UNIVARIATE TIME SERIES ANALYSIS OF SIMULATED ANNEALING DATA

by

Gregory Sorkin

Memorandum No. UCB/ERL M90/5

10 January 1990

COVER PAGE

UNIVARIATE TIME SERIES ANALYSIS OF SIMULATED ANNEALING DATA

by

Gregory Sorkin

Memorandum No. UCB/ERL M90/5

10 January 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

UNIVARIATE TIME SERIES ANALYSIS OF SIMULATED ANNEALING DATA

by

Gregory Sorkin

Memorandum No. UCB/ERL M90/5

10 January 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Univariate 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'. A typical application is a placement problem, say that of placing 100 circuits on a 10×10 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 will also define the distance between any two states as the length of a shortest path connecting them.

The simulated algorithm is as given in figure 0.

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.

It can easily be shown that this process is equivalent to that of a random walk on the graph, with self-loop edges present, 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.

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 my research has been to characterize the energy landscape itself (the graph and energy function) and the behavior of the annealing algorithm.

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

Figure 0: 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'). The cost of a placement is the total Manhattan length of wire required to connect the circuits; a netlist defines which circuits are to be connected. Since the structure of the netlist may influence the nature of the cost function and the behavior of annealing itself, 3 different netlists were considered. The first, **r**, came from a real sample of VLSI logic. The second, **f**, specifies interconnections between randomly-selected pairs of circuits. That is, the graph whose vertices are circuits and whose edges are those circuits to be connected is a random graph. The third, **h**, was constructed in a random but hierarchical fashion. That is, the graph of circuits/nets has cut-edge sets of size $\propto m^{2/3}$ to isolate vertex sets of size m.

For each netlist, both 1- and 2- dimensional energy functions were examined.

Each netlist and energy function was annealed at a number of temperatures, and the resulting energies as a function of time constitute the data sets. The temperatures used were infinity (which is well-defined in the annealing context since it is really $1/T$ that is used), 100, 10, and 1. These temperatures were chosen to match the energy scale of these problems: typical energy changes are in the thousands, while the minimum nonzero energy change possible is 1.

Finally, time can be defined as the number of moves *attempted* or the number of moves *accepted.*

Thus there are 48 data sets available for analysis: {r,h,f} × {dim=1,2} $\times\{T = \infty, 100, 10, 1\}\times$ {accepted, attempted}. For the time being we have looked primarily at timeseries with time defined as moves accepted, though attempted move data has been looked at for one or two cases. The data sets have names such as 'r13': 'r' means the 'real' netlist, the first '1' means 1 dimensional placement, and the final digit maps to temperature by $0 \mapsto \infty, 3 \mapsto 100, 2 \mapsto 10, 1 \mapsto 1$. An 'a' as the final character of a dataset means time is defined by attempted moves rather than accepted moves.

Our analysis is guided by a few basic facts. First, there is reason to expect that annealing is useful on spaces which have a fractal energy structure. This belief is supported by a recent proof that for a family of deterministic 1-dimensional fractals, annealing can give good results, requiring run time polynomial in the size of the problem (log number of states) and in the 'quality' of the anneal – the inverse of the expected final cost, assuming the absolute minimum cost is 0 [2].

Unfortunately it is difficult to test directly whether combinatorial problems have fractal structure; even the definition of fractalness in this context is problematical. In Euclidean space, a random self-affine fractal $f : X \mapsto Y$, $X = \Re^n, Y = \Re$, has the defining property that as surfaces in $\Re^{n+1}$, $\{(X, f(X))\}$ and $\{(r_x X, r_y f(X))\}$ are statistically identical for appropriate scaling constants r_x and r_y. Because the space X of interest for annealing may be the permutation space previously described, the rescaling of X is not a simple matter, and the 'similarity' of a subspace of X to X itself is difficult to define.

However, an implication of the definition of fractalness is that for points $x_1, x_2 \in X$, $\mathrm{E}\{(\mathrm{y}_1 - \mathrm{y}_2)^2 | \mathrm{d}(\mathrm{x}_1, \mathrm{x}_2)\} \propto \mathrm{d}(\mathrm{x}_1, \mathrm{x}_2)^{2\mathrm{H}}$ where $2H = r_y / \ln r_x$. This is a property that can be checked directly for the combinatorial spaces of interest. Results for the real netlist and 1 dimensional placement energy function are shown in figures 1–2. Random walks of length 200 moves were taken from some 500 different starting points, and the starting point paired with every fifth generated point defines the samples. Almost all point pairs in this permutation space on 105 objects have distance close to the average distance of about 99.5; the diameter of the space is 104, very close to the average distance. The 200-long walk was sufficiently long to generate point pairs at distances covering the full range, with each run generating pairs around 100 apart. Of course, each run in its earlier steps also generates pairs with distance about 5, 10, and so forth, so each distance range in the scatter plot is covered by at least 500 samples. Despite the overwhelming noise, it can be seen that the mean square energy differences of these random point pairs are approximately polynomially related to their distances.

Another property of fractals in Euclidean space is that a random walk on such a fractal yields fractal Brownian motion [4]. Fractional Brownian motion (fBm) is a fractal on $X = Y = \Re$, and is characterized by the parameter $2H$. (An example is the drunkard's walk, where $2H = 1$.) Fractals on permutation space, *defined* by the property $\mathrm{E}\{(\mathrm{y}_1 - \mathrm{y}_2)^2 | \mathrm{d}(\mathrm{x}_1, \mathrm{x}_2)\} \propto \mathrm{d}(\mathrm{x}_1, \mathrm{x}_2)^{2\mathrm{H}}$, also satisfy the property that random walks yield fBm, with the power law relation holding at least for short time scales. Since the power spectrum of fBm satisfies $f_{YY}(\lambda) \propto 1/\lambda^\beta$ with $\beta = 2H + 1$ [4], it is easy to check whether a time series is fBm. It will be shown that the data sets analyzed here do appear to be fBm. In fact, as their β seems to be in the range of 1.7 to 2, corresponding to H around .35 to .5, very likely they are simple random walks. This point will be discussed a bit more, but for the theoretical analysis of annealing in [2, 1] it is only important that some power law relation holds; it is immaterial whether the process is fBm or simple Brownian motion.

Finally, it is worth noting that in the definition of the annealing algorithm, only the energy changes matter: the absolute value of the energy function is irrelevant. (The same property is almost always true as well of computer programs for annealing, which compute the energy change directly rather than by computing the energy itself and subtracting.) Thus it may be natural to consider the first differences of the timeseries rather than the series itself.

In particular, we might hope that these differences are uncorrelated noise. A little thought reveals that this cannot quite be the case. Because of the finiteness of the problem space, annealing at a given temperature is (after equilibrium is reached!) a stationary process; uncorrelated first differences would give nonstationarity. Stationarity does hold if the process is defined not by $X(t) = X(t-1) + \epsilon(t)$ but by $X(t) = \alpha X(t-1) + \epsilon(t)$ with α close to 1 but $\alpha < 1$. Such a first-order autoregressive model will be shown to describe the data well.

3 Methodology

Figures 3-8 show the raw energy vs. accepted move data for each dataset. Because each run was started in a fairly low energy state, the runs at high temperatures take some time to reach equilibrium. Because the initial state was not of extremely low energy, some of the runs at low temperatures also took some time to equilibrate. Each run consisted of 15,000 accepted moves. Visual inspection of the datasets was used to estimate starting times by which the process seemed to be stationary, ranging from 2000 to 4000. (The h11 dataset continues to show a downward trend and large, low-frequency oscillations suggesting nonstationarity, and indeed the estimated α had value greater than 1.) Starting and stopping times for the datasets are indicated in the same figures; stopping times were 15,000 in all cases. The data before the start time were discarded and time was shifted to start from 0; it is this data that will be considered henceforth.

Because we were looking for evidence of fractalness we began with a periodogram estimate of the spectrum. There are a few choices to be made in the computation of the spectrum: whether to prewhiten the spectrum; what order smoothing to use; and how much of the data to taper. r13 was arbitrarily chosen as the guinea pig to resolve this. The first row of figure 9 (or figure 10, if one prefers a point plot of the periodogram) shows the results of using a few different tapers, all with no smoothing to avoid losing detail. We note that there does not appear to be any particular period in the data: no peaks stand out in the unsmoothed periodogram. As the differences caused by taper values seem minimal (not surprising considering the length of the series), a taper of 0.1 was chosen. The second row of the figure shows the results of various smoothers in combination with the .1 taper. Order 10 seems to be too small, leaving a substantial amount of noise. Order 100 can be seen to shift the apparent breakpoint between where the spectrum is constant and where it is power law. Because the spectrum is basically power-law, the large values are much larger than the nearby smaller ones, and, as long as there are any of them present, they dominate the average in the smoothing. Order 20 appears to be a reasonable compromise. Thus, tapering 10% of the data and smoothing the periodogram with a 20-point moving average was taken as the default method of estimating the spectrum.

Figure 11 compares the spectrum computed by this default method with the spectrum obtained from prewhitening the data by taking residuals after fitting an AR(1) model. The first frame shows the spectrum computed the standard way. The second shows the spectrum of the filtered (whitened) series, which was computed with taper .1 and smoothing order 20. It is plotted on a linear scale to emphasize its apparent uniformity (whiteness), indicating that the prewhitening should be successful. The third frame shows the frequency response function of the filter. Note that because the autoregressive parameter is near 1 but not exaclty 1, the response is not quite 0 at frequency 0, so the division of the whitened spectrum by the response function is possible (though in any case the power at frequency 0 is not visible on the log-log scale used). The final frame shows the spectrum obtained by the prewhitening. Figure 12 superposes the spectrum computed the default way (solid) with that computed by prewhitening (dashed), and they are seen to be virtually identical. Figure 13 is of the same type as figure 11, and confirms a similar result for the timeseries f12. Given this fact, for simplicity the rest of the spectra were computed without prewhitening.

For the 1-dimensional data, figures 14-16 show the spectral estimate, the non-constant portion of the spectrum estimated by an autoregressive fit of the logarithm of power against that of frequency, and the spectrum from an AR(1) model fitted to the data.

The power-law fit was done using points where spectral density was a factor of 10 or more below the density of the constant portion (the maximum density) as this was an easily systematized and consistent procedure which seemed to attend to the correct portion of the spectrum. This fit, interpreted as seen on the log-log scale, is influenced more by the high-frequency end of the spectrum than by the low-frequency end, as there are more points packed into that region by the log function. It is unclear what the 'proper' fit for the power law would be, and by inspection, the fit done seems close to what one would hope for, so no further refinements were made. Remember that a slope of -2 would be indicative of a random walk, within the limitations imposed by the finiteness of the space. The actual measured slopes cf about -1.8 seem ambiguous.

One property of a random walk, or of fBm with $\beta = 2$, is that $X(t)|X(t-1)$ is independent of $\{X(t-2), X(t-3), \ldots\}$. This can be tested directly by the partial autocorrelation coefficients and so the partial autocorrelation function (pacf) was computed. This information was used to round out the frequency-side analysis: the third line superposed in figures 14–16 is the spectrum computed from the pacf and the prediction error. It is a close fit to the measured spectrum.

Because the power-law fit was deemed less important than the AR(1) spectral fit, it is not shown for other than the r10, h10, and f10 samples of figures 14–16.

Instead, a different set of statistics is collected into each of the 24 figures 17–40 (one figure for each accepted-move timeseries). The first frame of each figure each shows the data itself (between times *start* and *end*.

The second frame shows the spectrum and the spectrum of a fitted AR(1) process (Splus's program ar was used).

The third frame displays the pacf itself up to lag 50. Independently, the pacf was also computed using the Akaike information criterion (AIC) to determine the order; the order returned was 1 in most cases, though it was 2 for a good number. Since the AIC is something of a black box (at least at this point), the display of the pacf to the relatively large lag of 50 gives a better feel for what is going on. The pacf(1) values are all close to 1; as previously mentioned they cannot quite be 1 for a stationary process. By contrast, all later values of the pacf are comparitively tiny, generally in the range of 0.01 and inconsistent in sign. Confidence intervals for the pacf's were not computed. However, their standard deviations should be $O(1/\sqrt{T})$ – roughly O(0.08) for timeseries of this length. That and the seemingly random mix of positive and negative values suggests that the true pacf does fall to 0 for lags greater than 1.

If the pacf is α for lag 1 and 0 elsewhere, it follows immediately that the acf at lag u is α^u. Thus in these cases the acf decays very slowly (this was confirmed by loooking at it for a number of the datasets), and is not interesting until viewed over lags over 100. The acf tools in Splus compute the direct estimate, and thus take time $O(\text{maxlag} \times T)$ which proves to be prohibitive for lags over about 50. The frequency-domain estimate of acf was coded as an S macro, but for reasons as yet not understood gave incorrect results. Because of these practical difficulties in estimating the acf over an interesting range and because the form of the acf is clearly dictated by the pacf, we have not shown the acfs.

The next step in evaluating the quality of the AR(1) model suggested by the very crisp pacfs is to compute the residuals. [Incidentally, it was verified experimentally that the pacf, autoregressive parameters, and residual produced by Splus's ar function are unaffected by the mean of the data.] Residuals for dataset r10 are shown in figure 41; they are not shown in general because their noisy appearance does not lend itself to meaningful interpretation.

Instead, the fifth frame on the figures shows the spectrum of the residuals; this is of course the same as the 'whitened' spectra constructed earlier for the two datasets used to test our spectrum estimation procedure. These residual spectra are shown on a linear scale because the spectrum should be uniform noise from end to end, a feature distorted by logarithmic plotting.

In the sixth and seventh frames are shown the pacf and the autocovariance function are of the residuals. If the residuals are indeed pure noise the pacf should be 0 for all lags greater than 0, as should the autocovariance. In fact the pacfs are generally of magnitude less than 0.02, with sign varying seemingly randomly, and our interpretation is that they are simply noise. We interpret the autocovariance similarly, as it is miniscule compared with the variance. Error estimates for the pacf and acf should be used as a more scientific test for whether their measured values are consistent with 0-mean noise; this is planned.

The crispness with which the pacf of the original timeseries falls to 0, the whiteness of the power spectrum of residuals, and the pacf and acf of the residuals, all seem to give strong evidence that the process is AR(1). One consequence of this is that in the region where the original spectrum obeys a power law, the power must be -2. The theoretical (not estimated) spectrum of any AR(1) has power = frequency^{-2} for higher frequencies, so it is important to resolve why the estimated powers were only about -1.8. Reexamination of figures 14–16 does show the deviation between the spectrum computed from the autoregressive parameters and the regression-fitted line; the former generally seems to better approximate the estimated spectrum. While the slope difference appears small, it in fact is precisely the discrepancy between -2 and -1.8. While the fit could perhaps be improved, it seems silly at best, and perhaps even dishonest, to try different fitting techniques, each equally reasonable *a priori*, until finding one giving us the result we now know we want.

Many results in annealing theory rely on knowing the density of states of the problem [5]. This is the same as the density of X for a $T = \infty$ annealing. It is generally assumed that the density of states is either Gaussian or Beta. The density at any other temperature follows from the $T = \infty$ density from Boltzmann's law, and in particular if the density is either Gaussian or Beta at $T = \infty$, it also has Gaussian (respectively Beta) density at all other temperatures. Quantile-quantile (qq) plots of each timeseries against a normal distribution (frames 4 of figures 17-40) indicate that this assumption is not too far from the truth, although the small deviations from normality are probably significant given the volume of data.

Finally, the distribution of residuals is that of the noise in the process, so qq plots were made of it too for all datasets (frames 8). The noise seems not to be Gaussian, a fact whose significance is not clear.

4 Attempted-move data

While some mention has been made of attempted move data, only a small amount of analysis has been done on it. The annealing program actually produces a vector of energies and a vector of attempted move numbers at which these energies were accepted; an S macro was written which expands this into a vector of energies for attempted move numbers $1, 2, \ldots$. Since at low temperatures the number of moves attempted before 15,000 moves are accepted can be in the hundreds of thousands, these vectors are restricted to a given maximum length.

This was done for r12 and r11; to produce timeseries r12a and r11a of lengths 32,768. The same analyses were applied to them (figures 42–43) and yield generally similar results, though of course the time scale is greatly expanded.

The approach was not pursued further at this time for two reasons. First, the volume of data is potentially even larger than that for the accepted move paradigm, and the processing is quite time-consuming. More importantly, it seems that other methods of analysis may be easier and more revealing. If the timeseries of the number of moves attempted until one is accepted were understood, that would probably lead directly to an understanding of the attempted-move process. For example, if the number of attempts is uncorrelated with itself (and with the energy), then one would guess that the attempted-move spectrum is identical to that of the accepted-move spectrum but for rescaling. In short, it may be more productive to treat this as an issue in bivariate time series analysis than to treat the attempted- and accepted-move processes as if they were unrelated.

5 Interpretation of Results

Simulated annealing is a random process which by construction is Markov on the state space consisting of all possible solutions of the given optimization problem. Viewed on the reduced state space defined by the energies alone, it is not clear whether the process will still be Markov. Some approaches to the automatic control of the cooling schedule get all their information from the energies seen, and implicitly rely upon the process being Markov in energy.

The partial autocorrelation function (pacf) provides a partial answer to the question of whether the process is Markov in energy: Restricting our attention to linear predictors of the energy, the fact that the pacf falls to 0 for times greater than 1 means that old energies do not improve the prediction of future energies, and the process is seemingly Markov. This is not a necessary nor obvious conclusion. A simple 4-state space can be constructed on which annealing at low temperature is not Markov in energy, and in general we might expect for example that observation of a number of low energies would indicate the process is in a low-lying region of the state space, so that future energies will also be low.

The characterization of the energy trajectory as an autoregressive means quite simply that, at least as far as energy is concerned, we know everything about annealing at a fixed temperature. If we could extend this understanding to the transition from one temperature to a slightly lower one we would have a complete characterization of the timeseries of energies as a function of the cooling schedule, which would enable computation of a good cooling schedule.

A characterization of this sort would rely on knowledge of the space such as that acquired here at great compuational expense (both in the annealing itself and in the data analysis) and thus may seem infeasible. However, one of the most important properties of the data apparent from this analysis is its uniformity. The different netlists all produce qualitatively similar timeseries data. So do the runs of annealing at different temperatures, where the spectra seem to have different power scales (i.e. they are shifted vertically on a log scale) but similar shapes and even similar roll-off frequencies. (Figures 44–49 show superposed spectra for each netlist and cost function at all 4 temperatures.)

This last fact is surprising. When annealing is run at low temperature, moves are accepted very rarely, and energy changes seen are small. While the fact that we are looking only at *accepted* moves means that the low acceptance rate has been factored out, the fact that energy changes are small suggests that many such changes need to be made before the energy limits of the problem itself are approached. Another way to put it is that the low-frequency power is related to the variance of the process. If the variance of energy at low annealing temperatures were the same as that at high temperature, then it would take longer for low-temperature anneals to reach this energy scale. The fact that the time required is about the same means that the energy scale is smaller.

