be accepted as a bivariate time series. Again we have in mind to test whether the landscape is fractal. In particular we sought to confirm two conjectures:

Conjecture 1: If we are in a state of high energy, it is more likely that a neighboring state has lower energy. If this is so, it would mean that the probability of accepting a move starting from a high-energy state would be higher than that of accepting a move starting from a low-energy state, because the probability of accepting a move is negatively correlated with the energy increase by that move. Thus if the conjecture is true, the number of moves which must be attempted before one is accepted is smaller starting from a high-energy state.

Conjecture 2: The space has nonuniform structure, such as that described by a self-similar fractal. Then both the energies and the number of moves attempted will vary depending on which part of the space we are in, and by default we expect a derivative relationship between these two series. In particular, we hope that a run of low energies indicates that we are in a low-lying part of the landscape, which in turn will bear upon the typical number of attempts needed.

Because of the above conjectures, and just to search for possible relationships between two of annealing's few observable quantities, we analyzed the bivariate series consisting of the energy after each accepted move and the number of attempts to the next acceptance.

### 3 Univariate Analysis

Because it can be important to prewhiten the ‘ $X$ ’ (energy) series, and because of the failure to compute confidence intervals in [3], we begin with a quick univariate analysis.

First, it was shown in [3] that the energy spectrum is closely matched by that of an AR(1) process. Also it was seen that when a higher-order AR( $p$ ) process was fitted the first coefficient was close to 1 but the others were near 0. However, confidence intervals or formal tests were not used to determine whether the other coefficients were in fact 0.

Since Splus apparently lacks a program for properly computing standard errors, the NAG program ‘transfer2’ (in a revised version called ‘transfer3’) was used instead. Transfer3 normally fits an ARMA model to bivariate data, but takes as an input the lag of the  $Y$  series behind  $X$ . Setting  $Y(t) = X(t)$  and using lag 1 means that a model of the form  $X(t) = a(1)X(t-1) + \dots + a(p)X(t-1-p)$  can be fitted. That is, transfer3 is fitting a MA transfer function, to give what is really a univariate AR model. Doing this for the r12 data with a MA(5) model returns the coefficient values and standard errors below: A number of the coefficients

parameter	estimate	standard error
ma(1)	0.94115	0.03169
ma(2)	-0.06652	0.04349
ma(3)	0.04888	0.04352
ma(4)	-0.06461	0.04360
ma(5)	0.03960	0.03180
mean	0.05278	0.36730

Table 1: AR(5) energy model

are small, but about 50% larger than the corresponding standard errors, so it is still unclear whether the process is Markov, in a restricted linear sense. Using more data might help to resolve this issue, as would applying the same analysis to some of the other series. Unfortunately transfer3 is currently restricted to use some number of points well under 2,000, so while the series itself had length 11,000 (used in the remainder of the analysis), the ARMA analysis was limited to 1000 points, resulting in comparatively large standard errors. Also, when the data for this example has not been demeaned, transfer3 inexplicably gives wrong moving average coefficients and fails to return an estimate of the mean. Thus the energy data were demeaned before applying transfer3, rendering meaningless the ‘mean’ estimate above. Note also that based on a test example, all but the first of transfer3’s MA parameter estimates have reversed sign, so that for all the data

presented here the proper interpretation is  $Y(t) = a(1)X(t-1) - a(2)X(t-2) - \cdots - a(k)X(t-k)$ .

For our present purposes the exact form of the energy series is immaterial. What is important is that it be approximately white, so that transients will not enter into the cross-spectrum and so that the standard error and confidence interval estimates will be relatively accurate. Applying a simple filter  $(1, -ar(1))$  to the energies gives the ‘residual energy’ series shown in frame 1 of figure 1. Its spectrum is fairly white, as per frame 3 of figure 1.

The second series, number of attempts, is shown in figure 1 frame 3. Its spectrum (frame 4) is fairly white to begin with.

The spectra were both computed using a cosine bell taper applied to 10% of the data at each end, and smoothing with a moving average filter of length 20. The 95% confidence interval shown was computed using the same taper but a filter of length 100.

The series and spectra have been plotted as points rather than lines because their relative whiteness means that a line plot looks like a solid blob due to the finite resolution of the plot. Note too that frame 2 of figure 1 does depict a function, *i.e.* each  $x$  value maps to just one  $y$  value, and so the sets of  $x$  values corresponding to each  $y$  are disjoint. The appearance to the contrary is due to plotter resolution.

## 4 Bivariate Analysis

### 4.1 Frequency-Side

Following the previous discussion, the  $X$  series used is the residual of the energies after an  $ar(1)$  filtering, and the  $Y$  series is simply the number of attempts required for the next acceptance. These series and their spectra, shown in figure 1, have just been discussed.

The importance of figure 2 is contained entirely in its frame 3. This is a plot of the coherency between  $X$  and  $Y$ , which is small at all frequencies. The approximate  $100\beta\%$  null level for coherence is given by  $1 - (1 - \beta)^{1/(2m+1)}$  for periodogram estimates smoothed with an  $m$ -long filter. (Actually, to account for tapering, rather than  $2m+1$  we used  $df/2-1$  where  $df$  is Splus’s degrees of freedom parameter.) Examination of frame 3 reveals that the coherence estimate is nearly always beneath the 95% null line; in fact it exceeds that level just about 5% of the time, consistent with the coherence equal to 0.

Since the coherence does appear to be 0, there is really no point in going on: the  $X$  and  $Y$  series are not

related by linear filtering, as far as can be determined from this sample function. However, as a pointless demonstration of technique we did compute the transfer function, whose real and imaginary parts are shown respectively in frames 1 and 2 of figure 2.

The impulse response is then calculated as the real part of the inverse Fourier transform of the transfer function extended to the interval  $[-\pi, \pi]$ , followed by application of the filter (.23, .54, .23). Scaling, and centering at lag 0, were done experimentally based on a test series with  $Y(t) = X(t)$ . The resulting impulse response is plotted in figure 2, frame 4.

While this computed impulse response function is presumably nonsense (as per the coherence), it would be nice to explain the periodic envelope which seems to contain it.

## 4.2 Time-Side Analysis

Again, from the coherence computation we can infer in advance that the results here will be useless. But, especially since the time-side analysis can be done independently of the frequency-side one (if one chooses not to use DFT techniques), it is reasonable to see how the time-side analysis would go.

The simplest computation is that of the autocorrelation functions for each series and the cross-correlation function, which are shown respectively in frames 1–3 of figure 3. These plots also include lines describing one standard error above and below 0 correlation. The estimated correlations are always within or close to this standard error interval, except of course for the autocorrelations at lag 0 which are identically 1. We conclude that the correlations are all 0, or (more precisely but less truthfully) that this sample function gives no reason to believe otherwise. There is an anomaly here that it would nice to understand: We would expect the correlation estimate to lie outside the one standard error interval roughly 32% of the time; in this case it is more like 0%.

Finally, we can fit an ARMA model to the data. We know that annealing is a Markov process where the states of the process are precisely the states of the problem (the vertices of the annealing graph). Because we are interested in the extent to which the sequence of energies encapsulates relevant information about the current state, we chose to predict the number of attempts required from the energy data alone (not from past values of number of attempts). Similarly we chose to use a simple white noise model, in part because the Markov nature of the underlying process supports this choice and partly because the transfer3 failed when given any other noise model. That is, we chose to use a pure moving average transfer function, whose

order was arbitrarily chosen to be 4. The minimum lag was set to 0, i.e. the number of attempts for the next move is to be estimated by energies of the current state and 3 previous states.

The function transfer3 (a working version of the NAg program transfer2) was used for the estimates. Because the results were observed to be independent of the initial estimates, the latter are not presented. As previously mentioned, transfer3 cannot handle very long series, so only the first 1000 points were used.

The output parameter and standard error estimates are given in the table below. Except for the mean,

parameter	estimate	standard error
ma(0)	0.00859	0.01692
ma(1)	0.00452	0.01695
ma(2)	0.01663	0.01695
ma(3)	-0.00075	0.01693
mean	6.67498	0.18792

Table 2: MA(4) transfer function

the parameter estimates are not significantly different from 0, as measured by the standard error.

While we indicated that an autoregressive transfer function model is not what we are really looking for, it does make sense to see if such a model might fit the data well, so we also fit an ARMA(2,3) model. The results are shown below, and again indicate parameter values which are not significantly far from 0. It seems

parameter	estimate	standard error
ma(0)	0.00852	0.01694
ma(1)	0.00455	0.01519
ma(2)	0.01733	0.02024
ar(1)	0.37883	1.14961
ar(2)	0.53175	1.10237
mean	6.66760	0.18808

Table 3: ARMA(2,3) transfer function

likely that AR coefficients after the first one should be interpreted as negative terms, in symmetry with the MA coefficients, but this was not checked.