This too may relate to a fundamental fractal property of the space: If the problem is fractal, restriction to a subspace corresponds to scaling down the energy. Thus if the temperature is scaled down, the timeseries may be just a rescaled version of that at higher temperature, the annealing process being essentially the same but with smaller energy and distance scales. This similarity would explain the similarity of time scale, though it is still not obvious why the roll-off frequency, which is related to the finiteness of the problem rather than to its fractalness over some energy/distance range, is approximately the same for all temperatures.

Another small mystery of these superposed spectra is that the $T = \infty$ and $T = 100$ spectal estimates often intersect. Since the energies are generally smaller at lower temperatures, along with their variances, one would expect lower-temperature spectra to lie strictly below higher-temperature ones, as is in fact the case for most of the temperatures. Whether this is significant, or even whether it is an artifact of the spectral estimation (as opposed to the true spectrum) is not yet understood.

6 Summary

A simulated annealing for placement was run on 3 different netlists, each at 4 different temperatures (including $T = \infty$, a random walk), and using two different cost functions corresponding to 1- and 2-dimensional placements. In each of these 24 cases, recording the energy after each accepted move yields a timeseries.

Another 24 timeseries could be obtained by recording the energy after each *attempted* move rather than just after the accepted moves. This was done for a couple of cases.

For each timeseries selected we performed a number of analyses. The timeseries itself was plotted to confirm stationarity and to reveal any obvious properties. A smoothed periodogram was constructed. In all cases it revealed a power law relationship between spectral energy and frequency, except at low frequencies where the spectral energy levels off to a constant. Such spectra are typical of real-world fractals where there is some limit to the frequency range.

The partial autocorrelation function (pacf) was constructed. In all cases the lag 1 value was close to 1, while the value at all other lags was near 0. This was taken as evidence that the process was a first-order autoregressive (AR(1)). Overlaying the power spectrum of the AR(1) model over that of the process itself supports this hypothesis. For more detailed verification, we took the residuals after this autoregressive fit and subjected them to analyses like those used for the series itself. The periodogram of the residuals appears to be purely noise, indication that the residuals themselves are uncorrelated. The pacf and acf of the residuals are close to 0 for all lags, indicating that no further ARMA modeling of the residuals is possible.

Thus the processes were concluded to satisfy an AR(1) model.

A couple of additional observations were made. For each timeseries, the energy values were examined on a q-q normal plot; they seem to be approximately normal. Since the distribution of energies is one of the central quantities used in the study of simulated annealing, it is useful to know that the energies do seem to follow this simple distribution. Also the distribution of residuals was examined in the same fashion. The fact that it does not seem to be normal means that while the process may be AR(1), the underlying noise process is not Gaussian white noise. The signficance of this is unknown.

Finally, for each netlist and cost function, the spectra of the AR(1) fitted processes (used as smooth versions of the true spectra) at all 4 temperatures were plotted together. Since they are all AR(1) spectra they are guaranteed to have a similar overall shape, but they are also seen to have similar rolloff points. This indicates a common time scale to annealing at any temperature, a surprising result.

Acknowledgements

Following my interest in simulated annealing, I did this work as the midterm project for a U.C. Berkeley course in timeseries analysis taught in Spring 1989 by Prof. David R. Brillinger. I am most grateful his guidance and interest. Following his suggestions, some of the weaknesses of this work (notably the absence of confidence intervals) are remedied in [3].

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. Bivariate time series analysis of simulated annealing data. Technical Report UCB/ERL M90/6, Univ. of California at Berkeley, January 1990.

[4] Richard F. Voss. Characterization and measurement of random fractals. *Physica Scripta*, T13:27–32, 1986.

[5] Steve R. White. Concepts of scale in simulated annealing. In *The Physics of VLSI (AIP Conf. Proc. No. 122)*, pages 261–267. 1984.

fig 1

E^2 vs. Distance

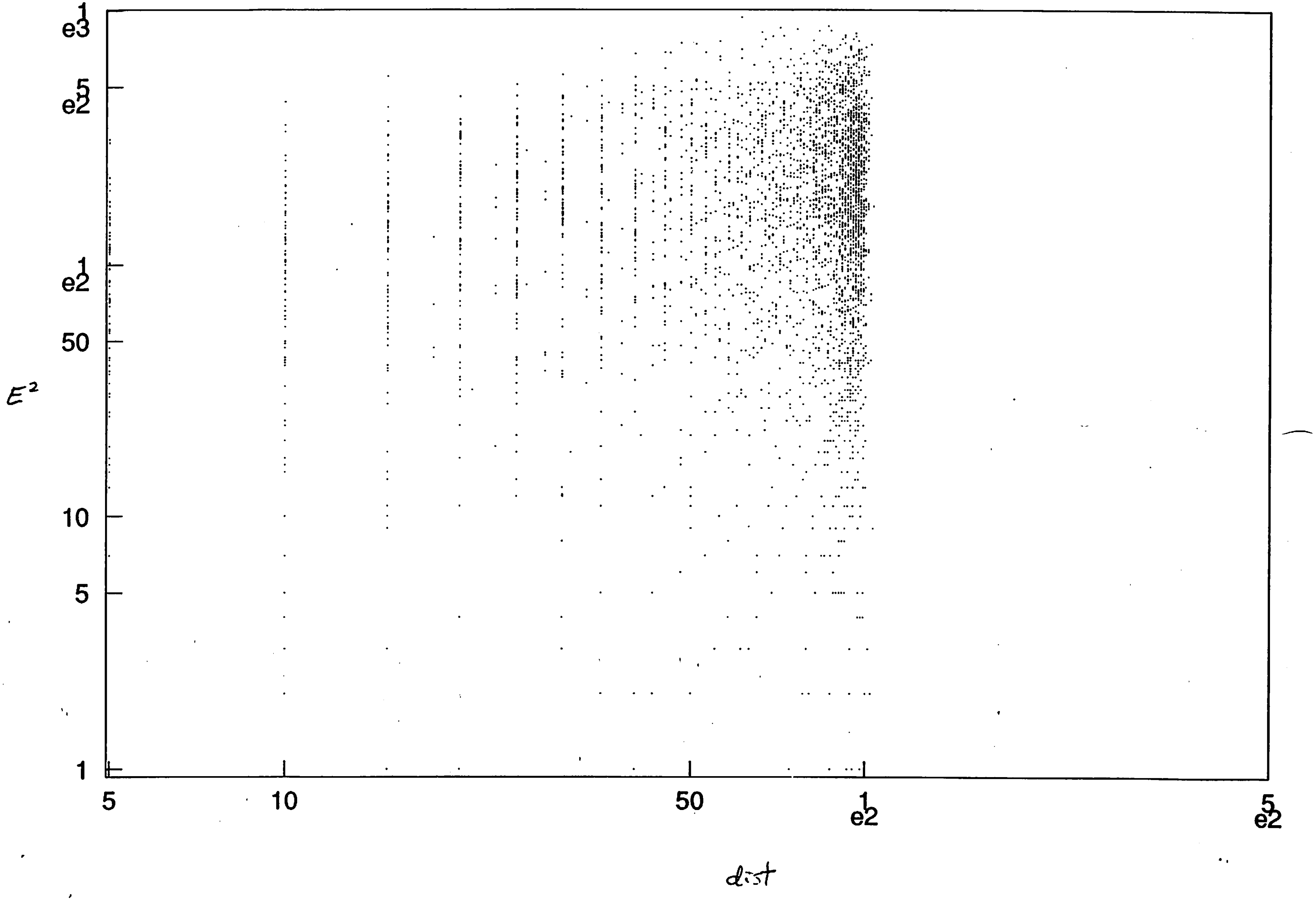
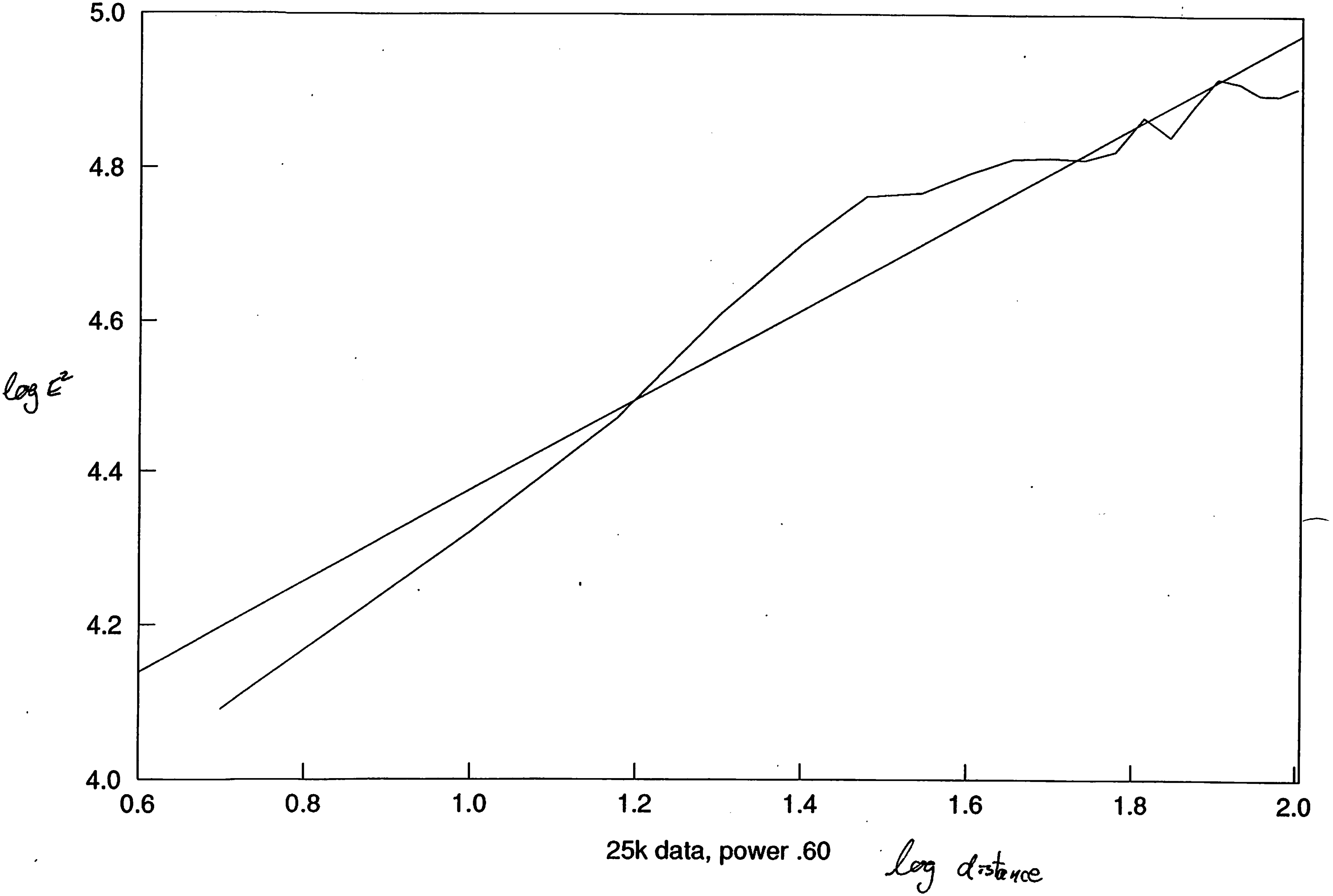