### 4.3 Nonlinear Relationships

All that the foregoing has shown is that there is no evidence for a linear relationship between the series of energies and number of attempts required. Intuitively we expect a relation between these quantities to be

monotonic (in some vague sense), but not necessarily linear. We expect a nonlinear relation to have some linear component which would be captured by the linear analysis described above, but it is possible that a small linear component has been missed.

To conduct a somewhat more thorough search for nonlinear relationships, we analyzed 2 different  $Y$  series: the reciprocal of the number of attempted moves, and the logarithm (base 10) of the number of attempted moves. The reciprocal is natural to look at because it can be thought of as an estimate of the probability of accepting a move. The log seemed a reasonable choice because it makes the distribution of  $Y$  values somewhat more uniform.

The  $X$  series was left untouched because we could not intuitively justify taking any function of it, and because the distribution of  $X$  values is close to normal, which seems a good thing.

Results for these series are virtually identical to those for the original series. Analyses for the bivariate series with  $Y$  equal to the inverse of the number of attempts are shown in figures 4–6, corresponding exactly to figures 1–3; and those for the logarithmic  $Y$  series are shown in figures 7–9. Tables for the ARMA analyses (MA only) are below.

parameter	estimate	standard error
ma(0)	-0.00008	0.00088
ma(1)	-0.00027	0.00088
ma(2)	-0.00147	0.00088
ma(3)	0.00023	0.00088
mean	0.33395	0.00976

Table 4:  $Y = 1/\#\text{attempts}$

parameter	estimate	standard error
ma(0)	0.00052	0.00114
ma(1)	0.00045	0.00114
ma(2)	0.00166	0.00114
ma(3)	-0.00033	0.00114
mean	0.65446	0.01268

Table 5:  $Y = \log(\#\text{attempts})$

## 5 Interpretation of Results

We began with the view that the state space of a typical annealing problem is complex, and in particular that there were low and high energy regions of the state space. Only part of that information would be captured by the energy of the state. For example we could be in a high energy state which nonetheless is in a region of generally low energy; in such a case we would expect that many neighbors would lie downhill, and so there would be an unusually high likelihood of generating a downhill step, which would automatically be accepted. Correspondingly, the number of attempts would be unusually small.

The results here present clear evidence for the nature of the annealing process, less clear relevance to the nature of the underlying space. The annealing process itself is well modeled as a simple Markov process. The energies follow an AR(1) model where the coefficient is close to 1, *i.e.* they are nearly a simple random walk on the line, but not quite so since the process must be stationary. (Annealing is by construction Markov on the configuration space, thus approaches stationarity. We start at time large, and energy is a function of state, so the energies are also stationary.) The distribution of waiting times leaving each state has been shown to be (linearly) independent of the energy and in fact independent of the sequence of energies. Even knowing that the next energy is smaller apparently does not bias the waiting time to be smaller.

This alone is enough to contradict Conjecture 1 (see ‘Background’). In fact, that conjecture was partially invalidated by the univariate analysis of energy, which showed that even if the current state has high energy, the next state is as likely as ever to have even higher energy. The bivariate analysis finishes off the conjecture, showing that the expected leaving time is also no lower from states of high energy.

Application of these observations to the underlying space – related to Conjecture 2 – is difficult. Conjecture 2 as a whole is ‘falsified’, as there is no apparent relation between the energy and attempts series. But it is not clear whether this is because the space has no structure, perhaps being some sort of high-dimensional analog of a random walk on the integers with no defined ‘regions’ of high or low energy; or if it is because the energy and attempted move series fail to capture the structure.

While the underlying space of a practical annealing problem is impossible to map out, it is possible to track distance data which (unlike energy and attempted move data) includes direct information about the topology of the space. We hope soon to analyze trivariate time series data including distance information to better elucidate the structure of the space.

## 6 Summary

Simulated annealing was run on a placement data from a real-world netlist at a fixed, intermediate temperature, and the energies and number of attempts per move were recorded. It was expected that the two time series would be related, and in particular that a high energy would correlate with a small number of moves attempted for the following step.

Univariate analysis shows that an AR(1) model provides a good fit to the energies, suggesting that any state information captured by the energy series is equally well captured by a single energy. It also reveals that the attempts series is approximately white, so that any simple prediction of the next number of attempts must come from the energy series, not from the numbers of attempts at other times.

Bivariate analysis fails to indicate any relationship between energy and attempts, in particular falsifying the expectation just mentioned. We conclude that either there is no general, slowly-varying characterization of state which would help describe the energy changes and numbers of attempts, or at least that these two series themselves fail to capture such a characterization.

## Acknowledgements

Following my interest in simulated annealing, I did this work as the final project for a U.C. Berkeley course in timeseries analysis, taught in Spring 1989 by Prof. David R. Brillinger. I am most grateful for his guidance and interest.

I would also like to acknowledge the financial support IBM Research, as well as that of MICRO, AT&T Bell Laboratories, Hughes Aircraft, Intel, LSI Logic, Philips Research Laboratory, and Rockwell International.

## References

- [1] Gregory B. Sorkin. Combinatorial optimization, simulated annealing, and fractals. RC 13674 (log #61253), I.B.M., April 1988.
- [2] Gregory B. Sorkin. Simulated annealing and fractals. Unpublished manuscript, March 1989.

[3] Gregory B. Sorkin. Univariate time series analysis of simulated annealing data. Technical Report UCB/ERL M90/5, Univ. of California at Berkeley, January 1990.

## 7 Appendix: Splus macros

```
MACRO doit30(var, attddtag)
({
# From structure including energy and attempted move data,
# perform frequency-side analysis.
#
?T_ar(var$data[(var$start-1):(var$stop)], aic=T, maxorder=1)
var_mstr(var, x= ?T$resid[-1], arl=?T)
?T_cbind(var$x, var$attd)

?T(spec)_ spectrum(?T, smooth=20, taper=.1, detrend=F, demean=T)
?fullspec(?T(spec), .95)
var_mstr(var, xspec20= ?T(spec))

?T(spec)_ spectrum(?T, smooth=100, taper=.1, detrend=F, demean=T)
?fixxspec(?T(spec), .95)
var_mstr(var, xspec100= ?T(spec))

?doit31(var, attddtag)

rm(?T, ?T(spec))
})
END
```

```

MACRO doit31(var, attddtag)
{
# Plots frequency-side results computed in doit30,
# including the data, univariate spectra, coherence, and transfer function.
#
par(mfrow=c(2,2))

plot(var$x, pch=". ", cex=.5)
title(main=encode(var$name, ": ar(1)-residual energies"),
      xlab="accepted move number", ylab="energy",
      sub=encode("ar(1)=", var$ar1$ar[1,1,1]), cex=.5)

plot(var$attd, pch=". ", cex=.5)
title(main=attddtag,
      xlab="accepted move number", ylab="", cex=.5)

plot(var$xspect20$freq, var$xspect20$spec[,1], pch=". ", cex=.5)
lines(var$xspect100$freq, var$xspect100$spec[,1]*var$xspect100$lower,
      lty=1, cex=.5)
lines(var$xspect100$freq, var$xspect100$spec[,1]*var$xspect100$upper,
      lty=1, cex=.5)
title(main=encode
      ("var$name, ": ar(1)-residual energy spectrum, ",
       round(100*var$xspect100$conf), "% CI", sep=""),
       xlab="frequency", ylab="power", cex=.5)

plot(var$xspect20$freq, var$xspect20$spec[,2], pch=". ", cex=.5)
lines(var$xspect100$freq, var$xspect100$spec[,2]*var$xspect100$lower,
      lty=1, cex=.5)
lines(var$xspect100$freq, var$xspect100$spec[,2]*var$xspect100$upper,
      lty=1, cex=.5)
title(main=encode
      ("attempts spectrum, ",
       round(100*var$xspect100$conf), "% CI", sep=""),
       xlab="frequency", ylab="power", cex=.5)

?printit()
par(mfrow=c(2,2))

plot(var$xspect20$freq, 0+var$xspect20$real, pch=". ", cex=.5)
title(main= encode(var$name, ": transfer function: real part", sep=""),
      xlab="frequency", ylab="", cex=.5)

plot(var$xspect20$freq, 0+var$xspect20$imag, pch=". ", cex=.5)
title(main="transfer function: imag part",
      xlab="frequency", ylab="", cex=.5)

plot(var$xspect20$freq, var$xspect20$coh, type="l", lty=1, cex=.5, ylim=c(0,1))
abline(h= var$xspect20$cohnnull, lty=2, cex=.5)
title(main=encode(
      "coherence, CI=", var$xspect100$conf, sep=""),
      sub=encode(
        .1*round(1000*mean(var$xspect20$coh>var$xspect20$cohnnull)),
        "% exceed null", sep=""),
      cex=.5)

plot(0+var$xspect20$imp, 0+var$xspect20$imp, type="l", lty=1, cex=.5)
title(main="impulse response", xlab="lag", ylab="", cex=.5)

?printit()
}
END

```

```

MACRO doit40(var)
{
# Compute and plot time-side analysis of bivariate data.
#
par(mfrow=c(2,2))

?T_cbind(var$x, var$attd)
?T(acf)_ acf(?T, maxlag=50, correlation=T)
var_mstr(var, acf= ?T(acf))

?T(lag)_ var$acf$lag[,1,1]
?T(se)_ 2/ sqrt(len(var$x)- abs(?T(lag)))
plot(0+ ?T(lag), var$acf$acf[,1,1], type="l", lty=1, cex=.5)
lines(?T(lag), ?T(se), lty=2, cex=.5)
lines(?T(lag), -?T(se), lty=2, cex=.5)
abline(h=0, lty=2, cex=.5)
title(main=encode(var$name,": Rxx", sep=""), xlab="lag", ylab="", cex=.5)

plot(0+ ?T(lag), var$acf$acf[,2,2], type="l", lty=1, cex=.5)
lines(?T(lag), ?T(se), lty=2, cex=.5)
lines(?T(lag), -?T(se), lty=2, cex=.5)
abline(h=0, lty=2, cex=.5)
title(main="Ryy", xlab="lag", ylab="", cex=.5)

?T(lag)_ c( rev(var$acf$lag[,2,1]), var$acf$lag[-1,1,2] )
?T(se)_ 2/ sqrt(len(var$x)- abs(?T(lag)))
plot( 0+ ?T(lag),
      c( rev(var$acf$acf[,2,1]), var$acf$acf[-1,1,2] ),
      type="l", lty=1, cex=.5)
lines( ?T(lag), ?T(se), lty=2, cex=.5)
lines( ?T(lag), -?T(se), lty=2, cex=.5)
abline(h=0, lty=2, cex=.5)
title(main="Ryx", xlab="lag [x(t), y(t-lag)]", ylab="", cex=.5)

?printit()
})
END

```

```
MACRO fixxspec(specstr, confpar)
{
# Operates on a spectrum. Changes power from decibel to linear scale.
# Computes confidence interval multipliers based on supplied
# confidence parameter.
specstr_mstr(specstr, spec= 10^(specstr$spec/10)/(2*pi))
?T(conf)_confpar
# 2m+2 = 2 (#pts averaged) = df/2; 4m+2 = df.
?T(lower)_specstr$df / qchisq((1+?T(conf))/2, specstr$df)
?T(upper)_specstr$df / qchisq((1-?T(conf))/2, specstr$df)
specstr_mstr(specstr,
  conf=?T(conf), lower=?T(lower), upper=?T(upper))

rm(?T(lower), ?T(upper), ?T(conf))
})
END
```

```
}
```

```

MACRO fullspec(specstr, confpar)
{
# Performs same function as fixxspec, but in addition computes
# transfer function and impulse response for bivariate series,
# as well as coherence null value.
  specstr_mstr(specstr, spec= 10^(specstr$spec/10)/(2*pi))

?T(conf)_ confpar
# 2m+2 = 2 (#pts averaged) = df/2; 4m+2 = df.
?T(lower)_ specstr$df / qchisq((1+?T(conf))/2, specstr$df)
?T(upper)_ specstr$df / qchisq((1-?T(conf))/2, specstr$df)
?T(gain)_ sqrt( specstr$coh * specstr$spec[,2] / specstr$spec[,1] )
?T(real)_ ?T(gain) * cos(specstr$phase)
?T(imag)_ ?T(gain) * sin(specstr$phase)
?T(realx)_ c( rev(?T(real)), ?drop(?T(real),1,0) )
?T(imagx)_ c( rev(?T(imag)), ?drop(?T(imag),1,0) )
?T(junk)_ fft( ?T(realx), ?T(imagx), inv=T )
?T(junk2)_ c( ?drop(?T(junk)$Real, len(?T(junk)$Real)/2, 0),
             ?drop(?T(junk)$Real, 0, len(?T(junk)$Real)/2) )
?T(imp)_ filter(?T(junk2), c(.23, .54, .23)) / len(?T(junk)$Real)
?T(impx)_ (-len(?T(imp))/2) : (len(?T(imp))/2 -1)
specstr_mstr(specstr,
  conf=?T(conf), lower=?T(lower), upper=?T(upper),
  cohnull= 1- (1-?T(conf))^ (1/ (specstr$df/2 -1)),
  gain= ?T(gain), real= ?T(real), imag= ?T(imag),
  imp= ?T(imp), impx= ?T(impx)
)

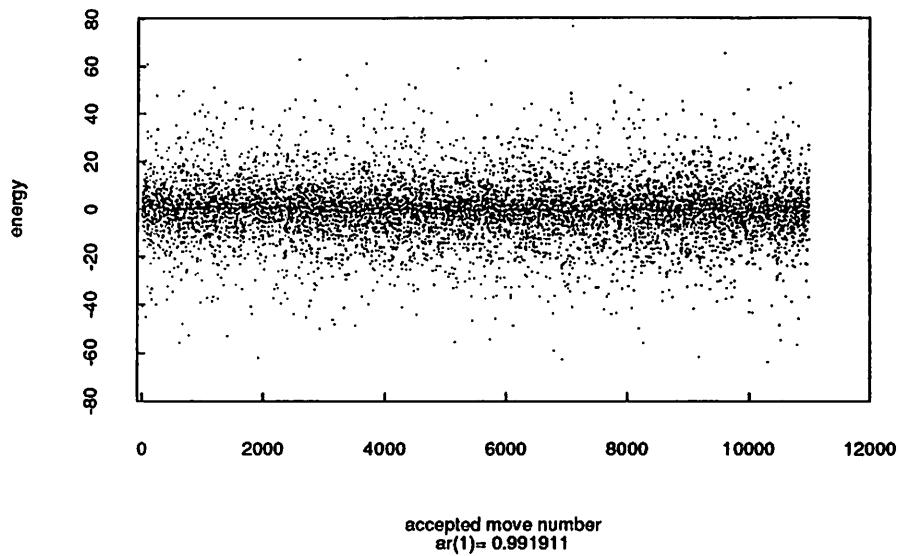
rm(?T(lower), ?T(upper), ?T(gain), ?T(real), ?T(imag), ?T(conf),
?T(realx), ?T(imagx), ?T(junk), ?T(junk2), ?T(imp), ?T(impx))
}
END