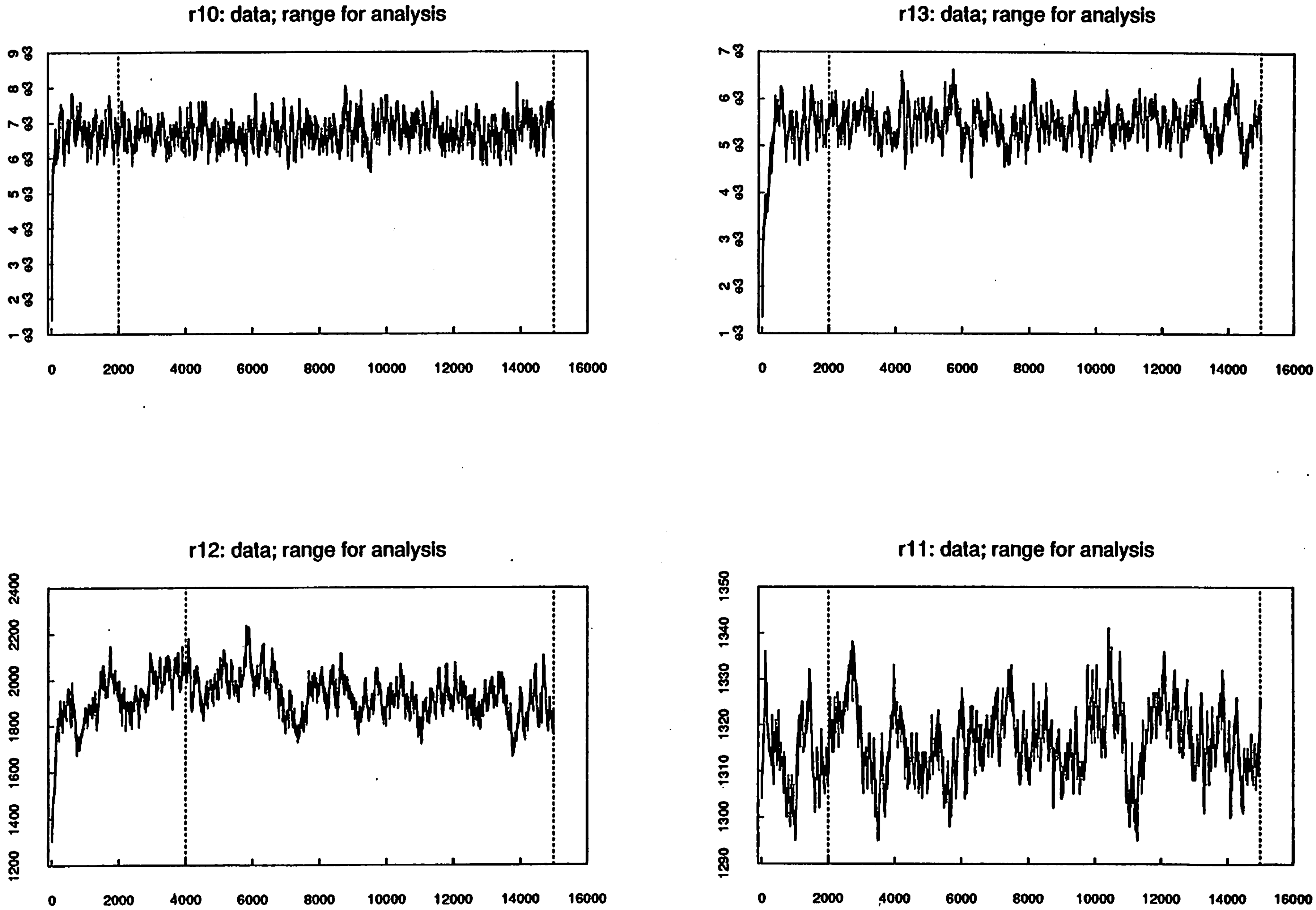
fig 2

mean sq. E vs Dist ; fit line



25k data, power .60

Fig 3

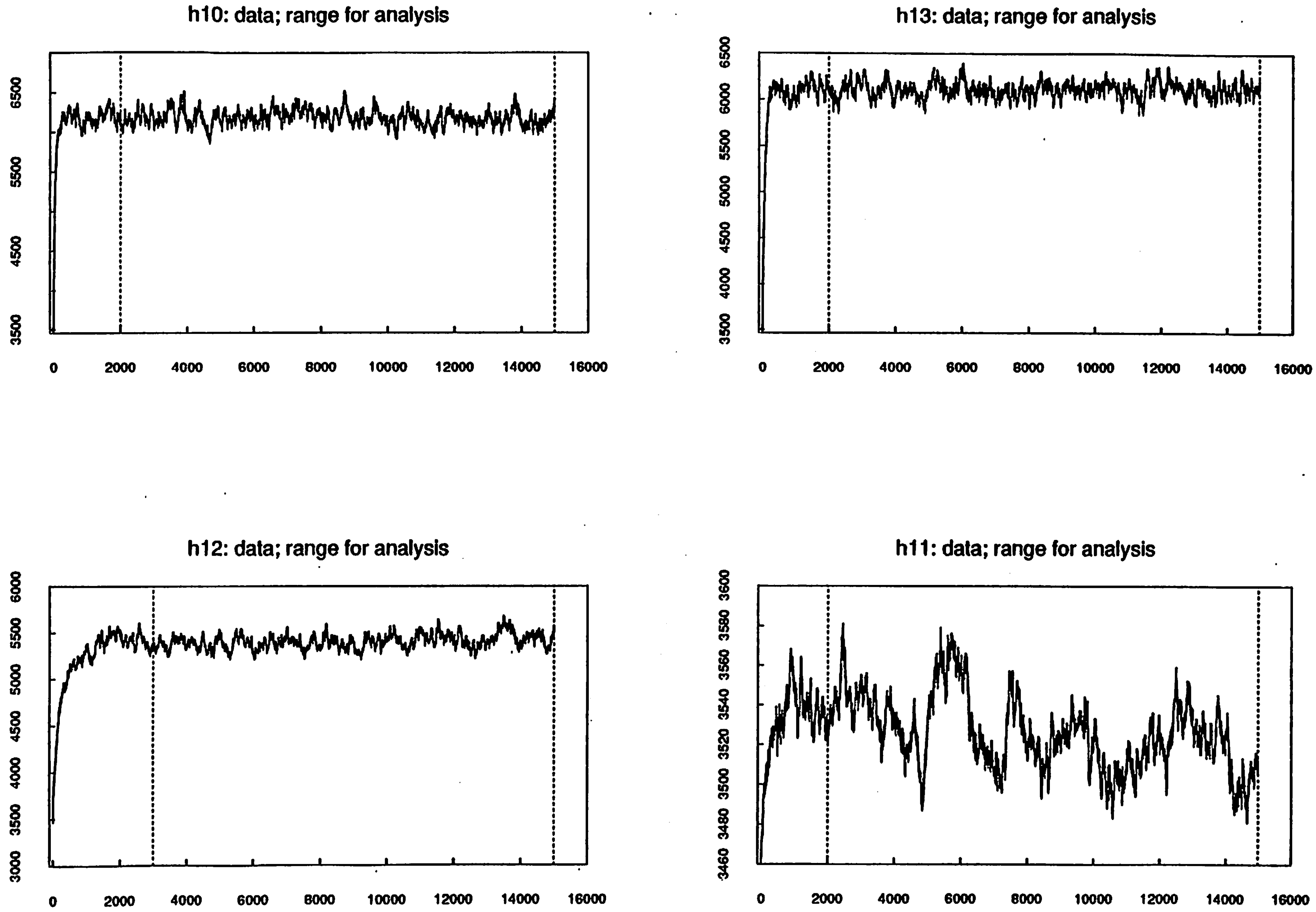

r10: data; range for analysis

r13: data; range for analysis

r12: data; range for analysis

r11: data; range for analysis

Fig 4

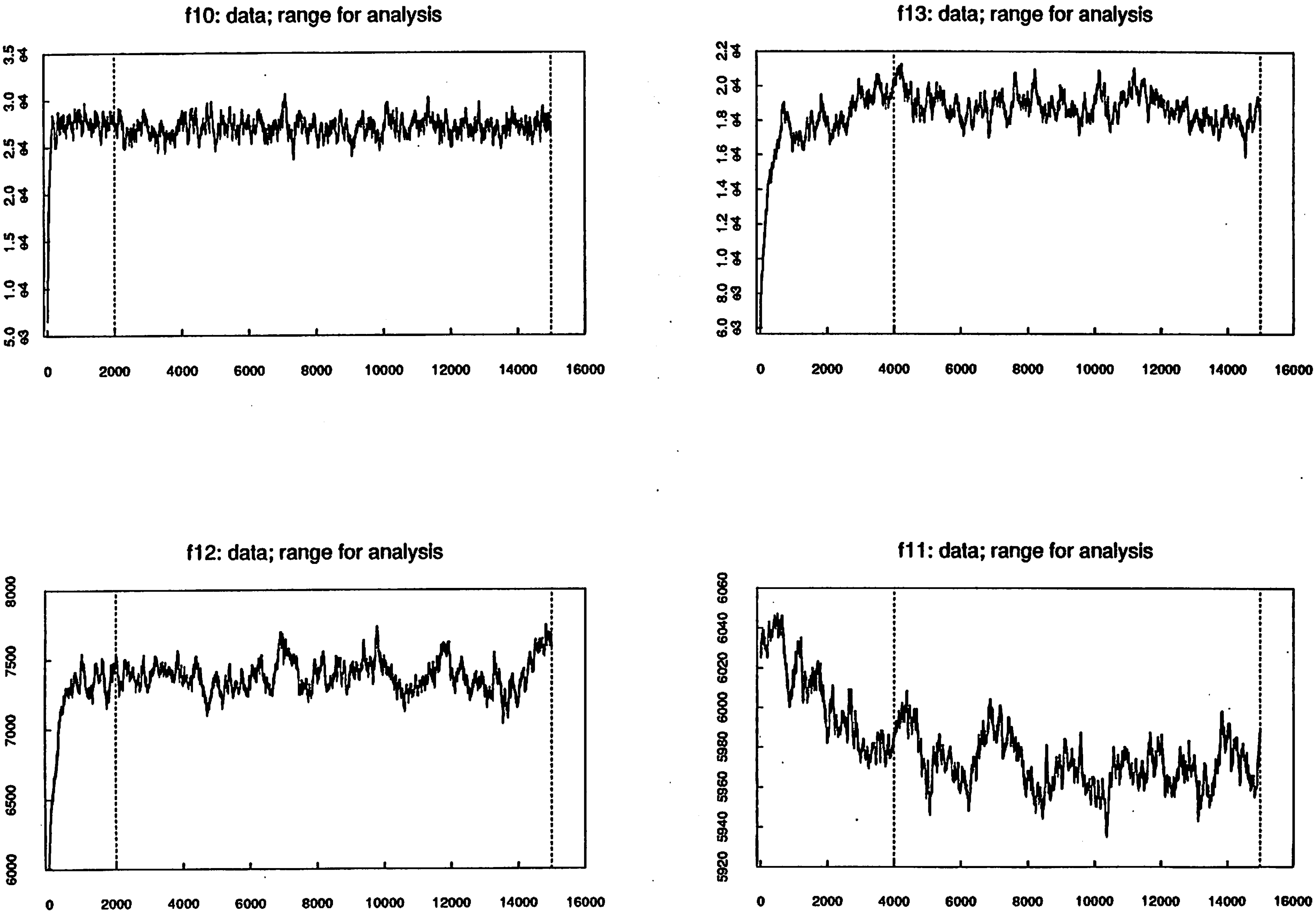

h10: data; range for analysis

h13: data; range for analysis

h12: data; range for analysis

h11: data; range for analysis

fig 5



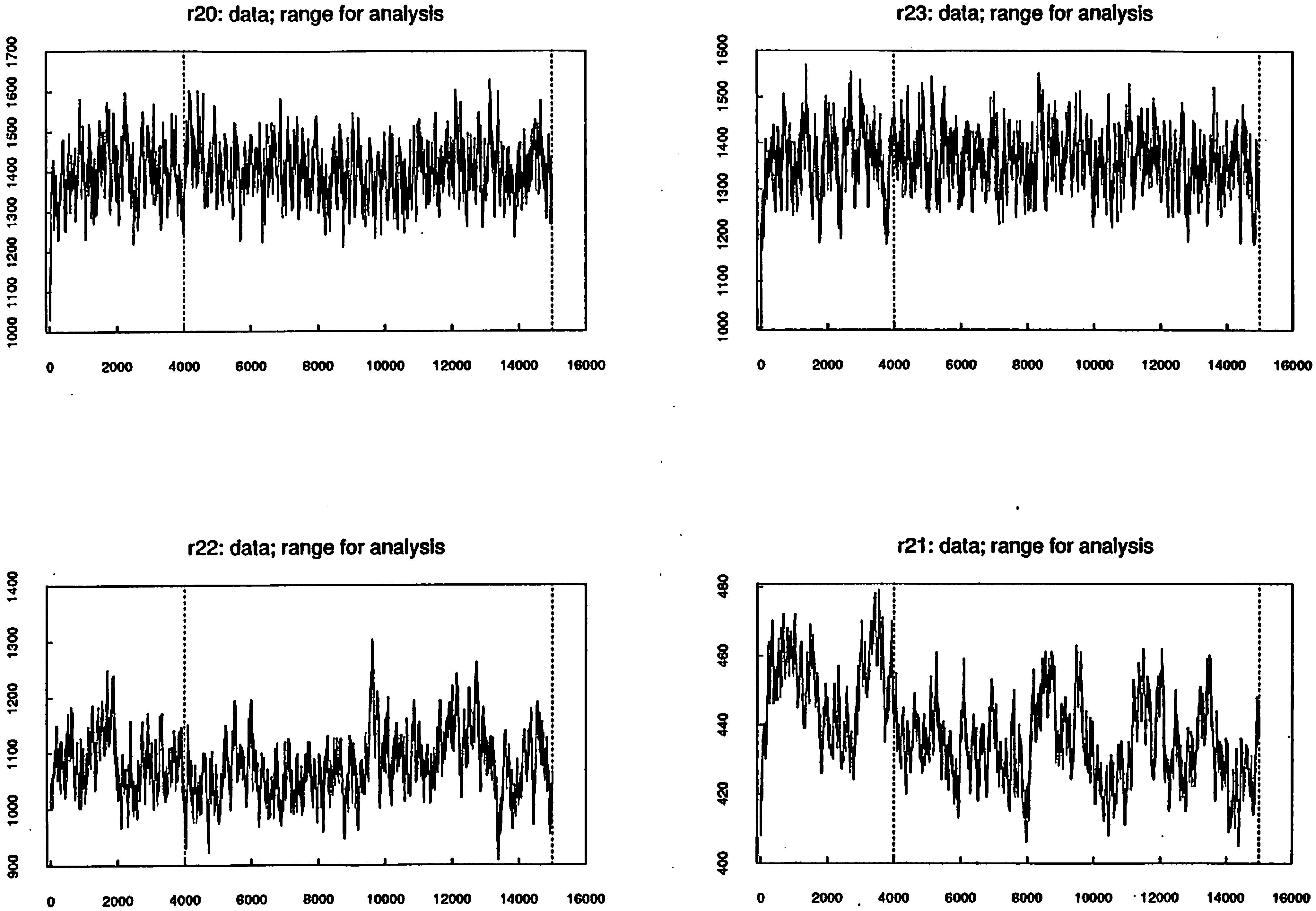
f10: data; range for analysis

f13: data; range for analysis

f12: data; range for analysis

f11: data; range for analysis

fig 6



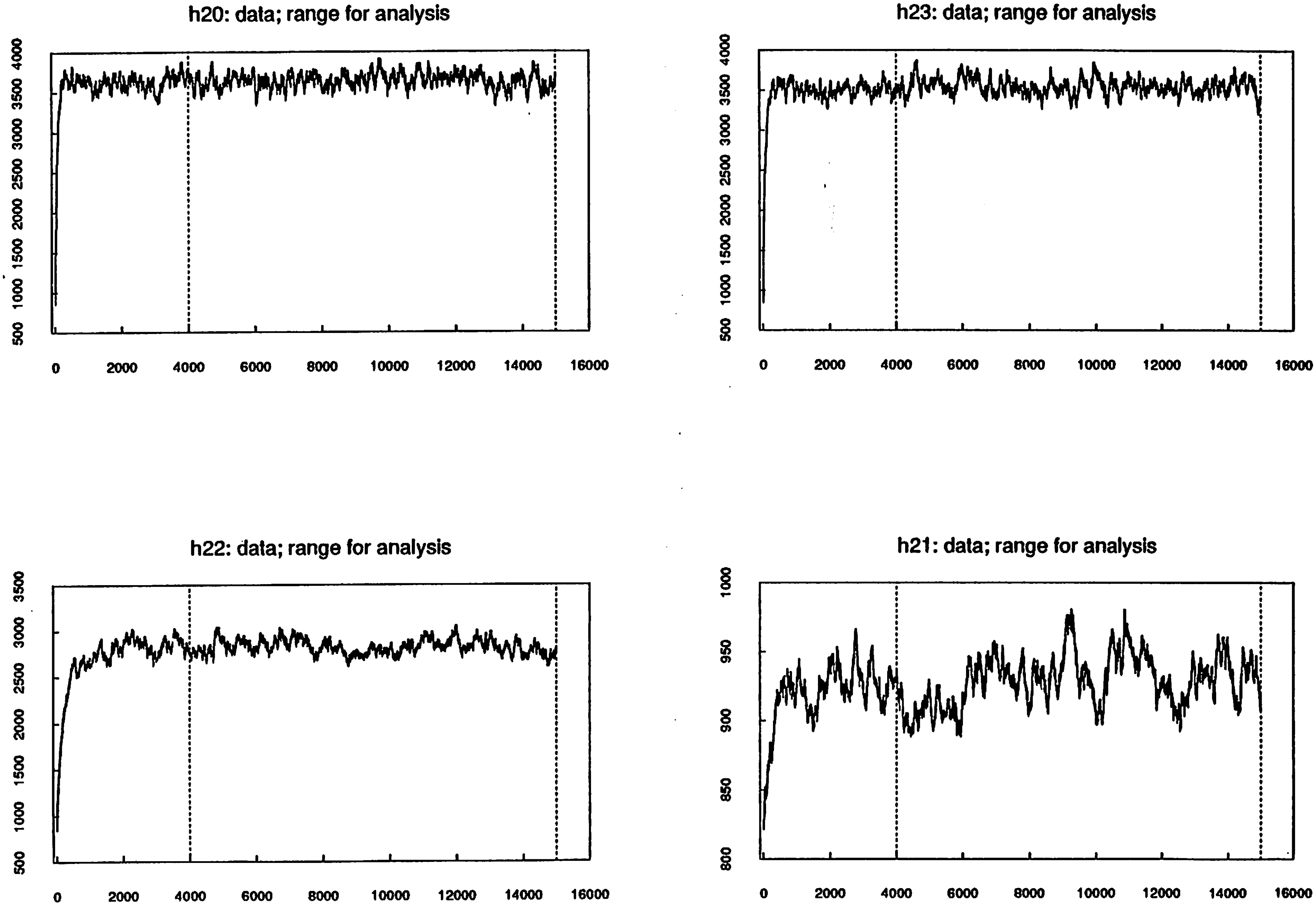
r20: data; range for analysis

r23: data; range for analysis

r22: data; range for analysis

r21: data; range for analysis

fig 7



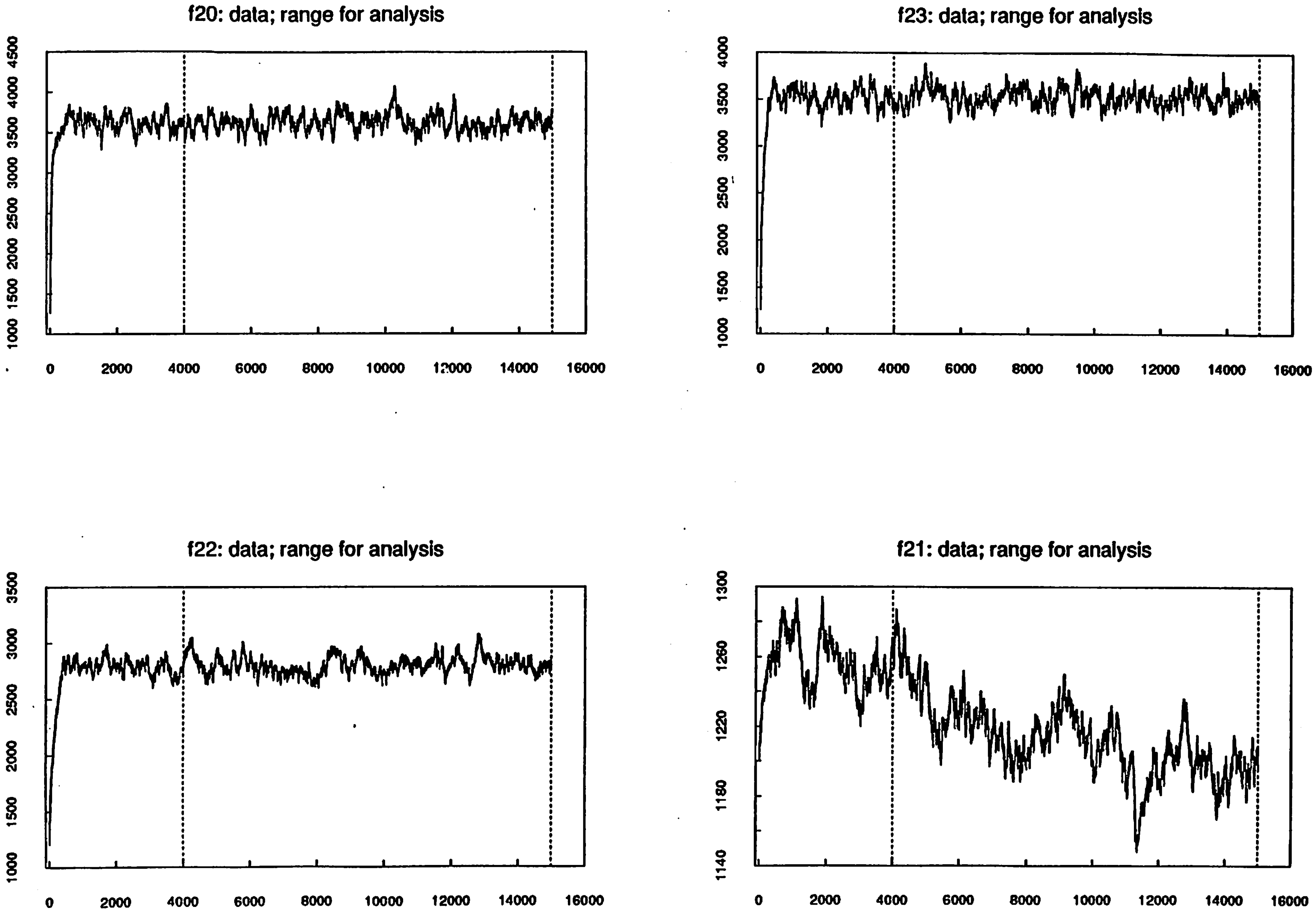
h20: data; range for analysis

h23: data; range for analysis

h22: data; range for analysis

h21: data; range for analysis

Fig 8


f20: data; range for analysis

f23: data; range for analysis

f22: data; range for analysis