```

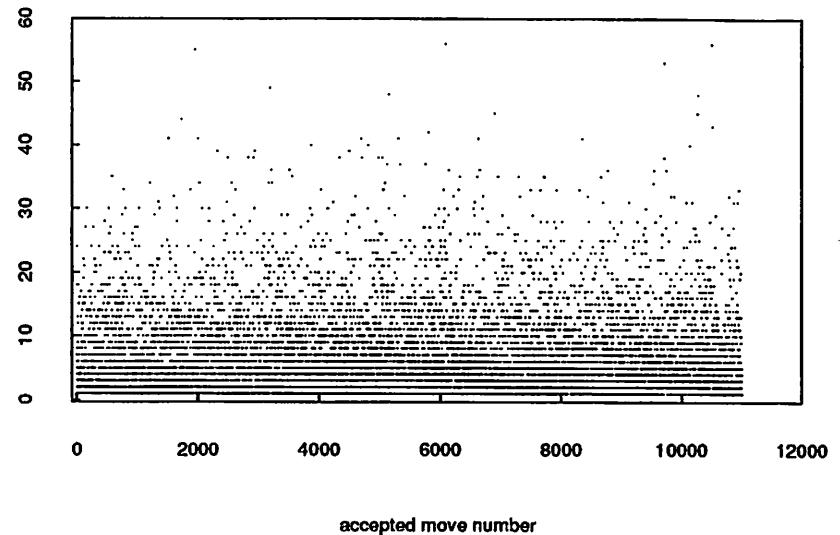
}

fig 1

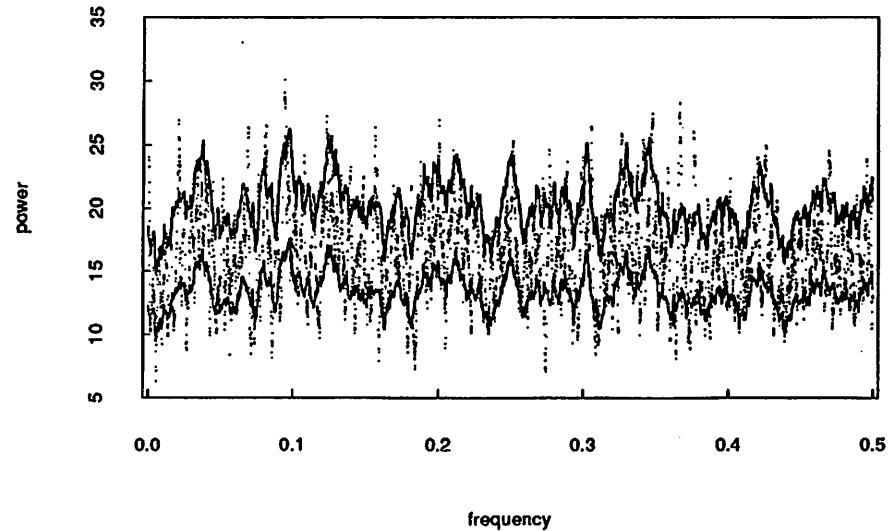
r12 (plain) : ar(1)-residual energies



#attempts



r12 (plain): ar(1)-residual energy spectrum, 95% CI



attempts spectrum, 95% CI

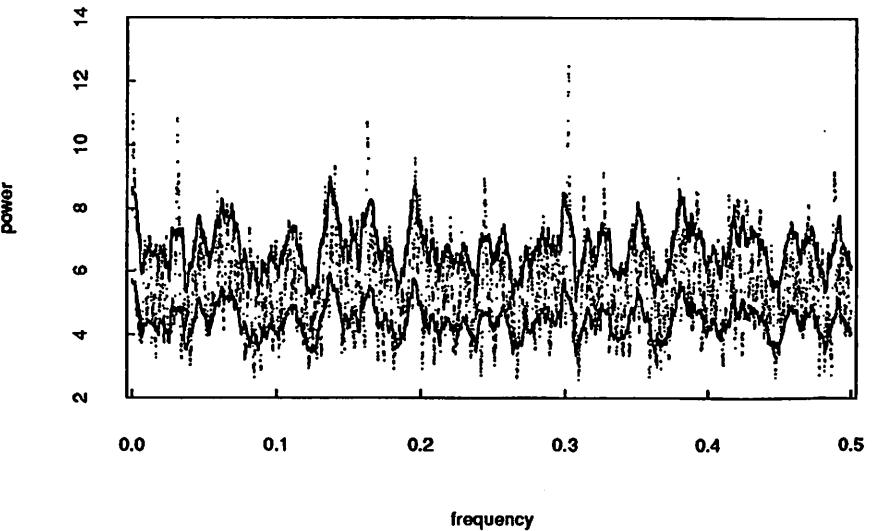
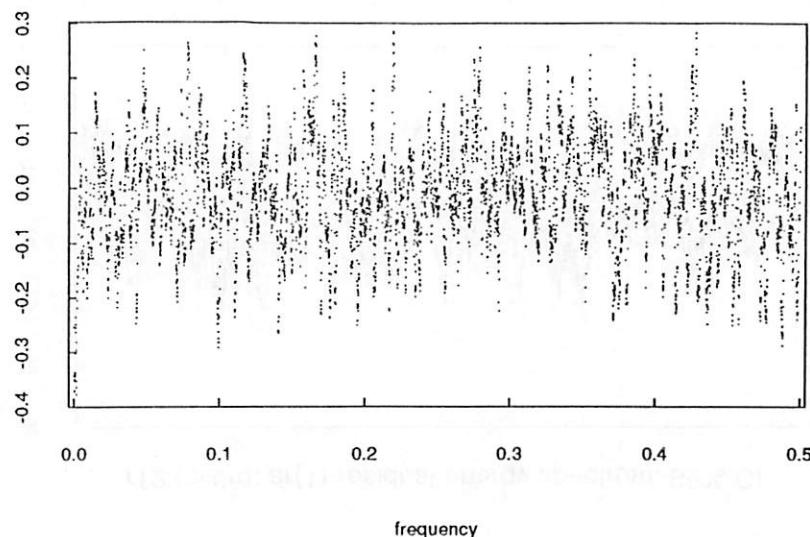
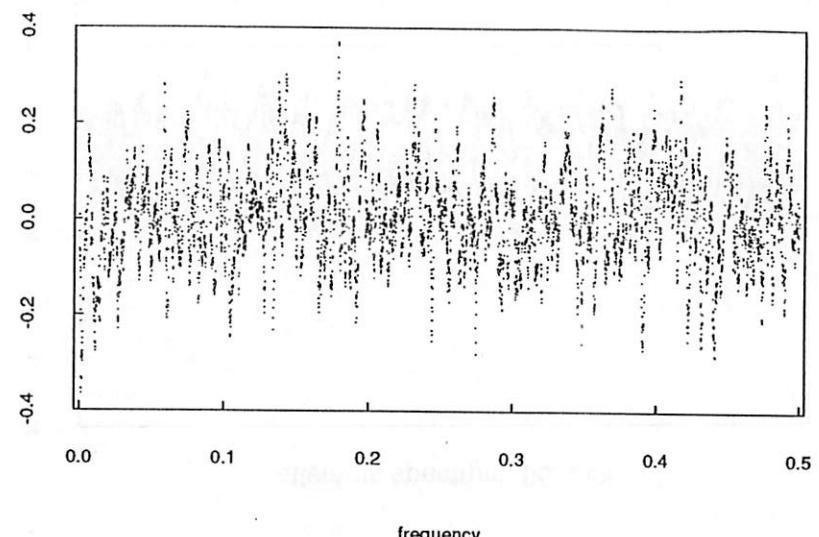


fig 2

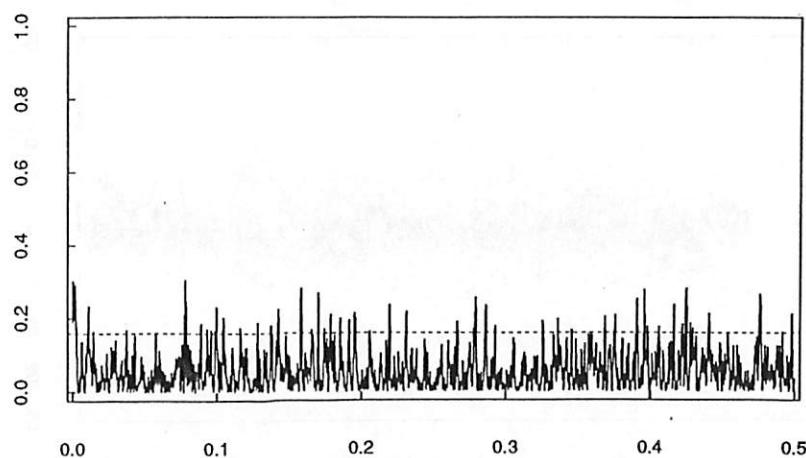
r12 (plain): transfer function: real part



transfer function: imag part

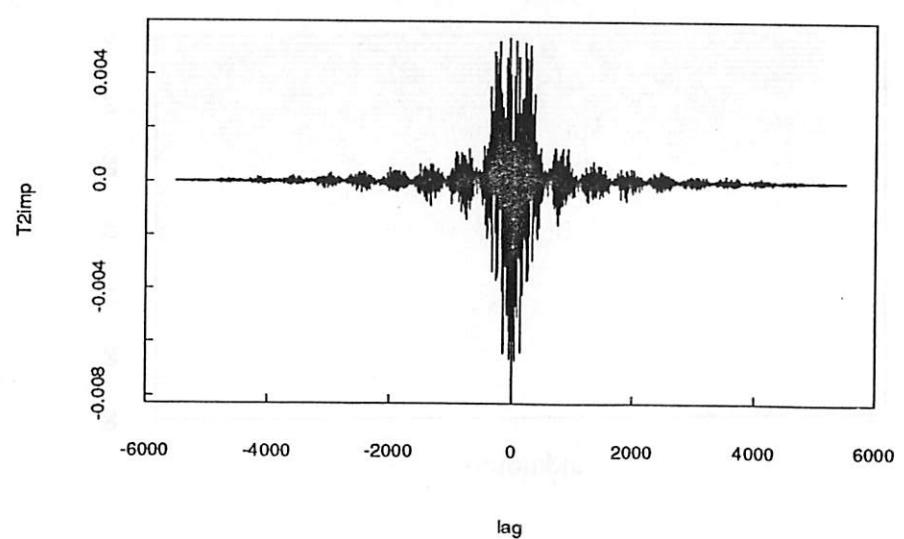


coherence, CI=0.95



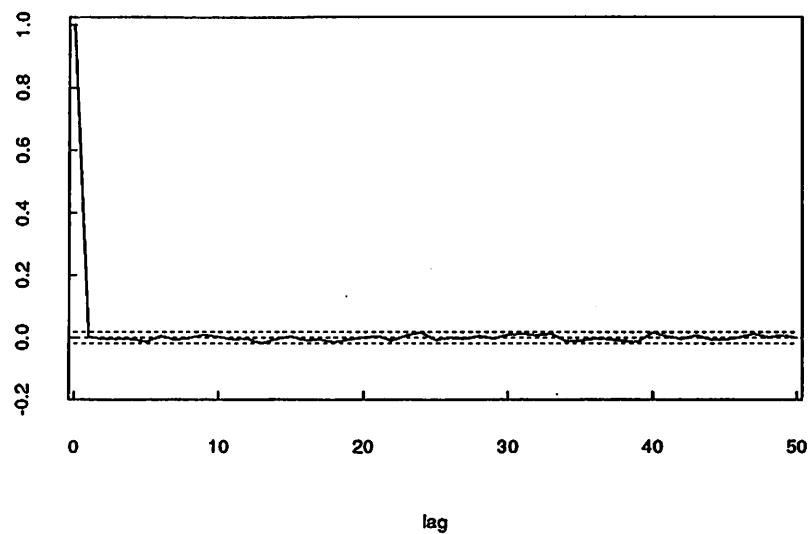
4.9% exceed null

impulse response

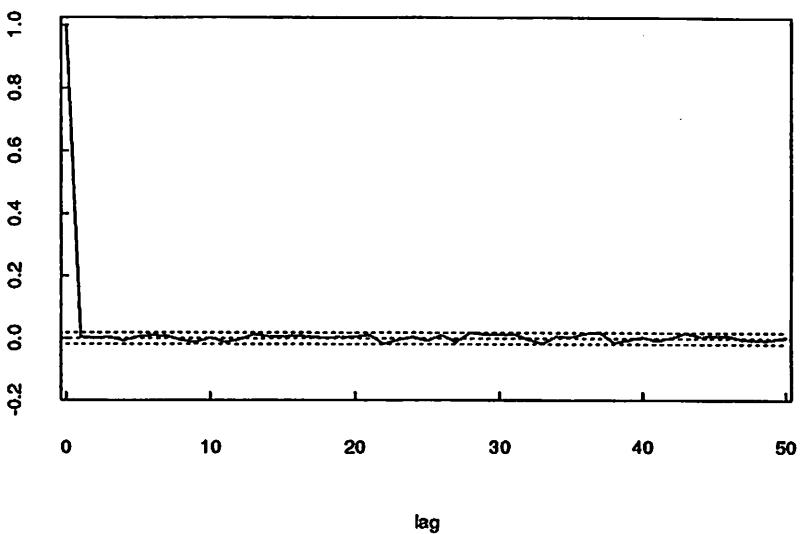


f.9.3

r12 (plain): R<sub>xx</sub>



R<sub>yy</sub>



R<sub>yx</sub>

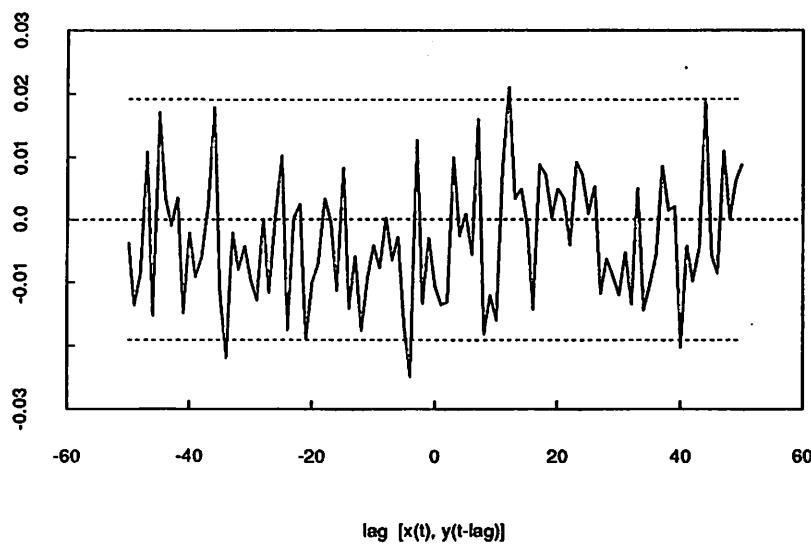
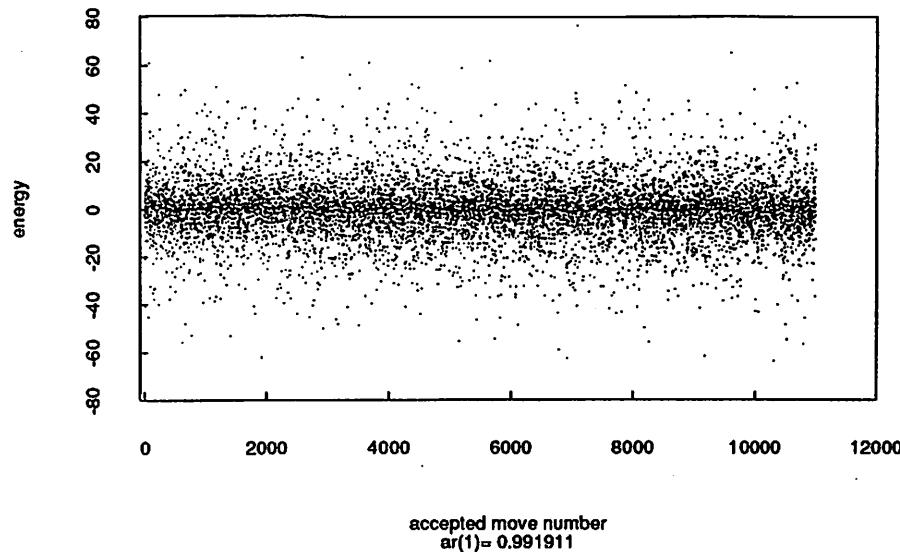
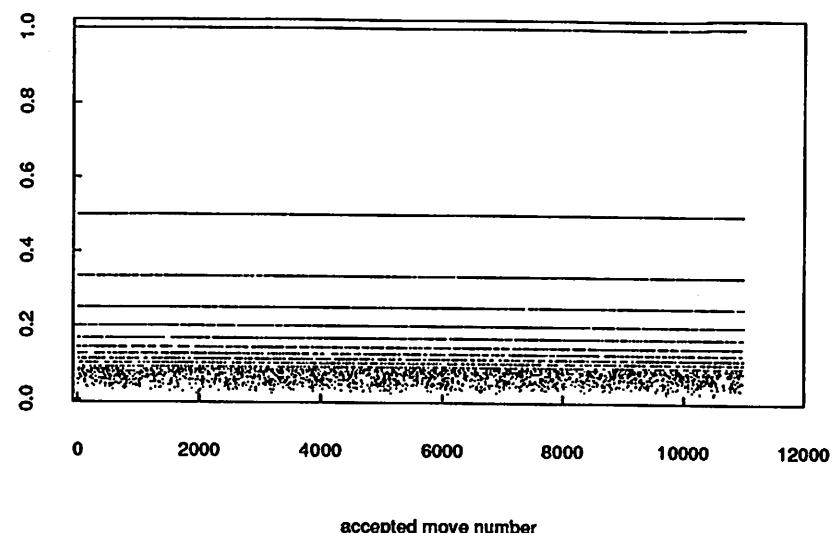