f21: data; range for analysis

fig 9

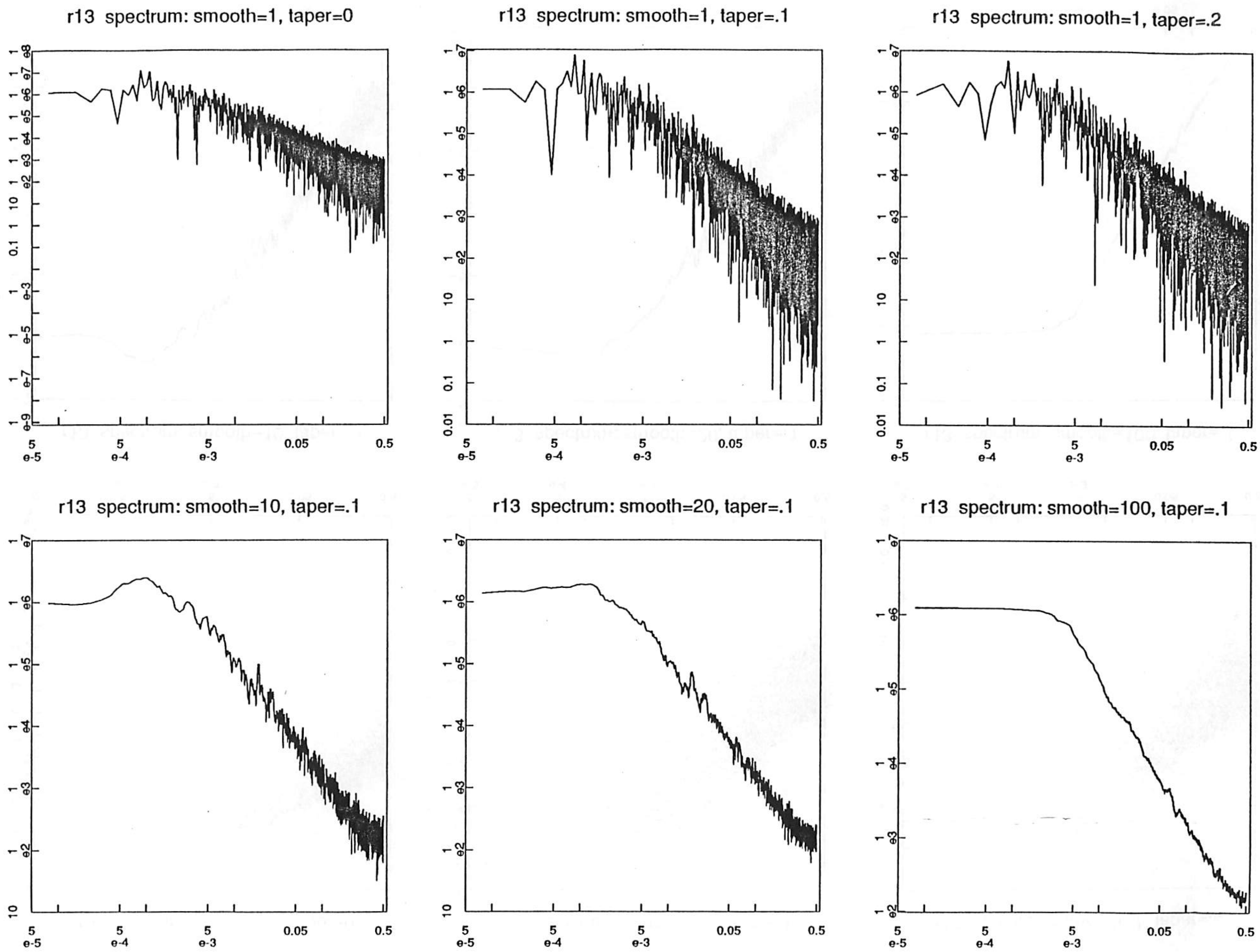
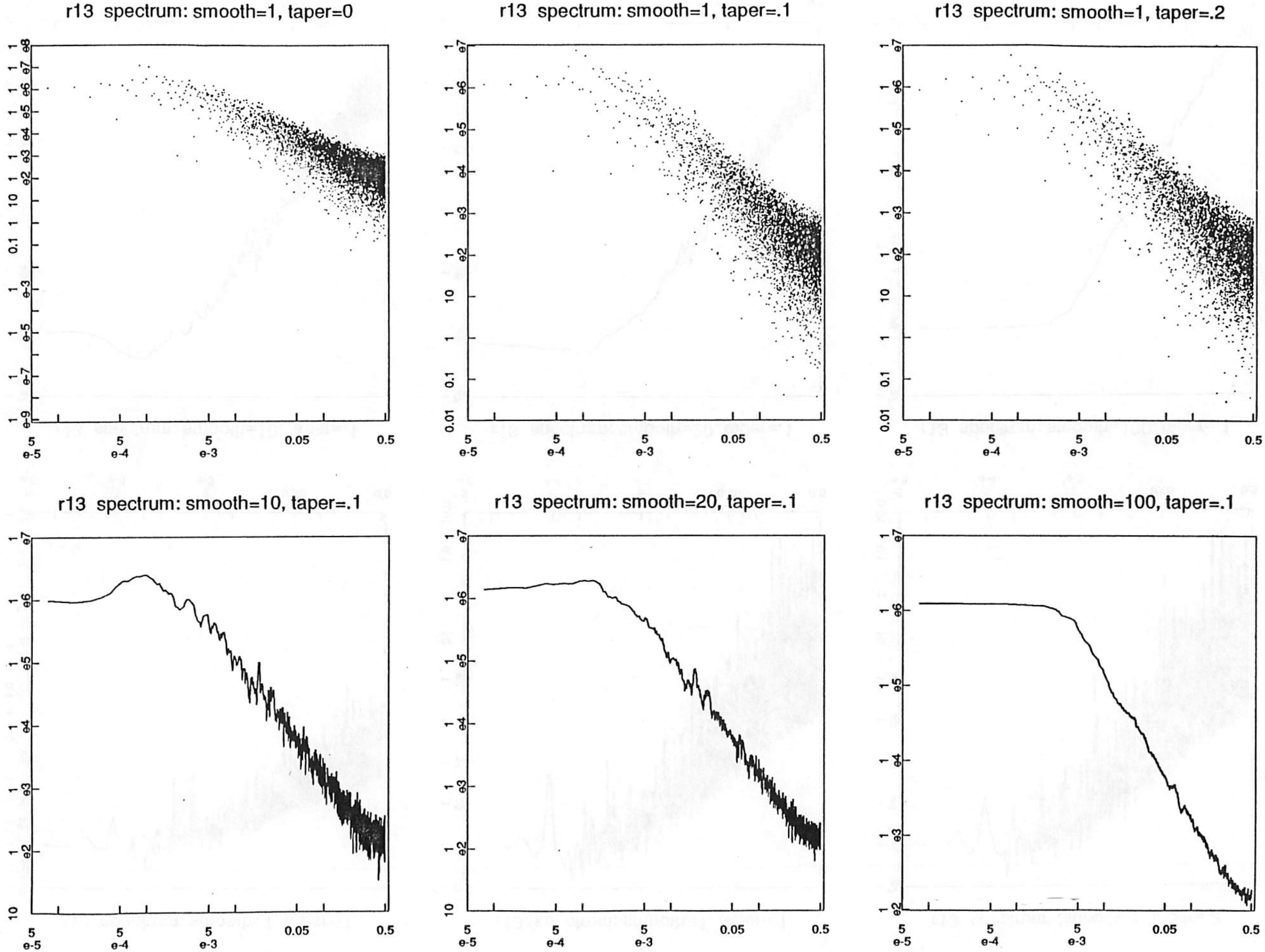

r13 spectrum: smooth=1, taper=0
r13 spectrum: smooth=1, taper=.1
r13 spectrum: smooth=1, taper=.2
r13 spectrum: smooth=10, taper=.1
r13 spectrum: smooth=20, taper=.1
r13 spectrum: smooth=100, taper=.1
1e-9
1e-7
1e-5
1e-3
0.1
1
10
1e2
1e3
1e4
1e5
1e6
1e7
1e8
0.01
5e-5
5e-4
5e-3
0.05
0.5

fig 10



r13 spectrum: smooth=1, taper=0

r13 spectrum: smooth=1, taper=.1

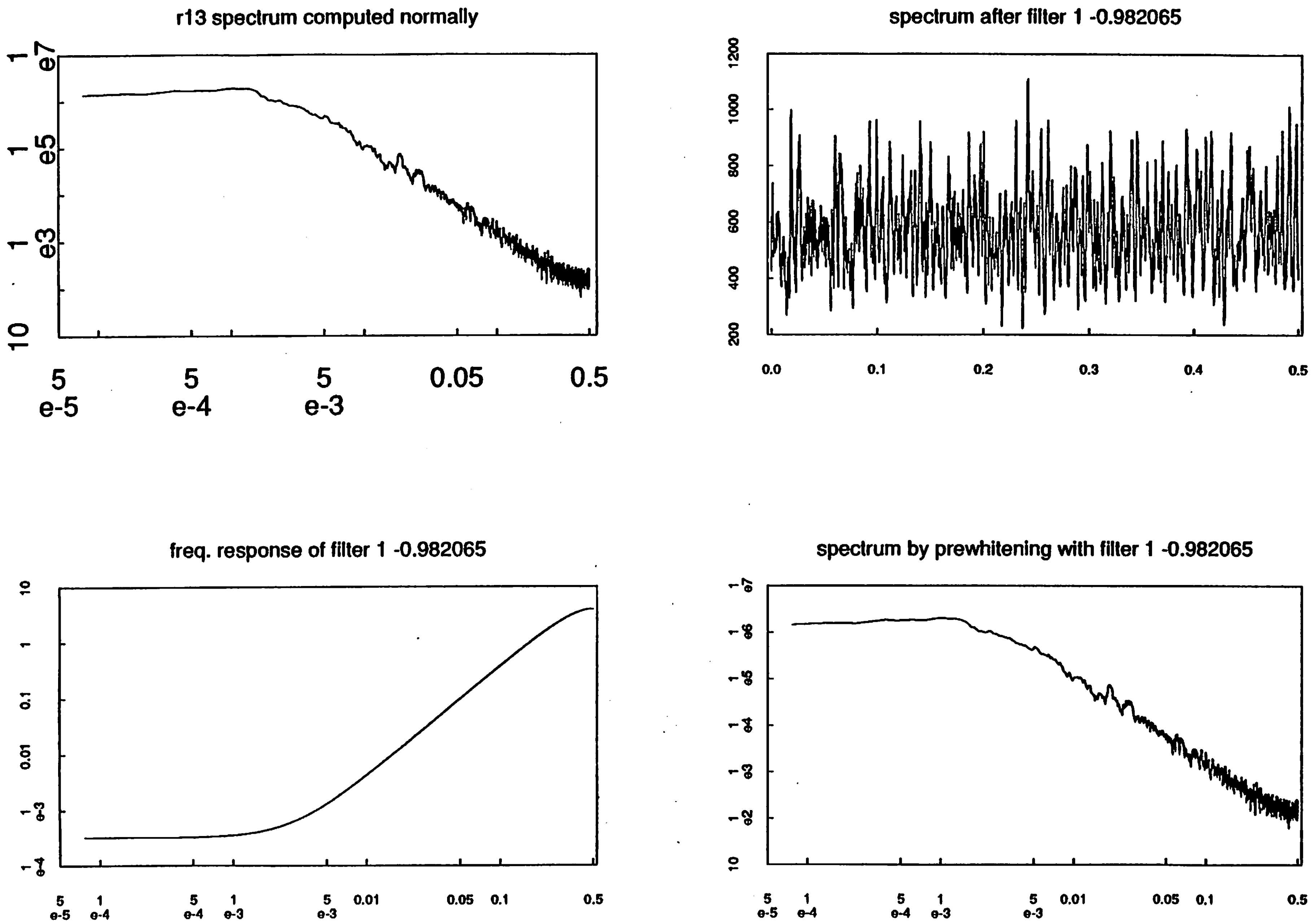
r13 spectrum: smooth=1, taper=.2

r13 spectrum: smooth=10, taper=.1

r13 spectrum: smooth=20, taper=.1

r13 spectrum: smooth=100, taper=.1

Fig 11



r13 spectrum computed normally

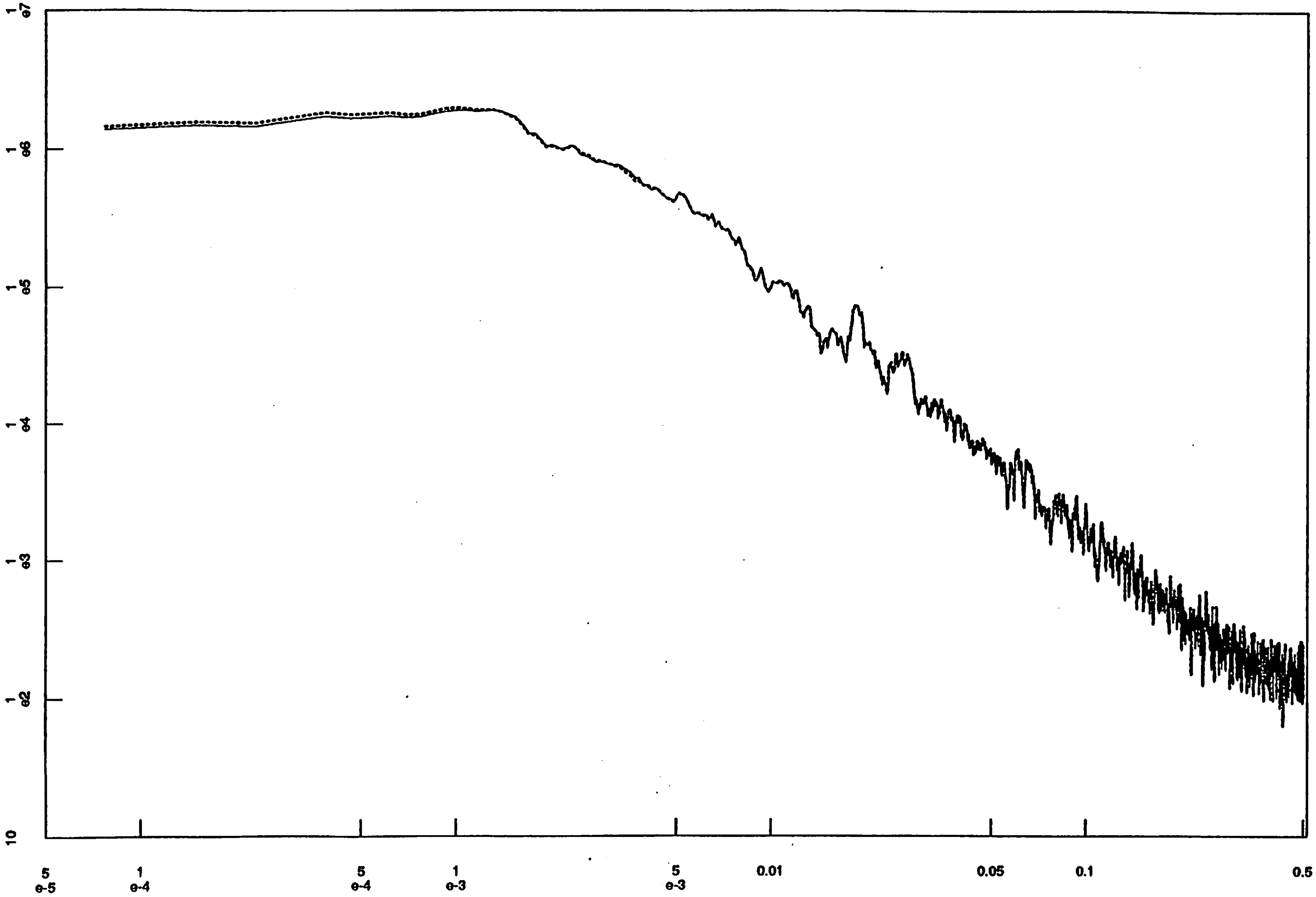
spectrum after filter 1 -0.982065

freq. response of filter 1 -0.982065

spectrum by prewhitening with filter 1 -0.982065

Indicate uncertainty bands

fig 12



r13 spectra: normally computed; prewhitened

fig 13

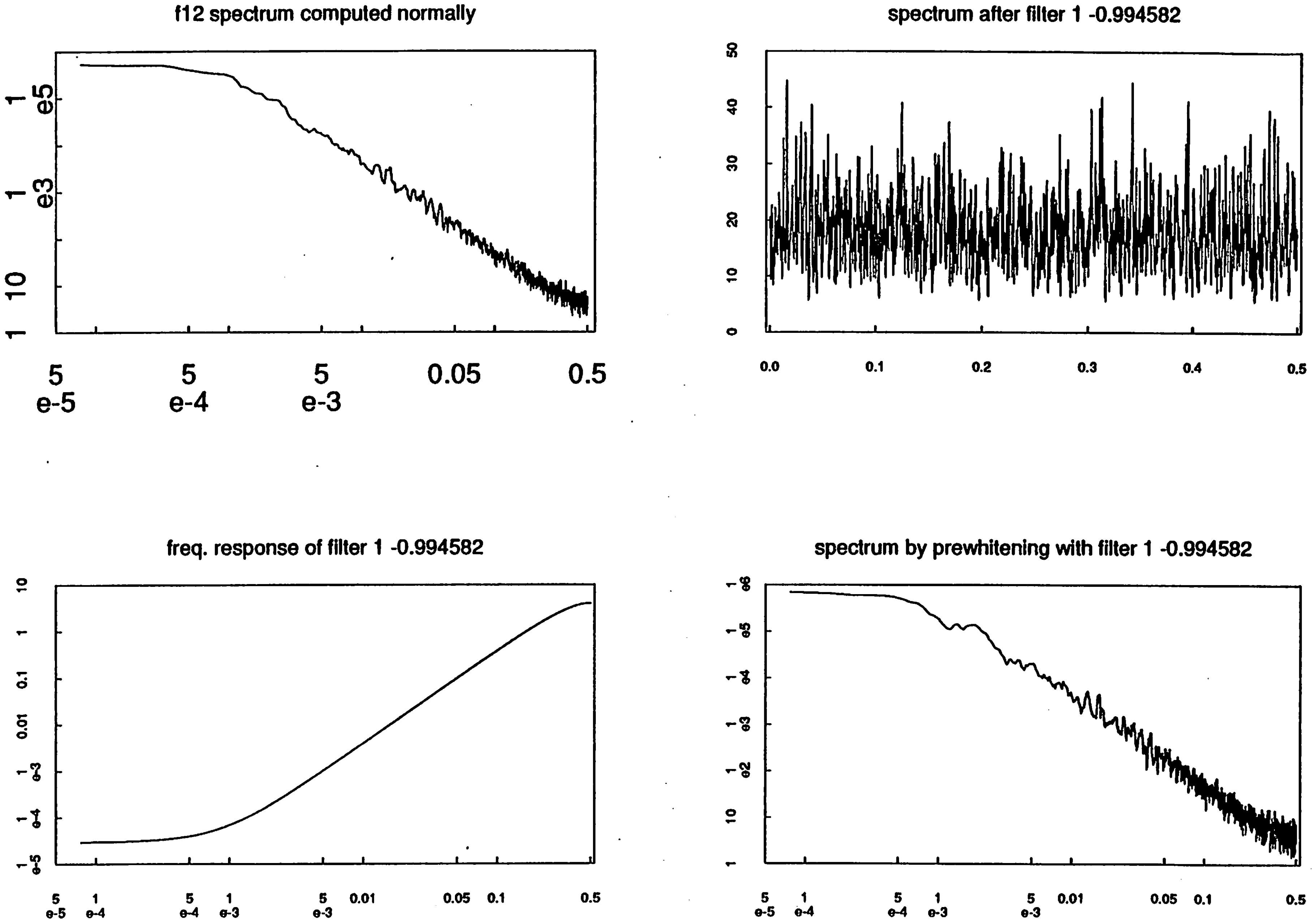
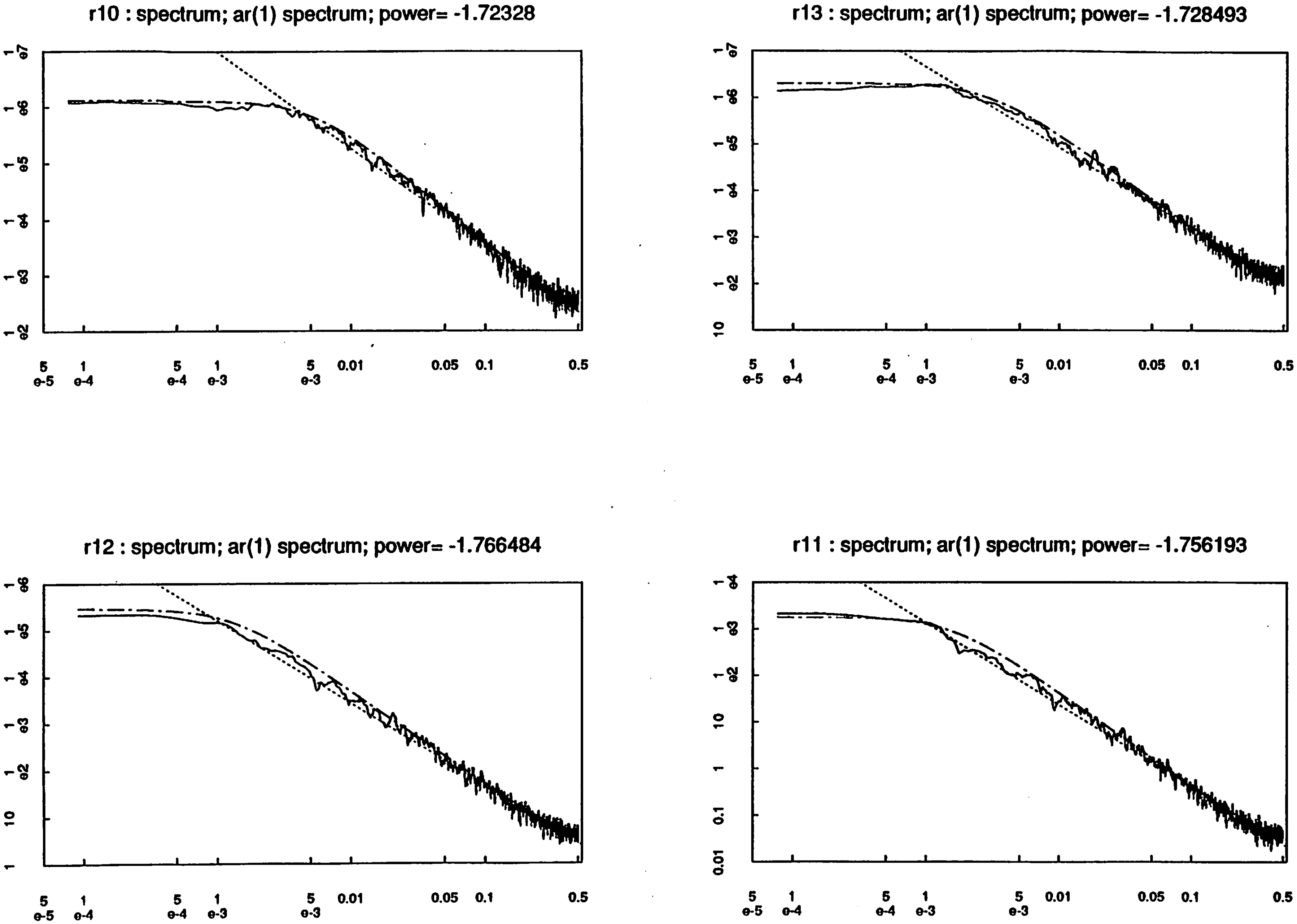