fig H

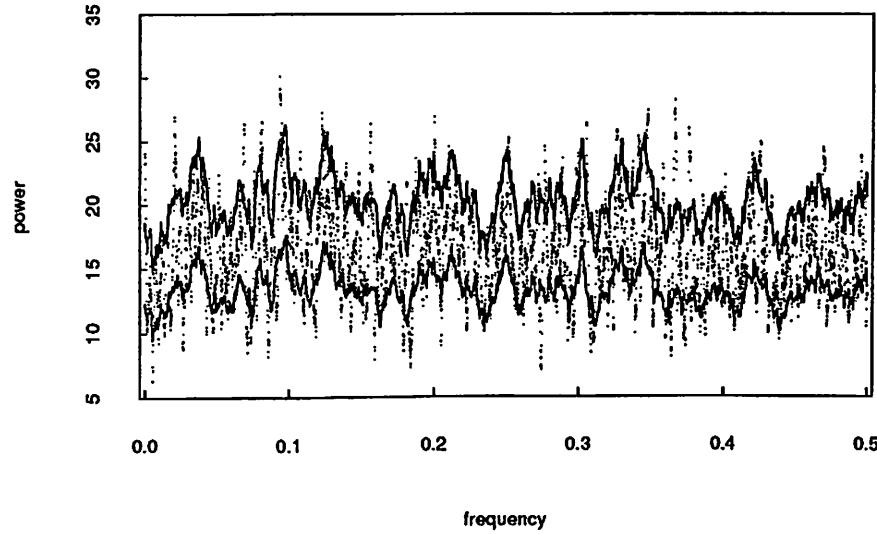
r12 (inverse) : ar(1)-residual energies



1 / #attempts



r12 (inverse): ar(1)-residual energy spectrum, 95% CI



attempts spectrum, 95% CI

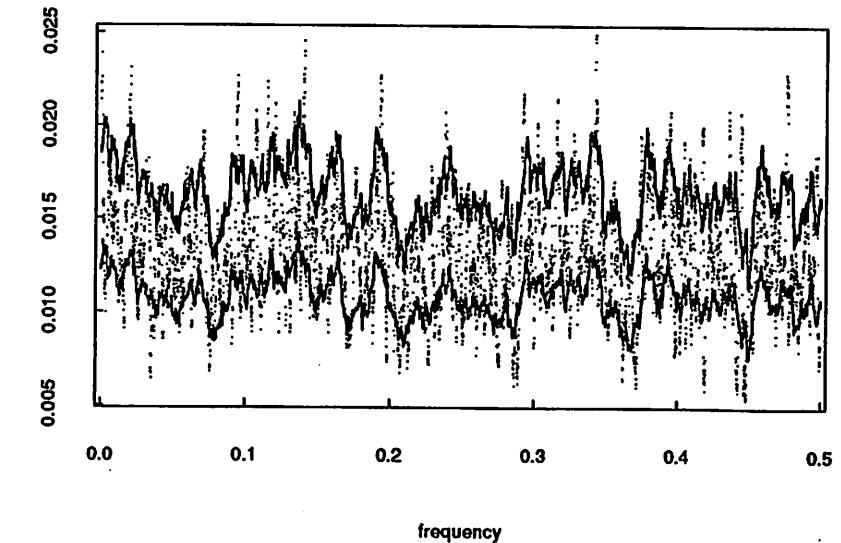
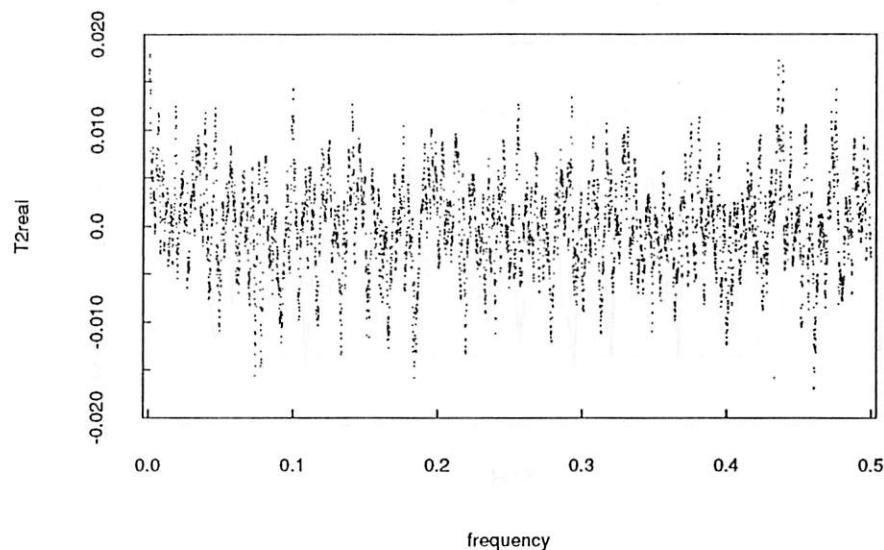
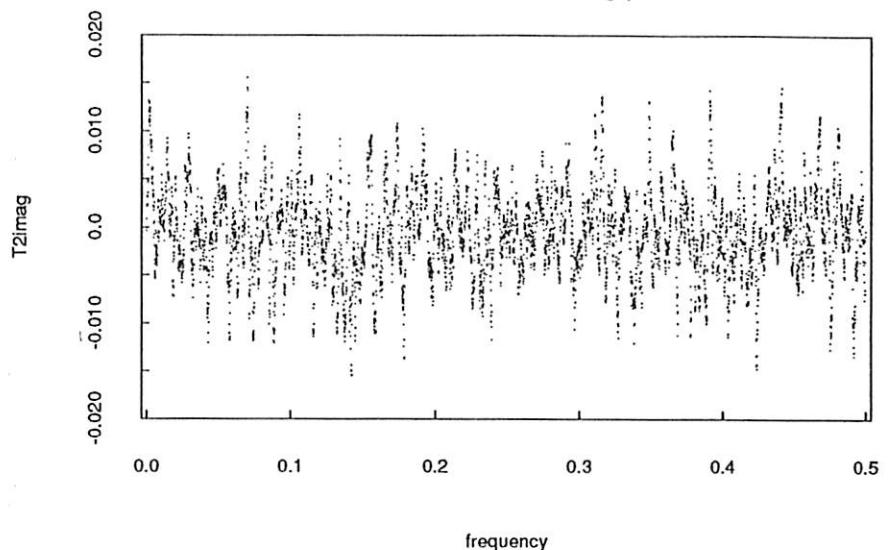


fig 5

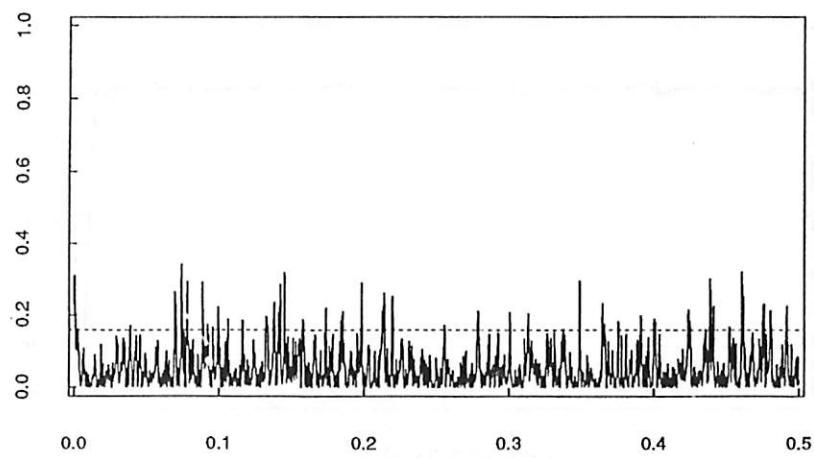
r12 (inverse): transfer function: real part



transfer function: imag part



coherence, CI=0.95



5.1% exceed null

impulse response

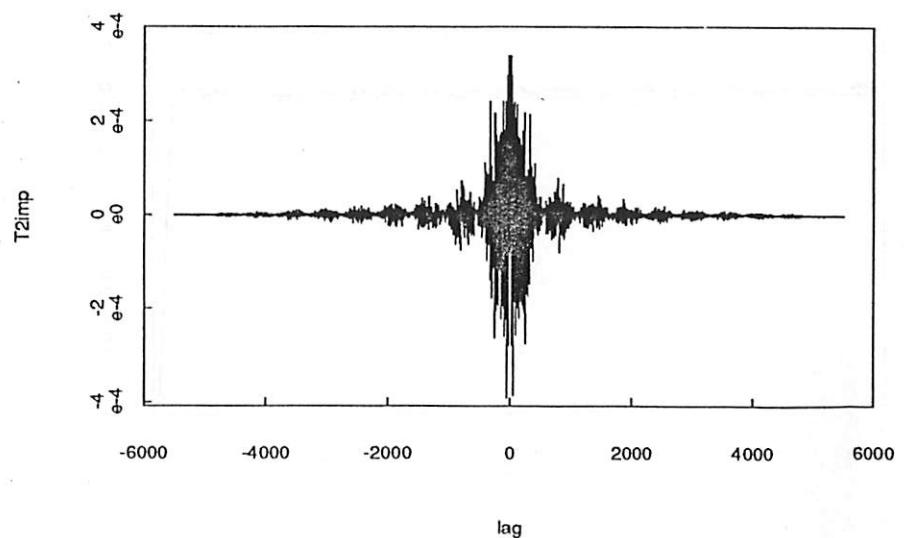
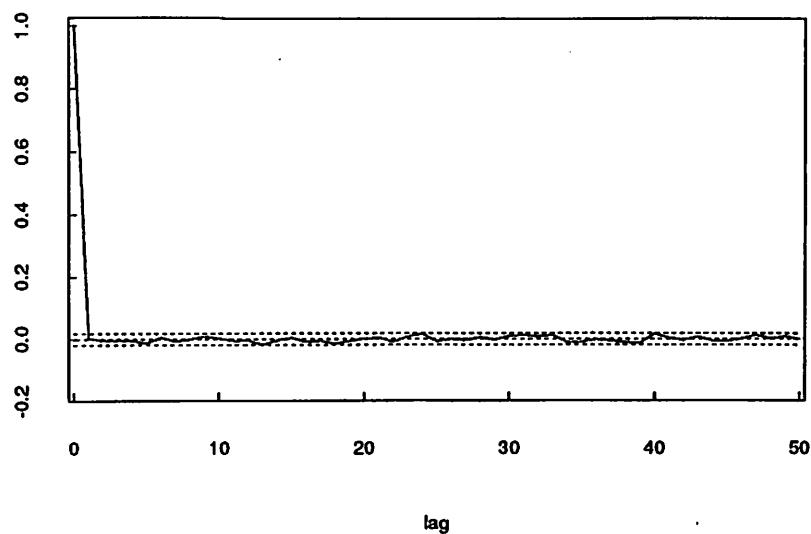
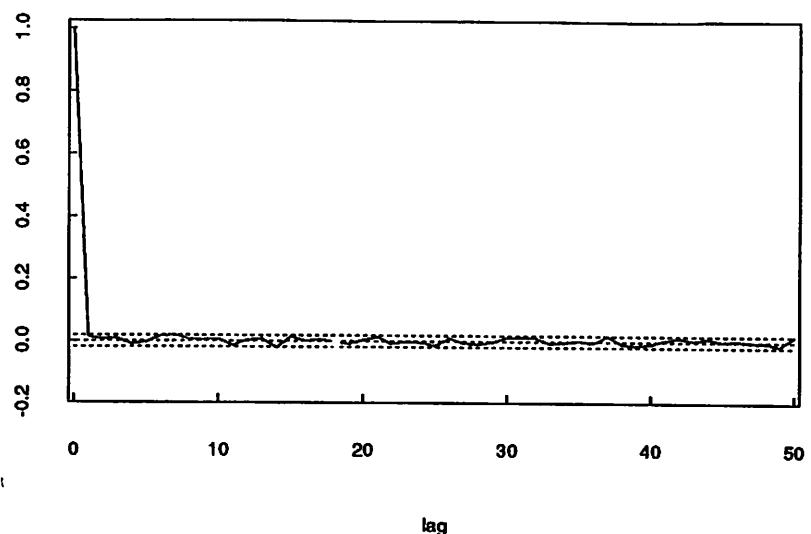