f12 spectrum computed normally
1 e5
1 e3
10
1
5 e-5
5 e-4
5 e-3
0.05
0.5
spectrum after filter 1 -0.994582
50
40
30
20
10
0
0.0
0.1
0.2
0.3
0.4
0.5
freq. response of filter 1 -0.994582
10
1
0.1
0.01
1 e-3
1 e-4
1 e-5
5 e-5
1 e-4
5 e-4
1 e-3
5 e-3
0.01
0.05
0.1
0.5
spectrum by prewhitening with filter 1 -0.994582
1 e6
1 e5
1 e4
1 e3
1 e2
10
1
5 e-5
1 e-4
5 e-4
1 e-3
5 e-3
0.01
0.05
0.1
0.5

fig 14



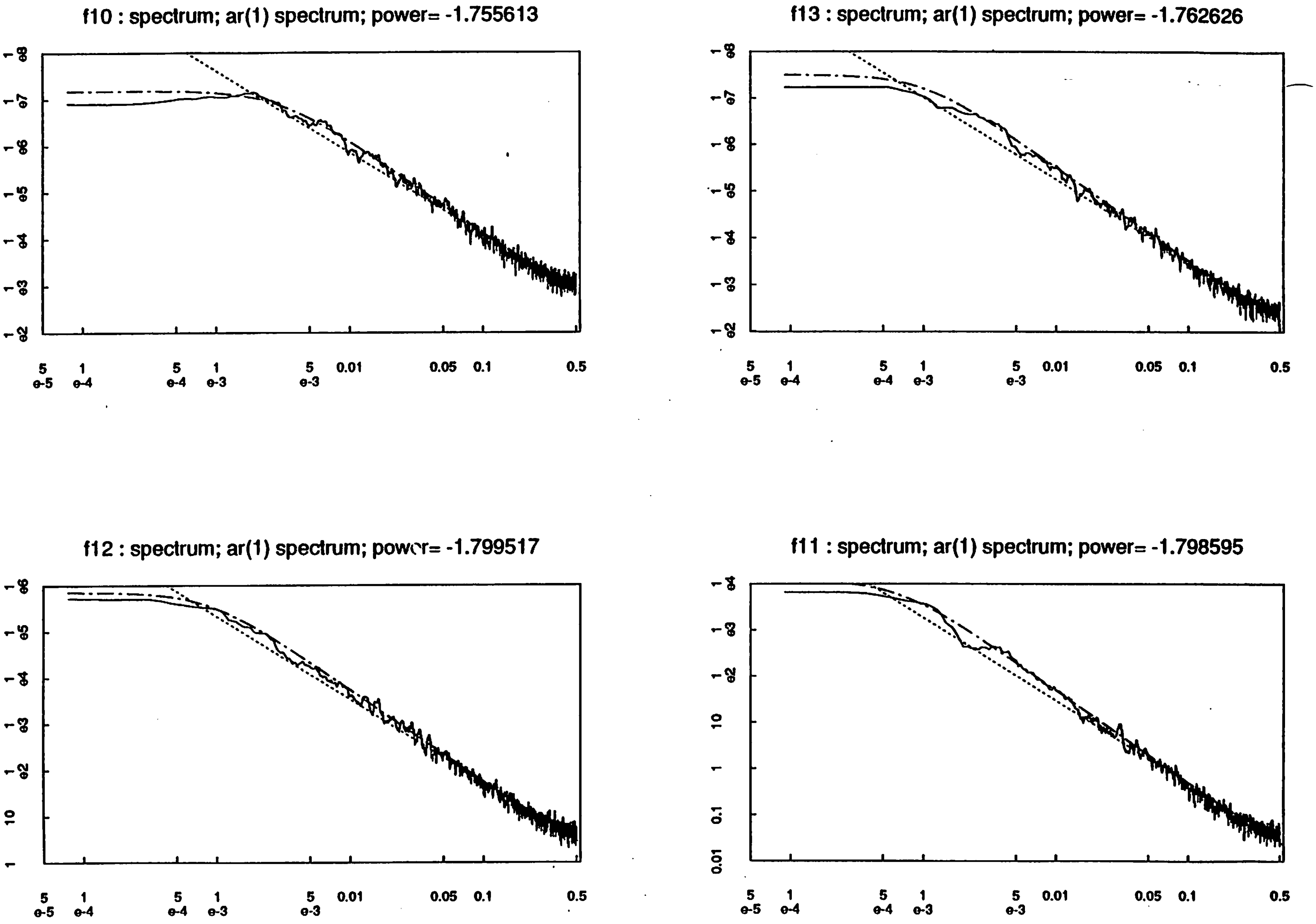
r10 : spectrum; ar(1) spectrum; power= -1.72328

r13 : spectrum; ar(1) spectrum; power= -1.728493

r12 : spectrum; ar(1) spectrum; power= -1.766484

r11 : spectrum; ar(1) spectrum; power= -1.756193

Fig 15



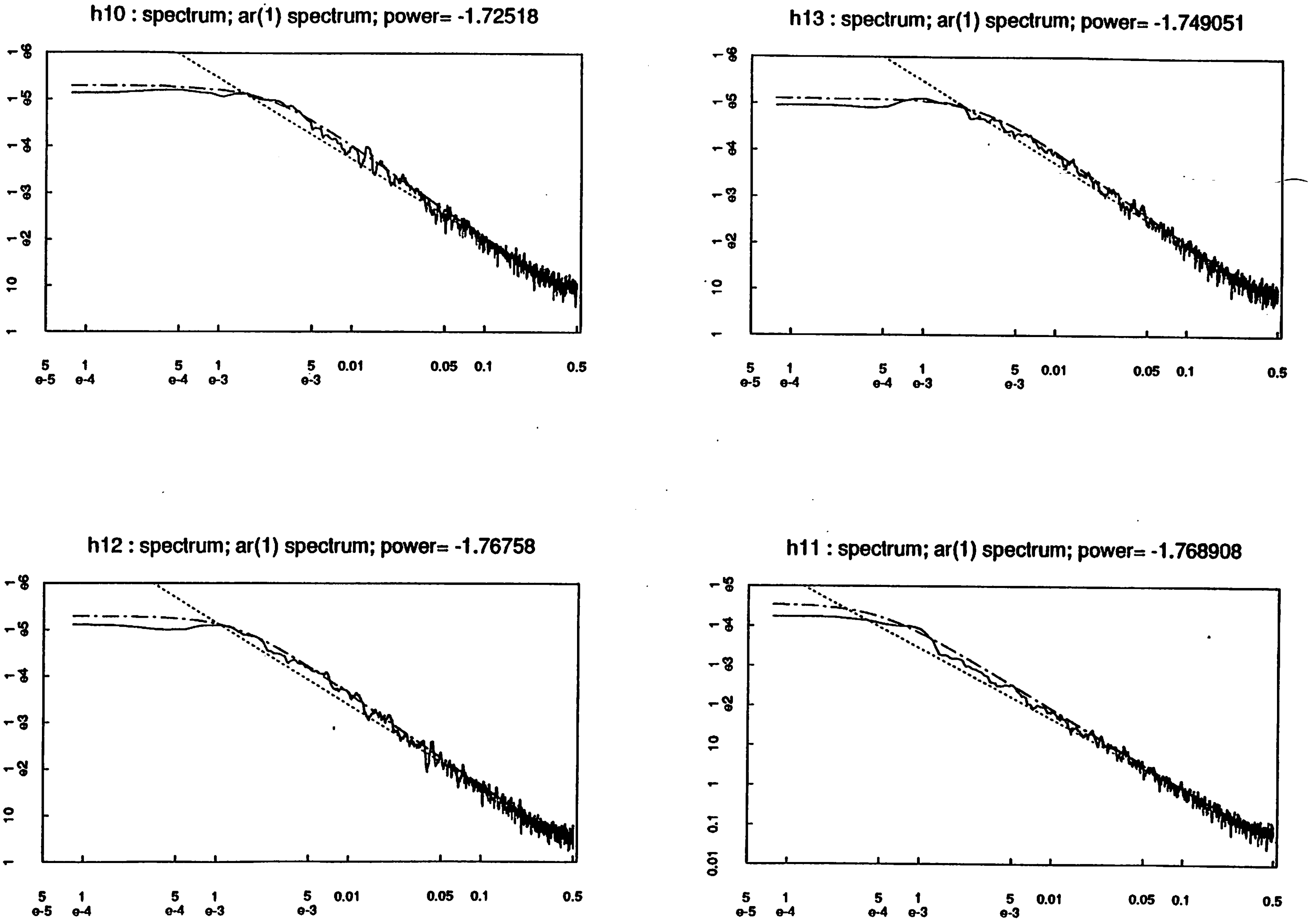
f10 : spectrum; ar(1) spectrum; power= -1.755613

f13 : spectrum; ar(1) spectrum; power= -1.762626

f12 : spectrum; ar(1) spectrum; power= -1.799517

f11 : spectrum; ar(1) spectrum; power= -1.798595

Fig 16



h10 : spectrum; ar(1) spectrum; power= -1.72518

h13 : spectrum; ar(1) spectrum; power= -1.749051

h12 : spectrum; ar(1) spectrum; power= -1.76758

h11 : spectrum; ar(1) spectrum; power= -1.768908 fig17

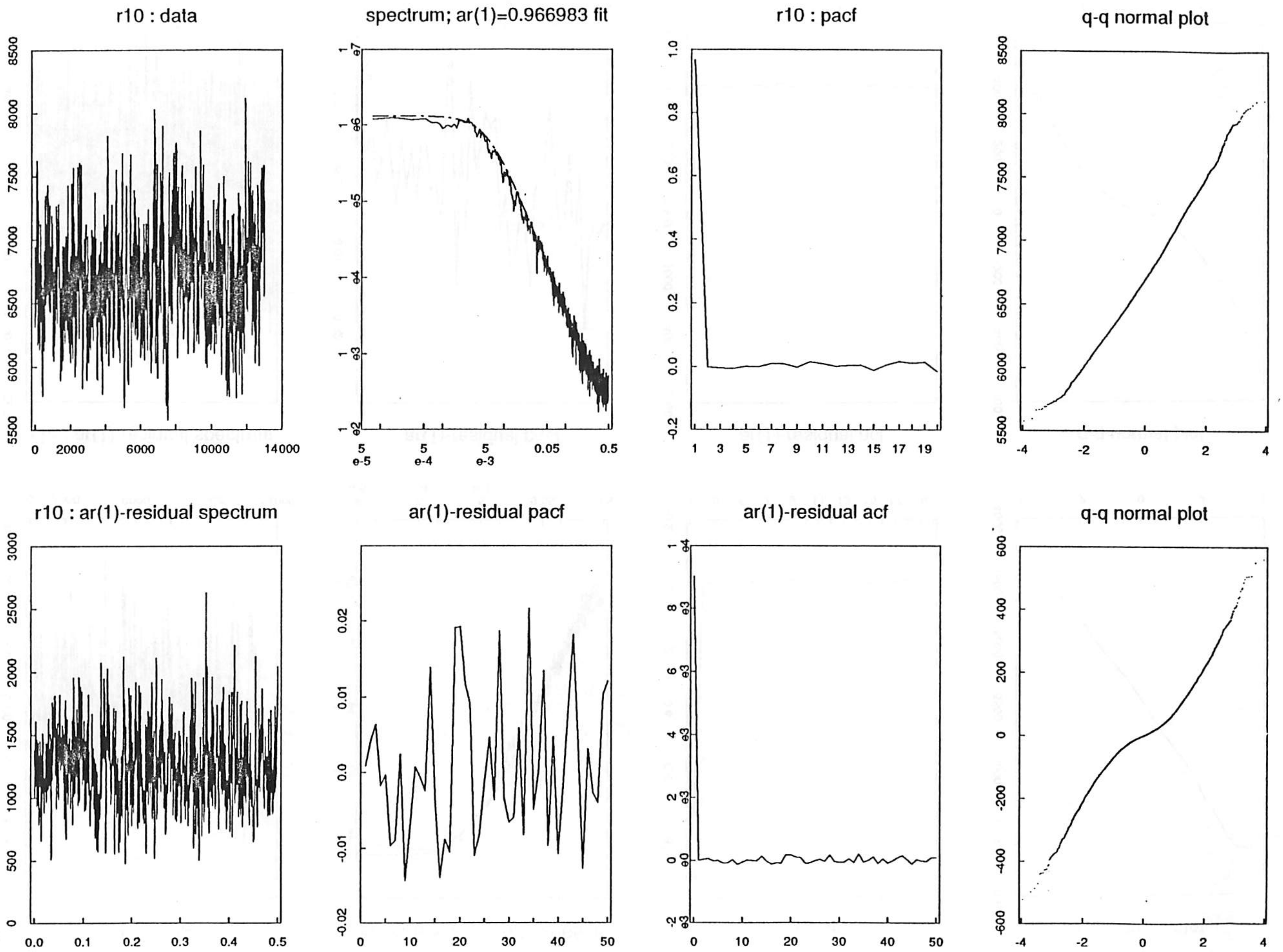

r10 : data
8500
8000
7500
7000
6500
6000
5500
0
2000
6000
10000
14000
spectrum; ar(1)=0.966983 fit
1 e7
1 e6
1 e5
1 e4
1 e3
1 e2
5 e-5
5 e-4
5 e-3
0.05
0.5
r10 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
8500
8000
7500
7000
6500
6000
5500
-4
-2
0
2
4
r10 : ar(1)-residual spectrum
3000
2500
2000
1500
1000
500
0
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.02
0.01
0.0
-0.01
-0.02
0
10
20
30
40
50
ar(1)-residual acf
1 e4
8 e3
6 e3
4 e3
2 e3
0 e0
-2 e3
0
10
20
30
40
50
q-q normal plot
600
400
200
0
-200
-400
-600
-4
-2
0
2
4

Fig 18

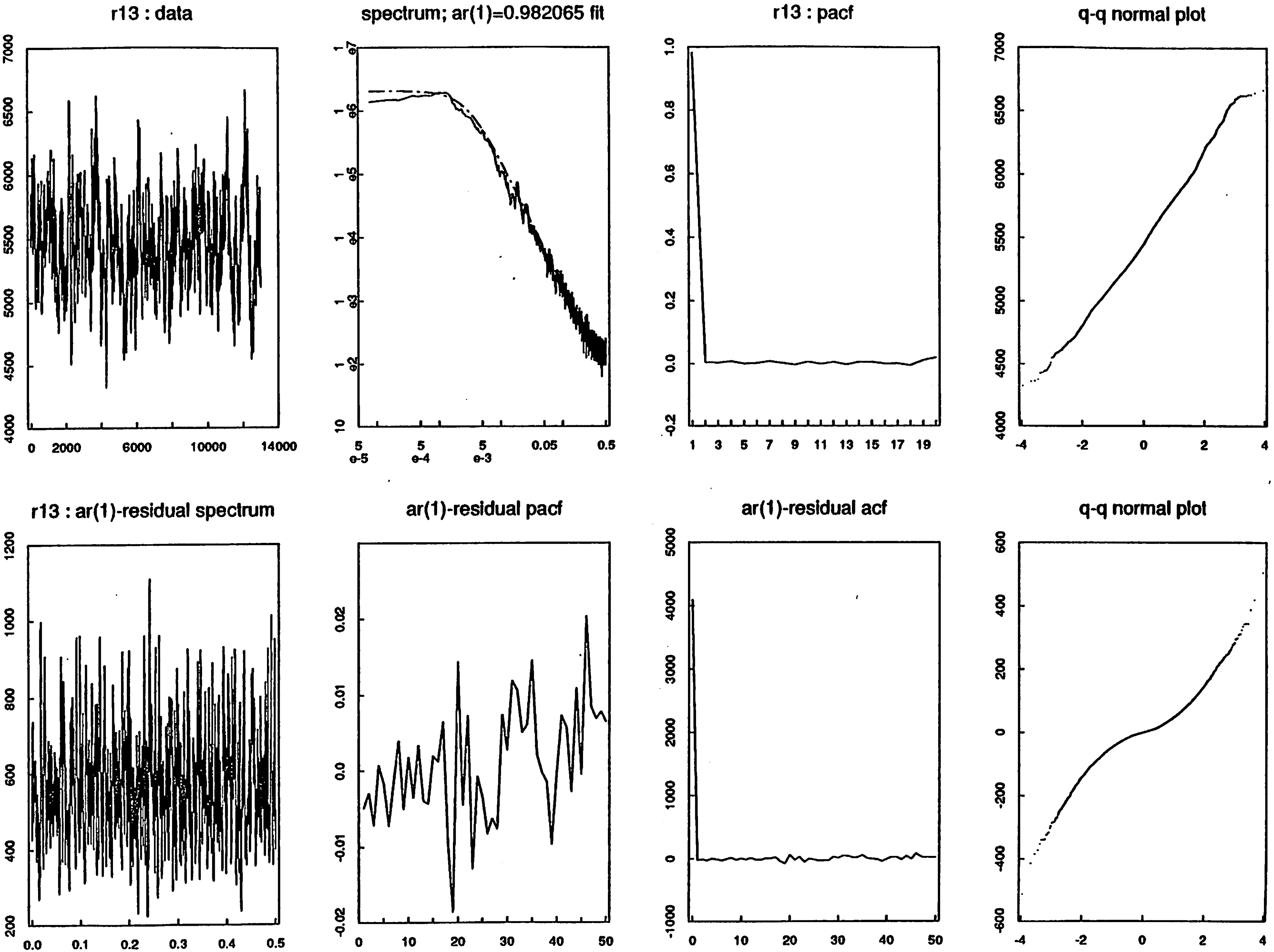

r13 : data
7000
6500
6000
5500
5000
4500
4000
0
2000
6000
10000
14000
spectrum; ar(1)=0.982065 fit
1 e7
1 e6
1 e5
1 e4
1 e3
1 e2
10
5 e-5
5 e-4
5 e-3
0.05
0.5
r13 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
7000
6500
6000
5500
5000
4500
4000
-4
-2
0
2
4
r13 : ar(1)-residual spectrum
1200
1000
800
600
400
200
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.02
0.01
0.0
-0.01
-0.02
0
10
20
30
40
50
ar(1)-residual acf
5000
4000
3000
2000
1000
0
-1000
0
10
20
30
40
50
q-q normal plot
600
400
200
0
-200
-400
-600
-4
-2
0
2
4

fig 19

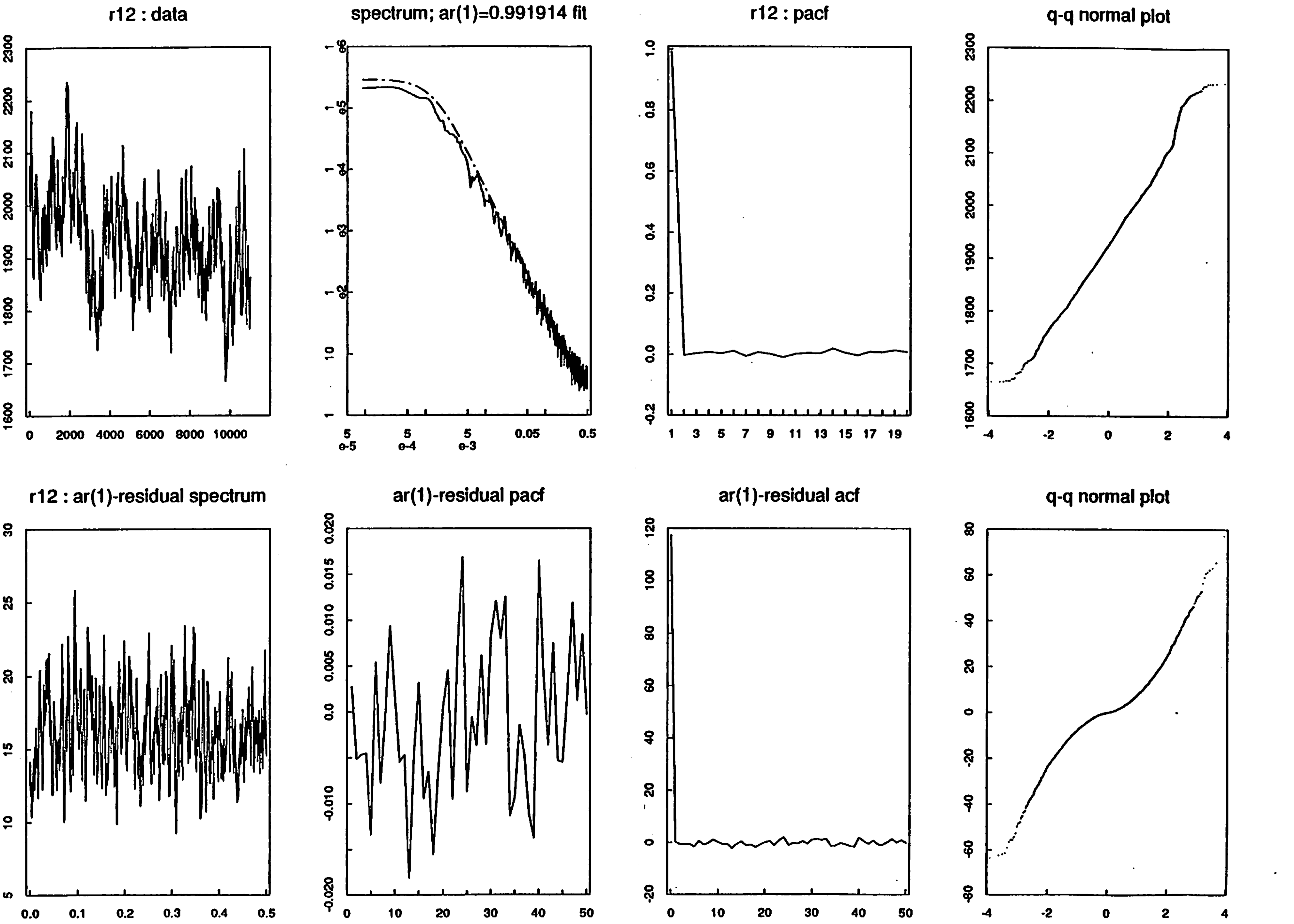

r12 : data
1600
1700
1800
1900
2000
2100
2200
2300
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.991914 fit
1 e6
1 e5
1 e4
1 e3
1 e2
10
1
5 e-5
5 e-4
5 e-3
0.05
0.5
r12 : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
7
9
11
13
15
17
19
q-q normal plot
1600
1700
1800
1900
2000
2100
2200
2300
-4
-2
0
2
4
r12 : ar(1)-residual spectrum
5
10
15
20
25
30
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.020
-0.010
0.0
0.005
0.010
0.015
0.020
0
10
20
30
40
50
ar(1)-residual acf
-20
0
20
40
60
80
100
120
0
10
20
30
40
50
q-q normal plot
-80
-60
-40
-20
0
20
40
60
80
-4
-2
0
2
4

fig 20

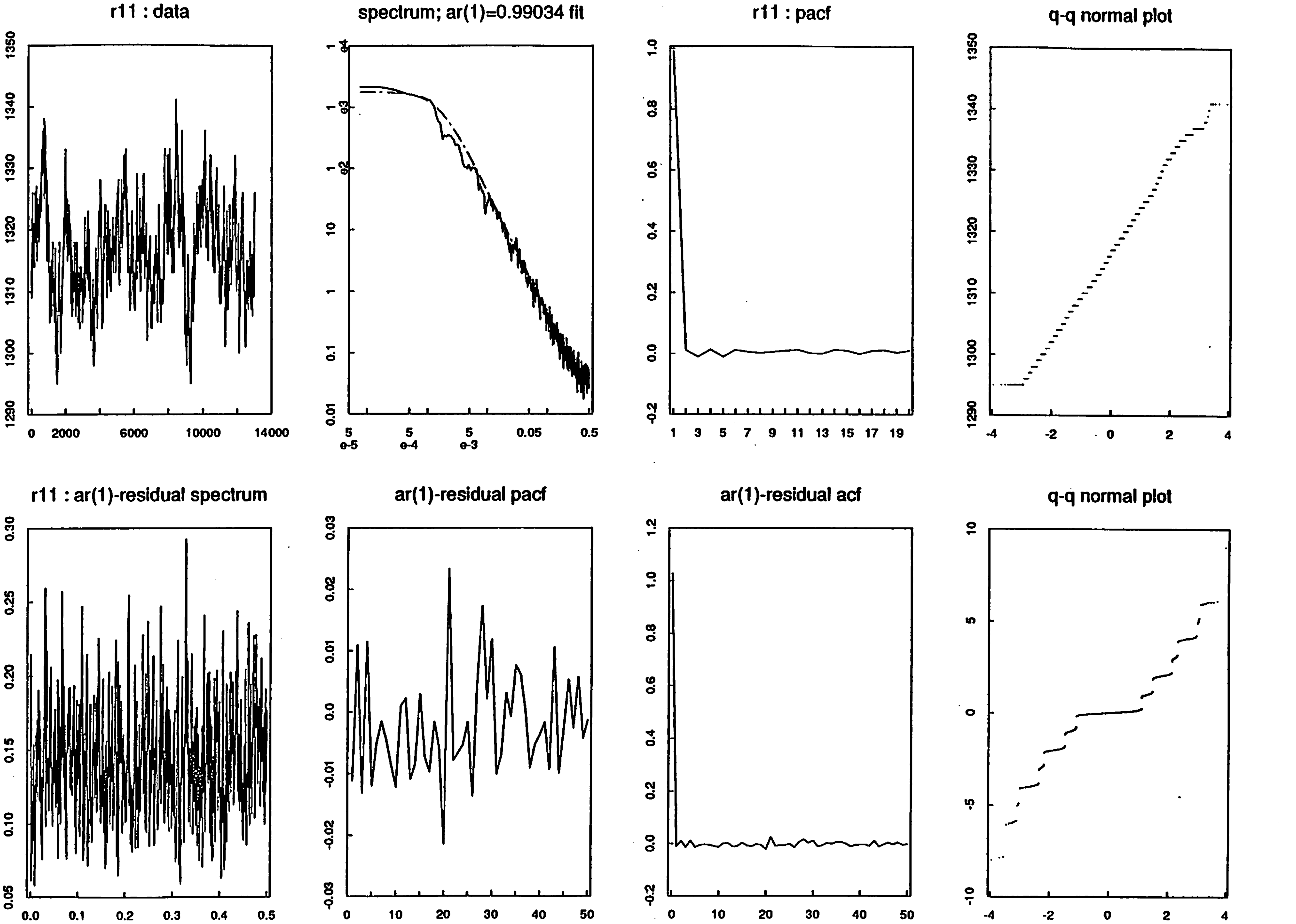

r11 : data
1290
1300
1310
1320
1330
1340
1350
0
2000
6000
10000
14000
spectrum; ar(1)=0.99034 fit
1 e4
1 e3
1 e2
10
1
0.1
0.01
5 e-5
5 e-4
5 e-3
0.05
0.5
r11 : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
7
9
11
13
15
17
19
q-q normal plot
1290
1300
1310
1320
1330
1340
1350
-4
-2
0
2
4
r11 : ar(1)-residual spectrum
0.05
0.10
0.15
0.20
0.25
0.30
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.03
-0.02
-0.01
0.0
0.01
0.02
0.03
0
10
20
30
40
50
ar(1)-residual acf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
10
20
30
40
50
q-q normal plot
-10
-5
0
5
10
-4
-2
0
2
4

fig21

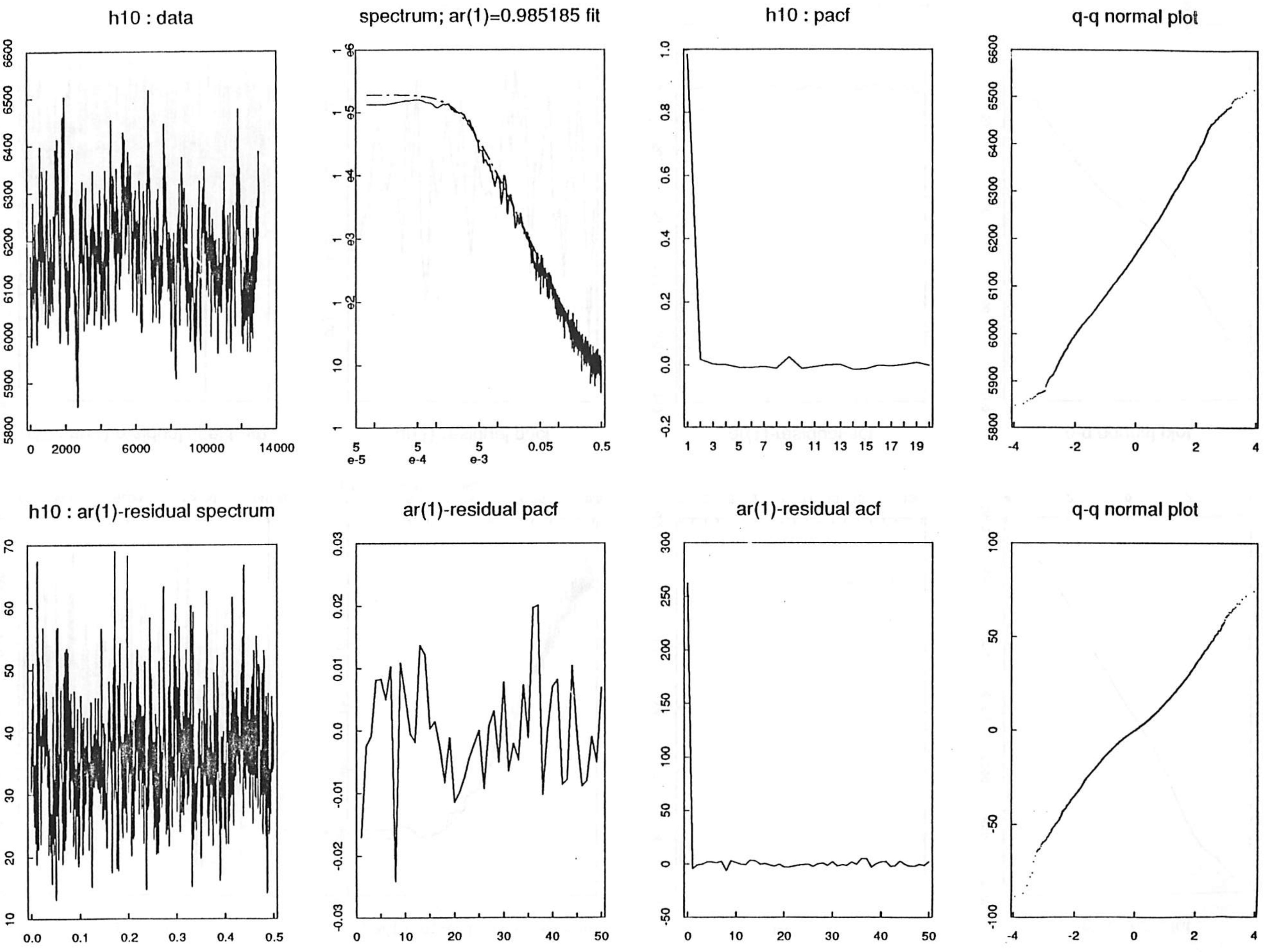
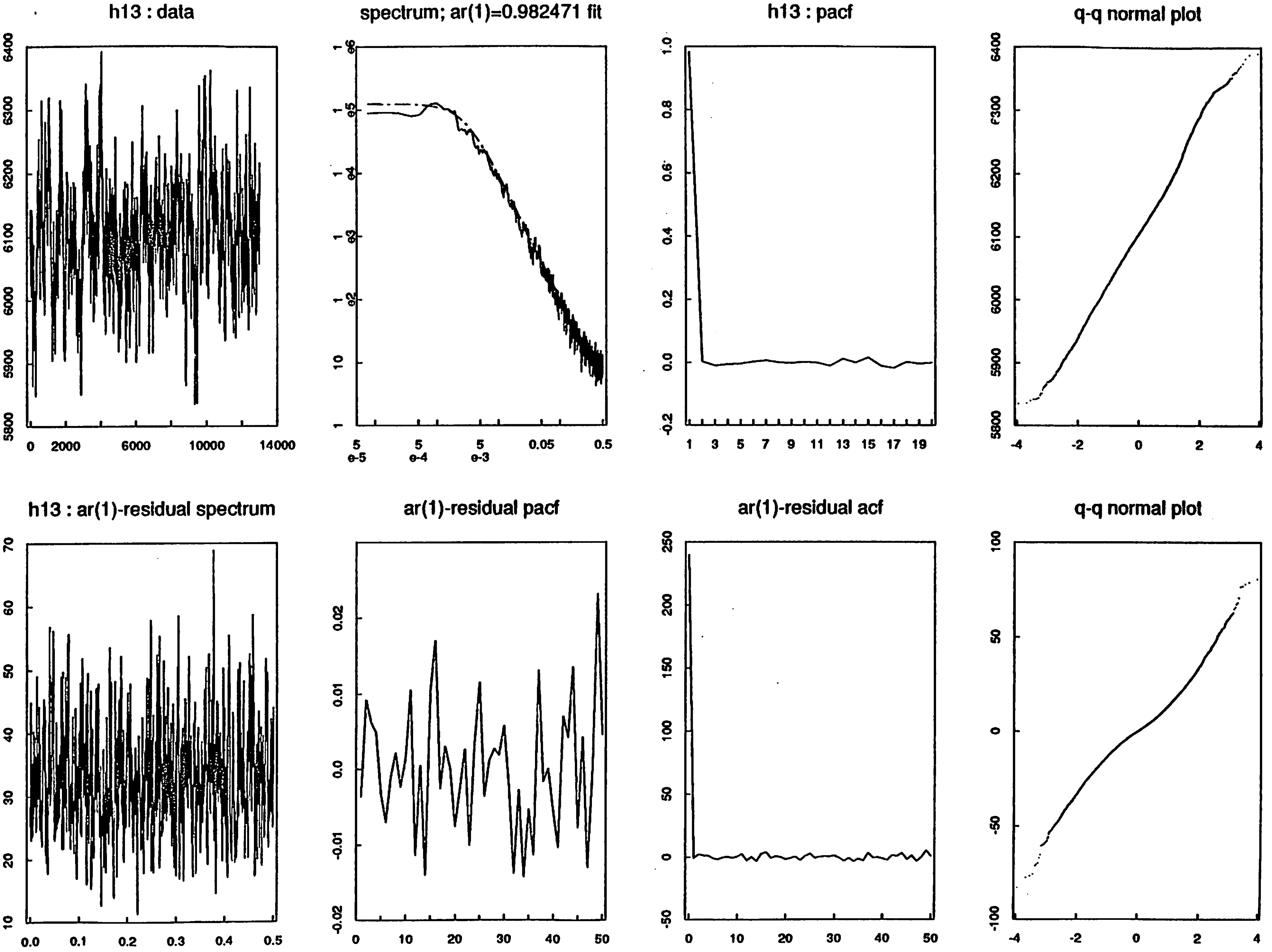