fig 6

r12 (inverse): R<sub>xx</sub>



R<sub>yy</sub>



R<sub>yx</sub>

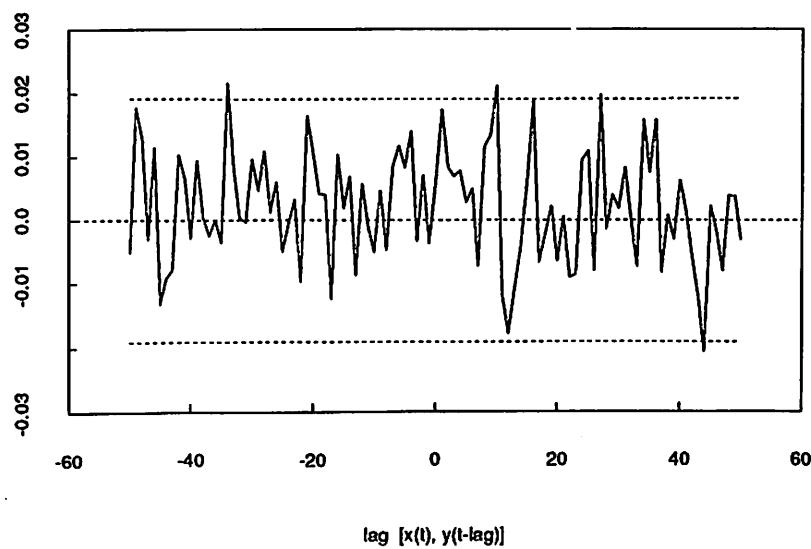
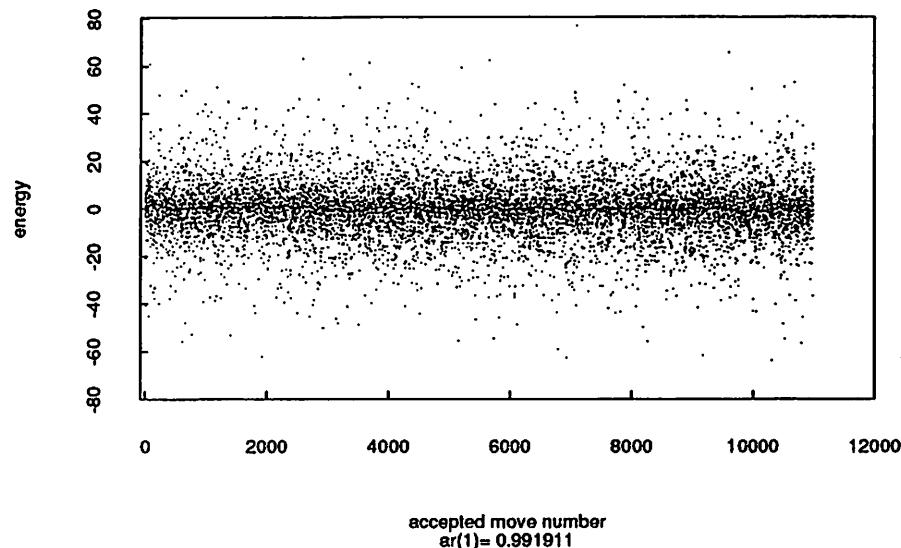
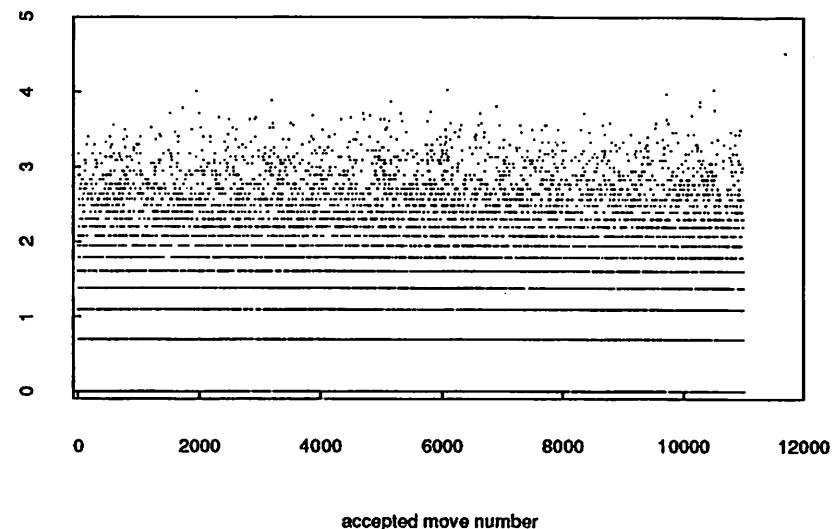


fig 7

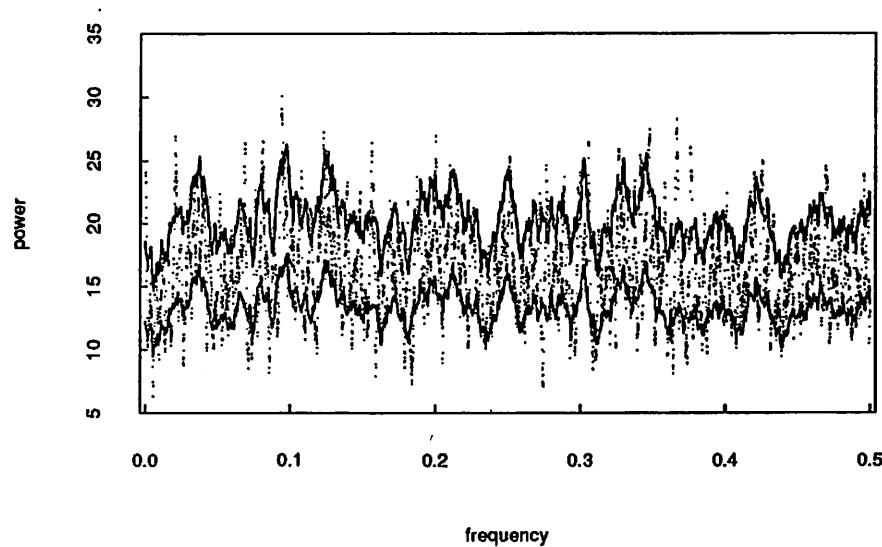
r12 (log) : ar(1)-residual energies



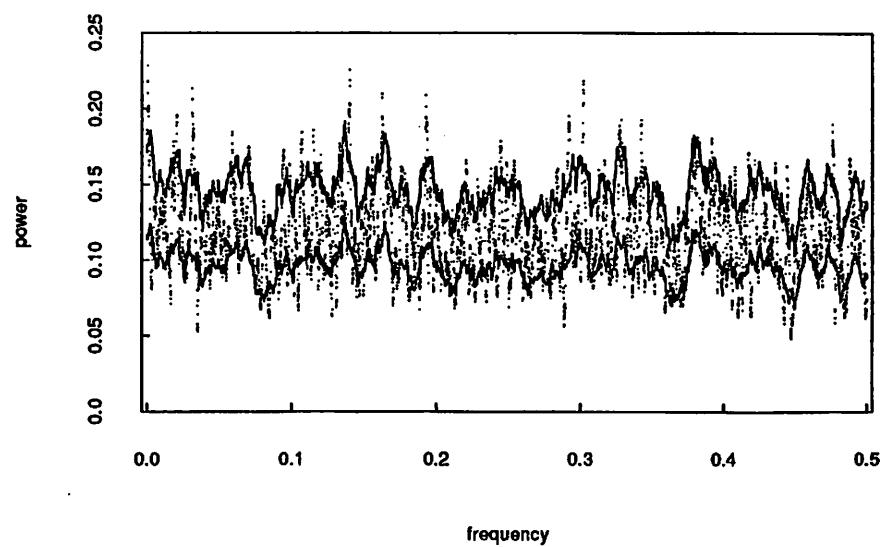
log(#attempts)



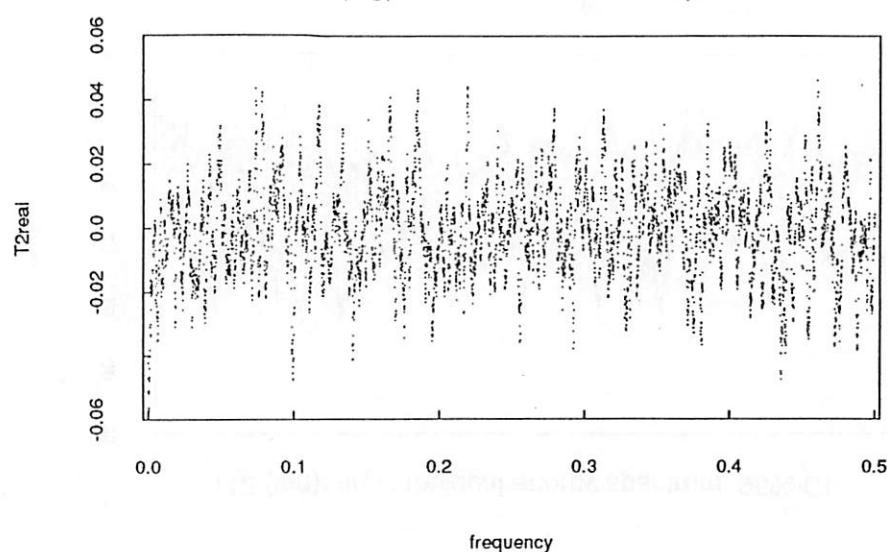
r12 (log): ar(1)-residual energy spectrum, 95% CI



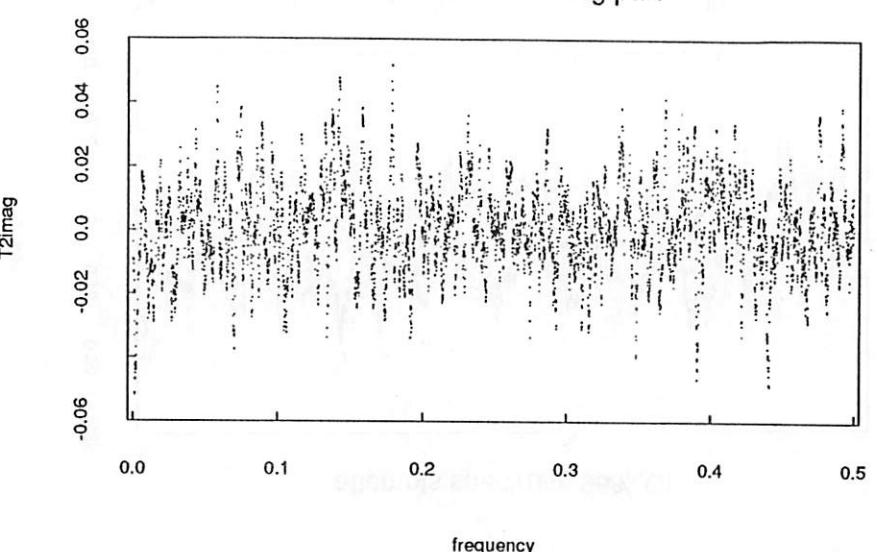
attempts spectrum, 95% CI



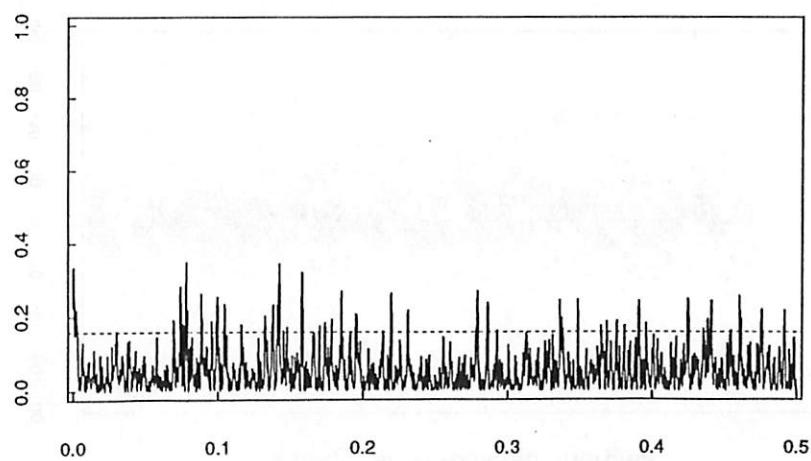
r12 (log): transfer function: real part



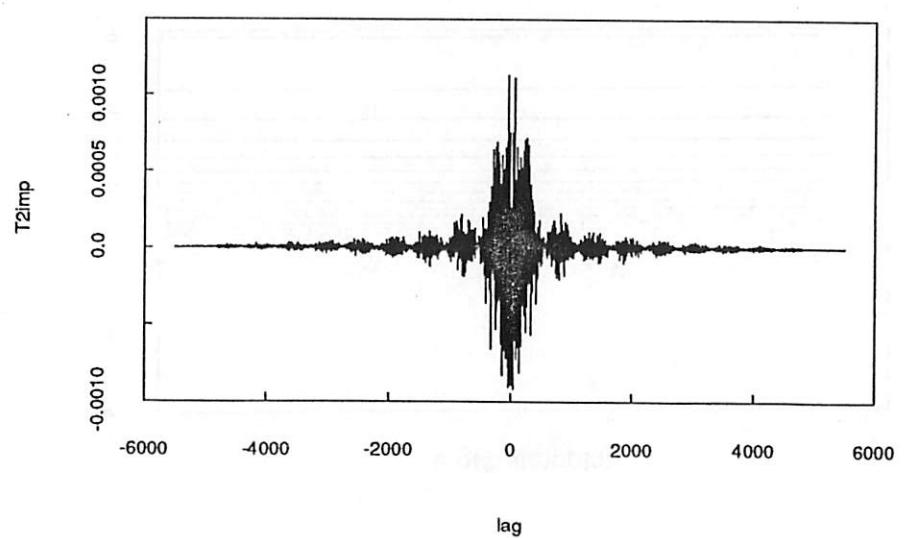
transfer function: imag part



coherence, CI=0.95

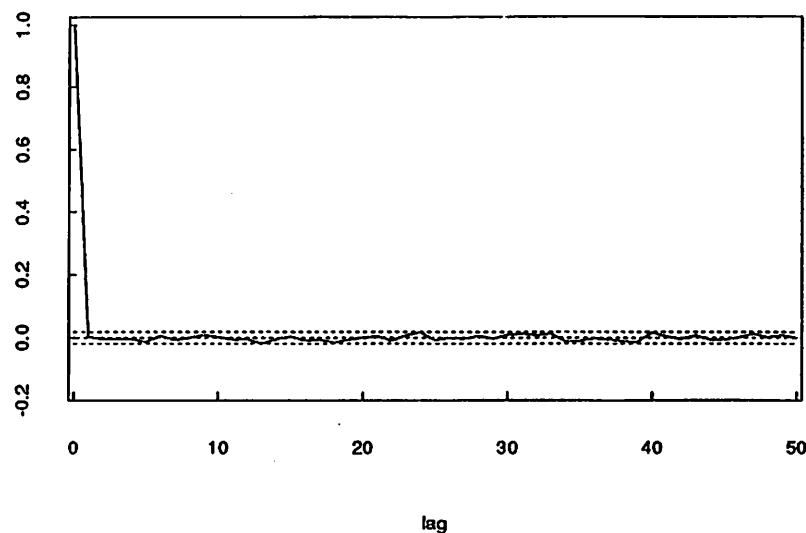


impulse response

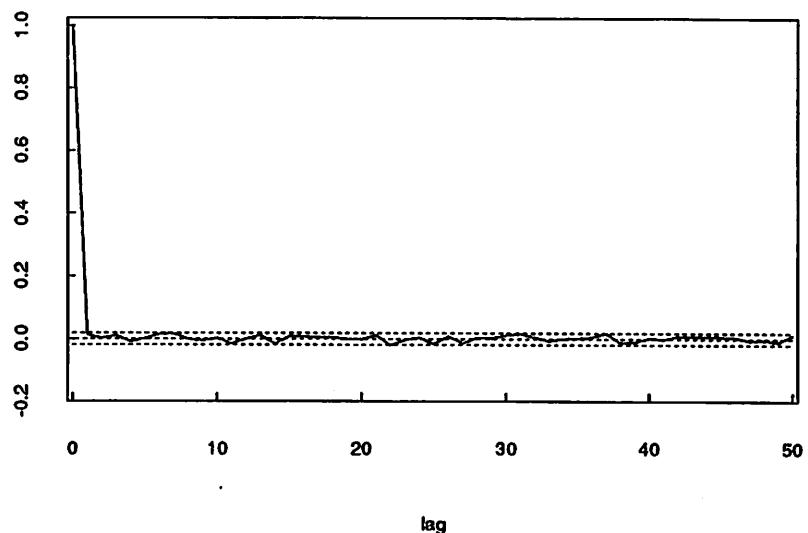


9  
c  
f.  
c  
d

r12 (log): Rxx



Ryy



Ryx