h10 : data
6600
6500
6400
6300
6200
6100
6000
5900
5800
0
2000
6000
10000
14000
spectrum; ar(1)=0.985185 fit
1 e6
1 e5
1 e4
1 e3
1 e2
10
1
5 e-5
5 e-4
5 e-3
0.05
0.5
h10 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
6600
6500
6400
6300
6200
6100
6000
5900
5800
-4
-2
0
2
4
h10 : ar(1)-residual spectrum
70
60
50
40
30
20
10
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.03
0.02
0.01
0.0
-0.01
-0.02
-0.03
0
10
20
30
40
50
ar(1)-residual acf
300
250
200
150
100
50
0
-50
0
10
20
30
40
50
q-q normal plot
100
50
0
-50
-100
-4
-2
0
2
4

fig 22



h13 : data

spectrum; ar(1)=0.982471 fit

h13 : pacf

q-q normal plot

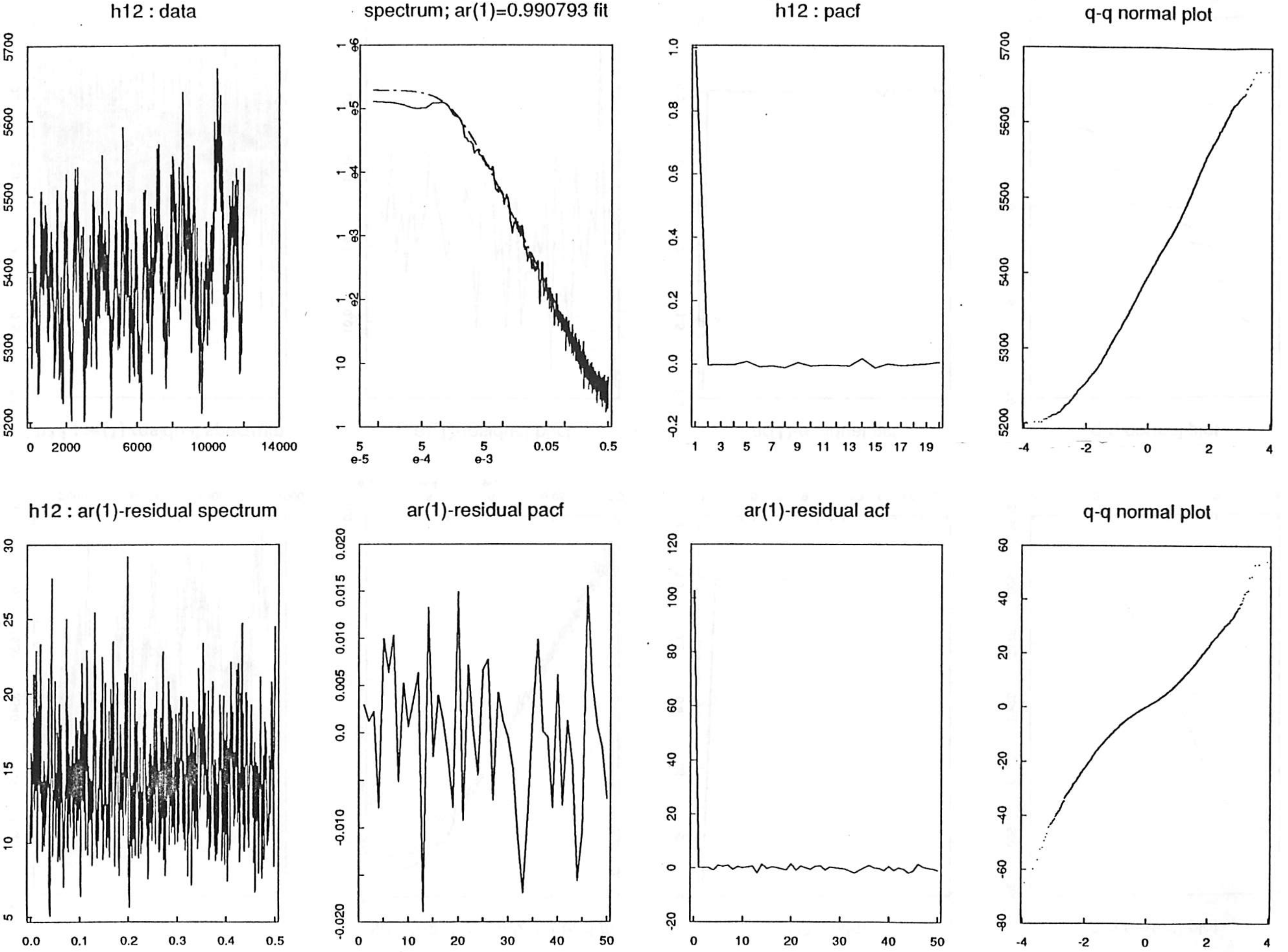
h13 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

fig 23



h12 : data

spectrum; ar(1)=0.990793 fit

h12 : pacf

q-q normal plot

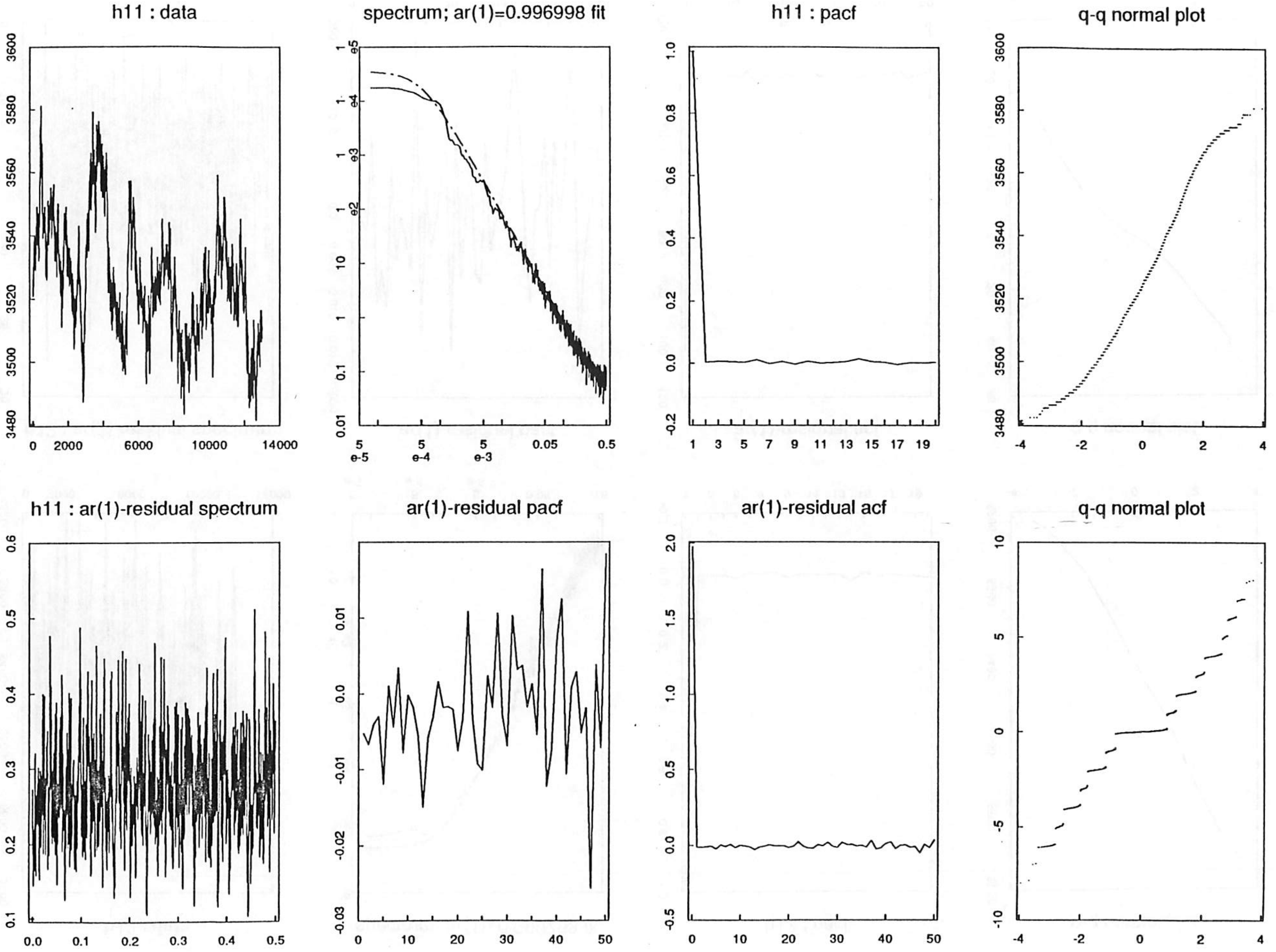
h12 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

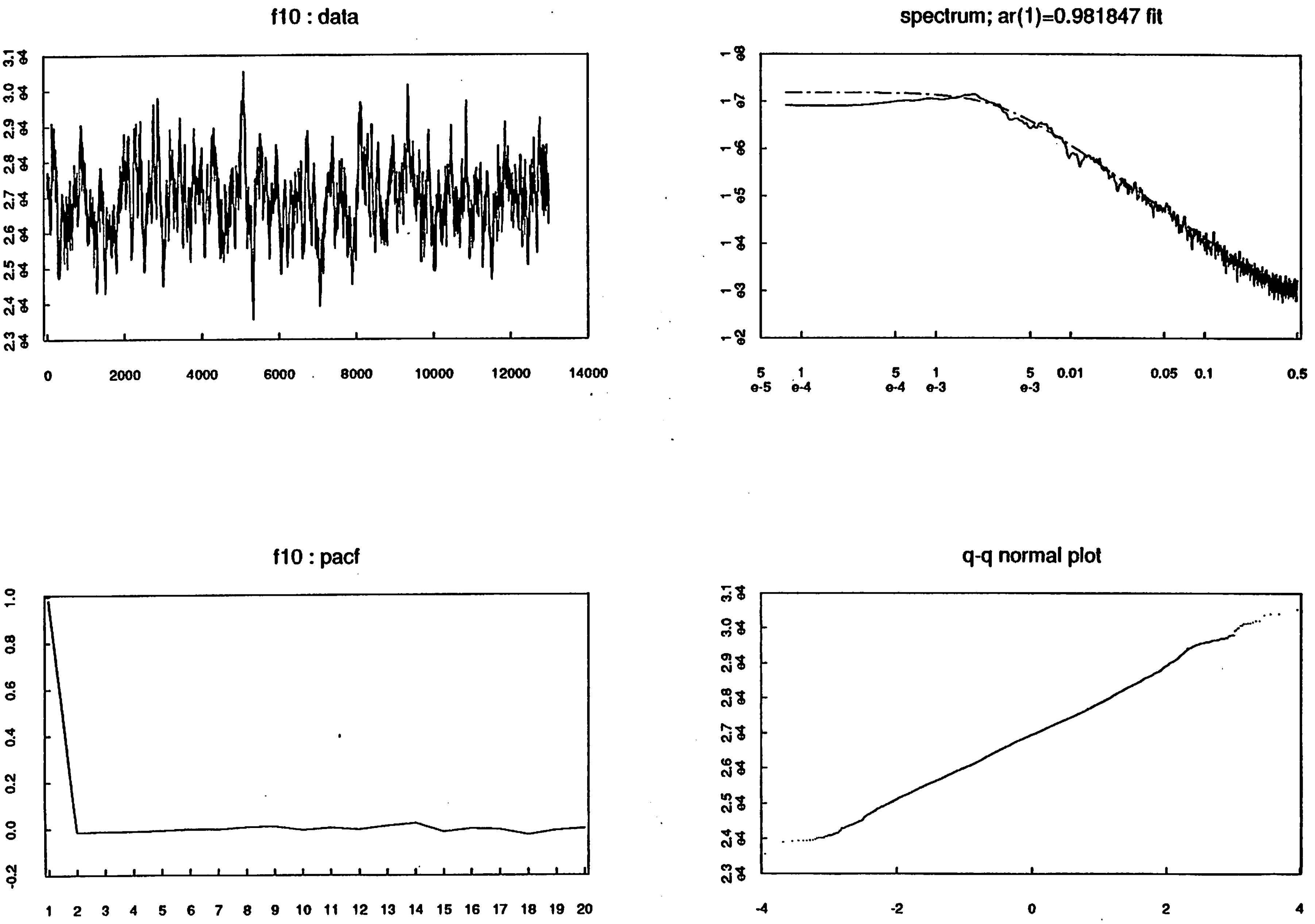
q-q normal plot

fig 24



h11 : data
3480
3500
3520
3540
3560
3580
3600
0
2000
6000
10000
14000
spectrum; ar(1)=0.996998 fit
0.01
0.1
1
10
1 e2
1 e3
1 e4
1 e5
5 e-5
5 e-4
5 e-3
0.05
0.5
h11 : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
7
9
11
13
15
17
19
q-q normal plot
3480
3500
3520
3540
3560
3580
3600
-4
-2
0
2
4
h11 : ar(1)-residual spectrum
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.03
-0.02
-0.01
0.0
0.01
0
10
20
30
40
50
ar(1)-residual acf
-0.5
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
50
q-q normal plot
-10
-5
0
5
10
-4
-2
0
2
4

fig 25a



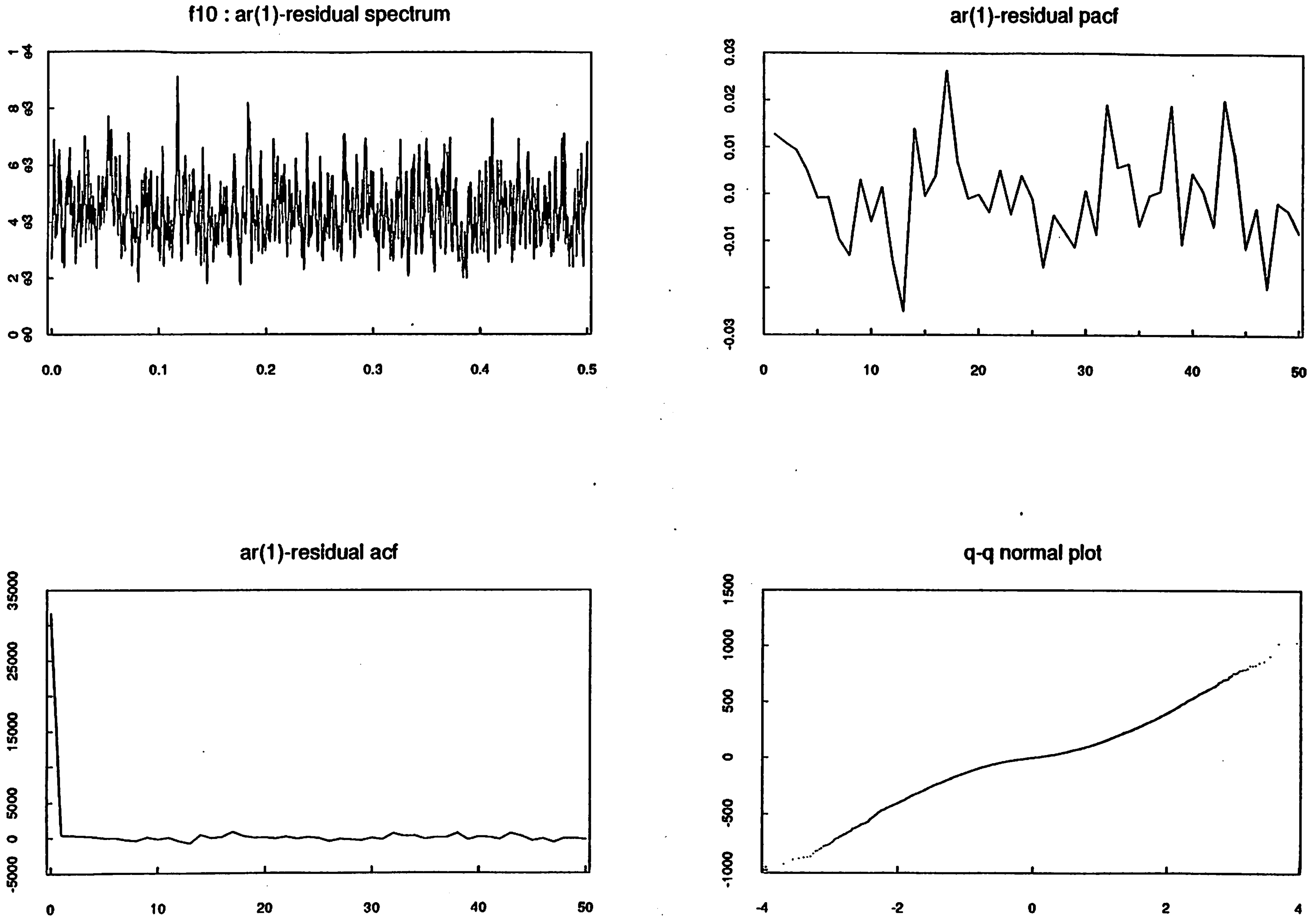
f10 : data

spectrum; ar(1)=0.981847 fit

f10 : pacf

q-q normal plot

fig 26b

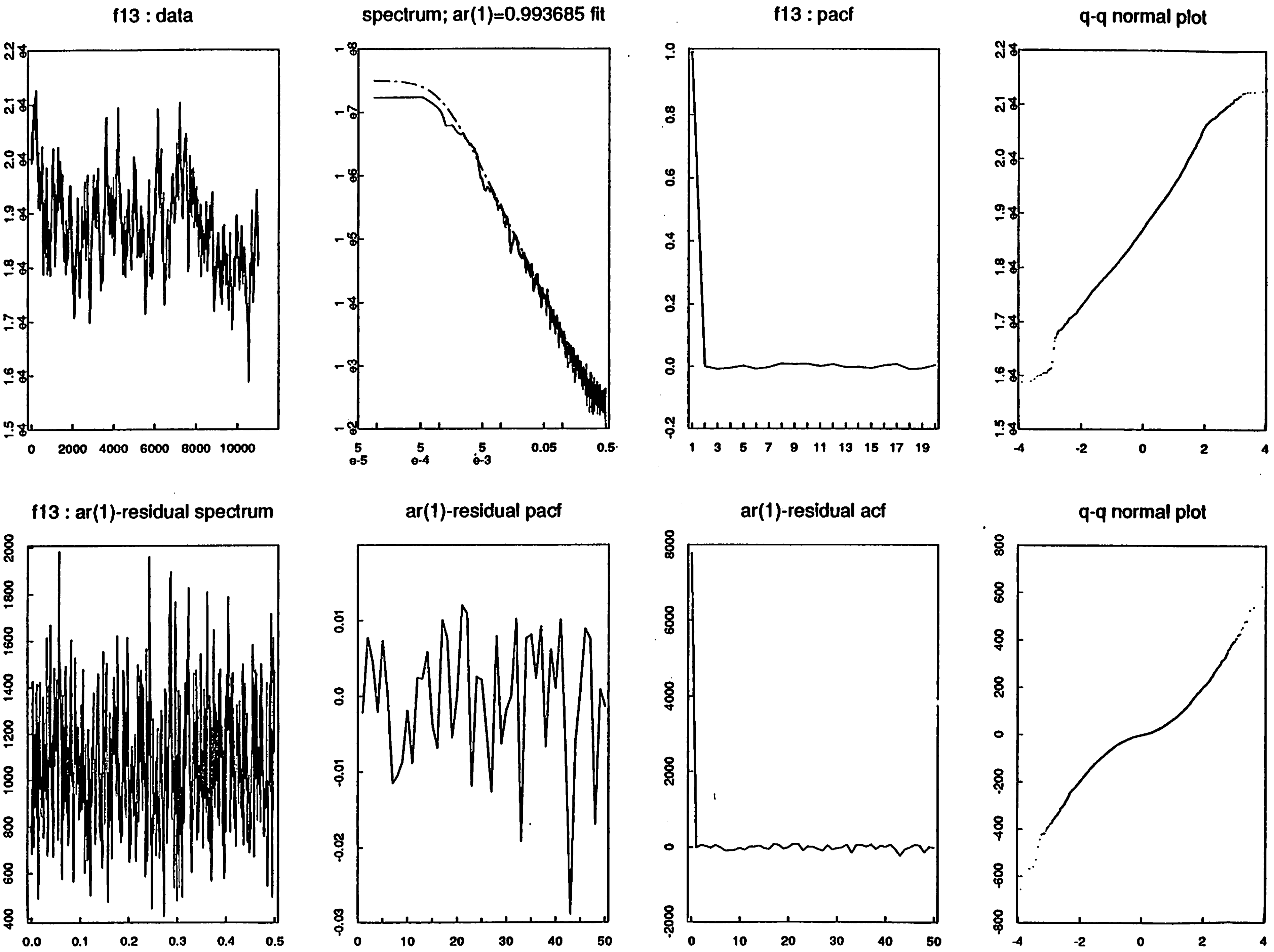

f10 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

fig 26



f13 : data

spectrum; ar(1)=0.993685 fit

f13 : pacf

q-q normal plot

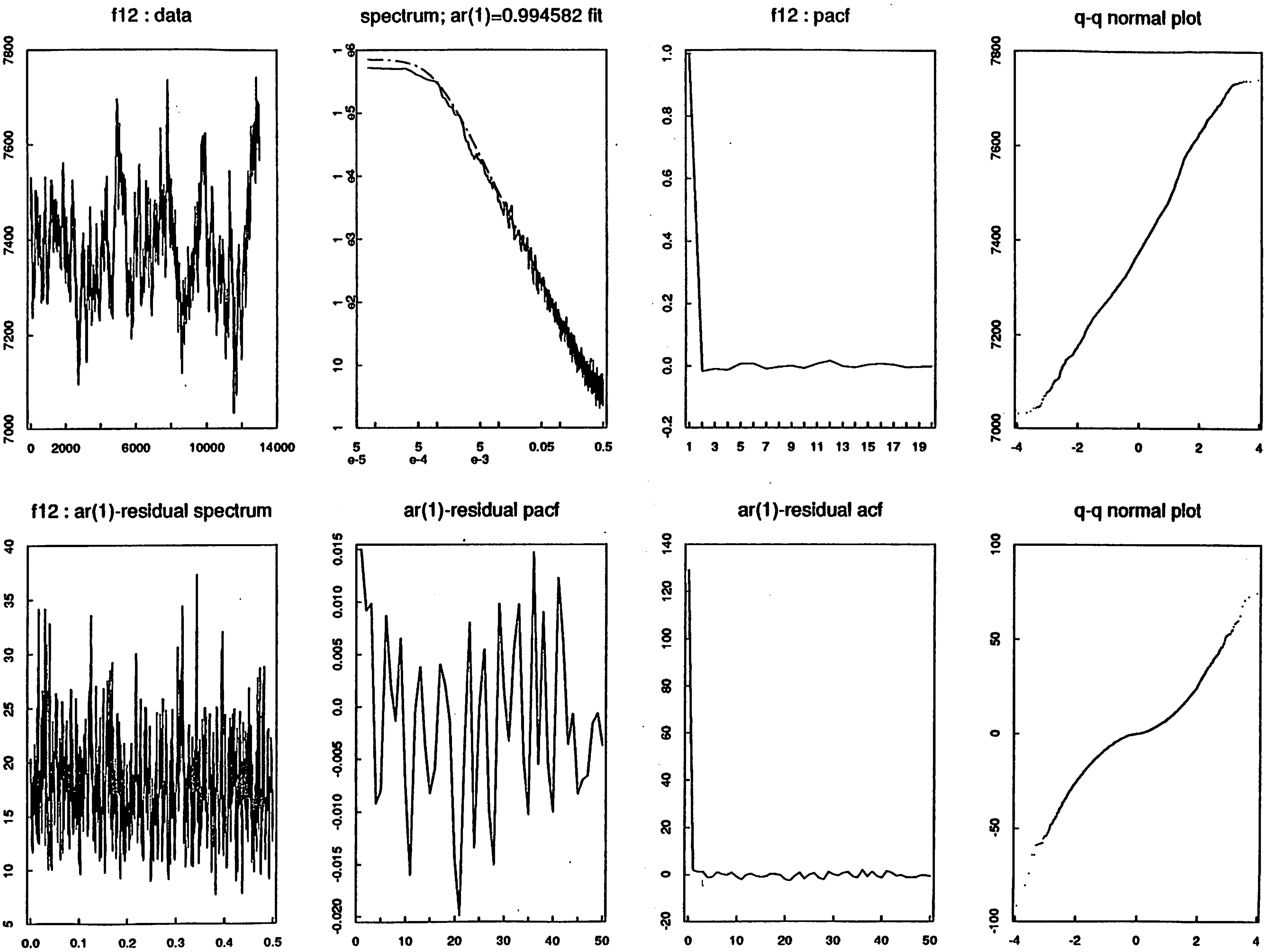
f13 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

fig 27



f12 : data

spectrum; ar(1)=0.994582 fit

f12 : pacf

q-q normal plot

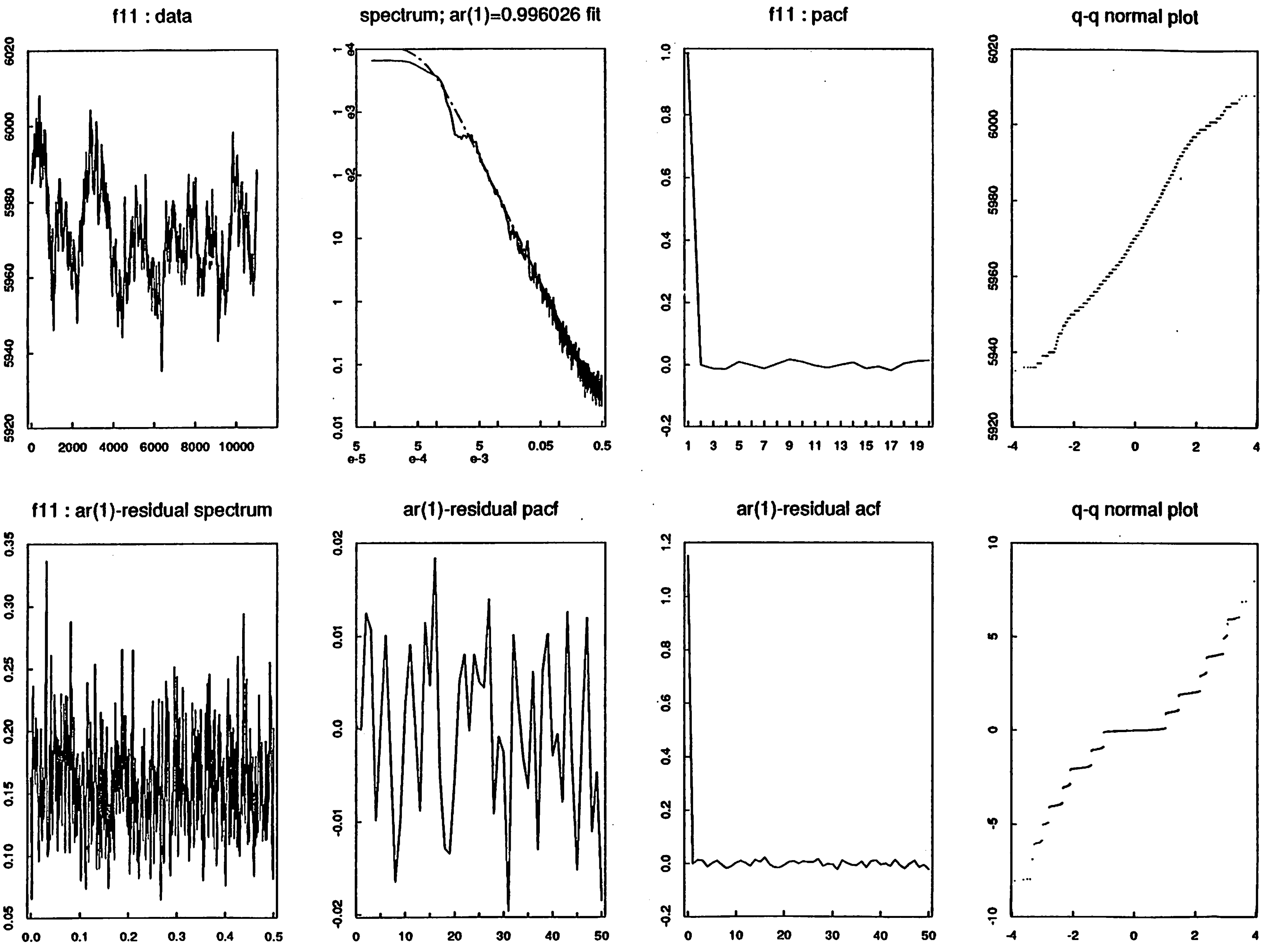
f12 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

fig 28



f11 : data

spectrum; ar(1)=0.996026 fit

f11 : pacf

q-q normal plot

f11 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

Fig 21

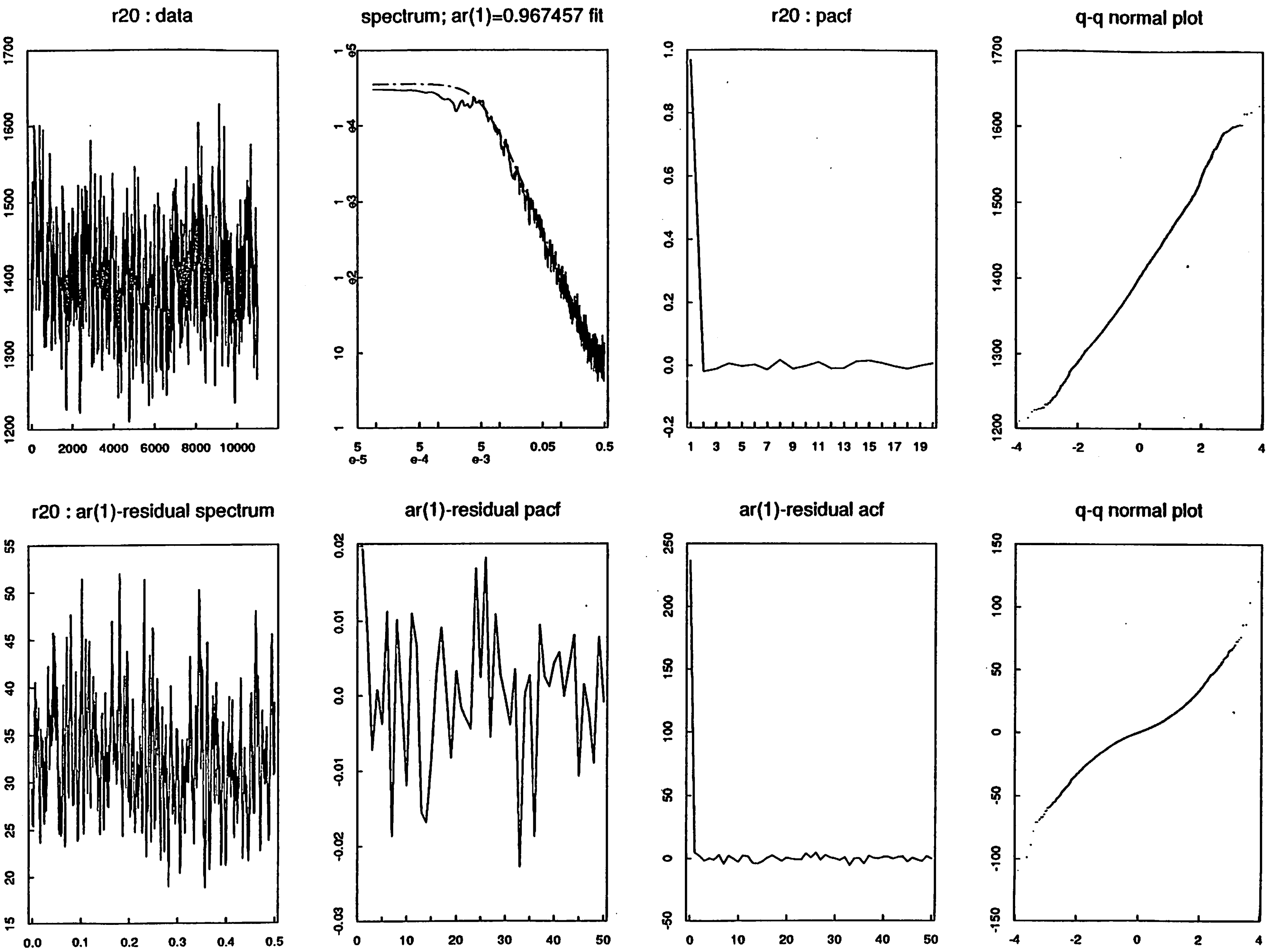

r20 : data
1200
1300
1400
1500
1600
1700
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.967457 fit
1
10
1 e2
1 e3
1 e4
1 e5
5 e-5
5 e-4
5 e-3
0.05
0.5
r20 : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
7
9
11
13
15
17
19
q-q normal plot
1200
1300
1400
1500
1600
1700
-4
-2
0
2
4
r20 : ar(1)-residual spectrum
15
20
25
30
35
40
45
50
55
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.03
-0.02
-0.01
0.0
0.01
0.02
0
10
20
30
40
50
ar(1)-residual acf
-50
0
50
100
150
200
250
0
10
20
30
40
50
q-q normal plot
-150
-100
-50
0
50
100
150
-4
-2
0
2
4

Fig 30

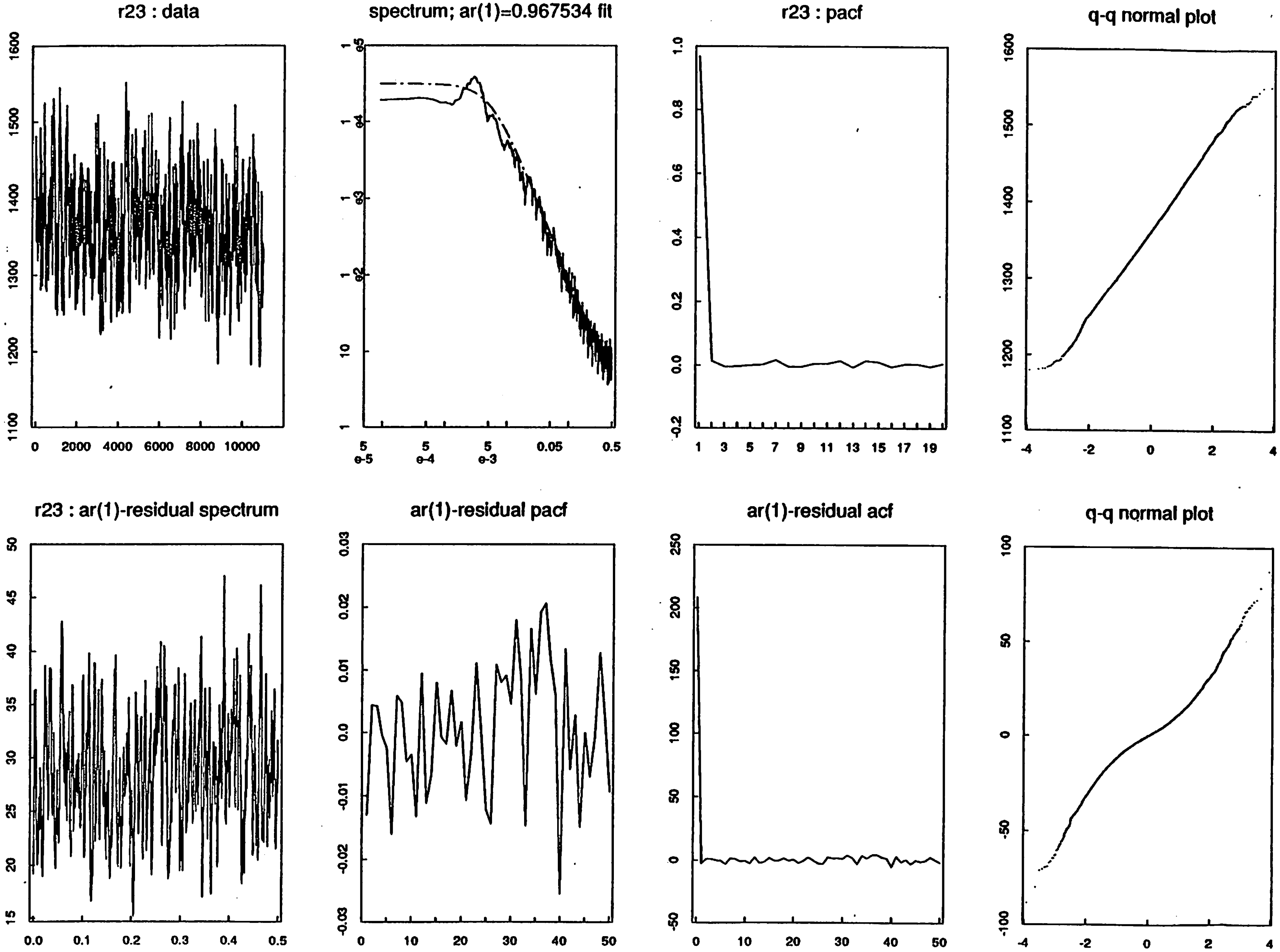
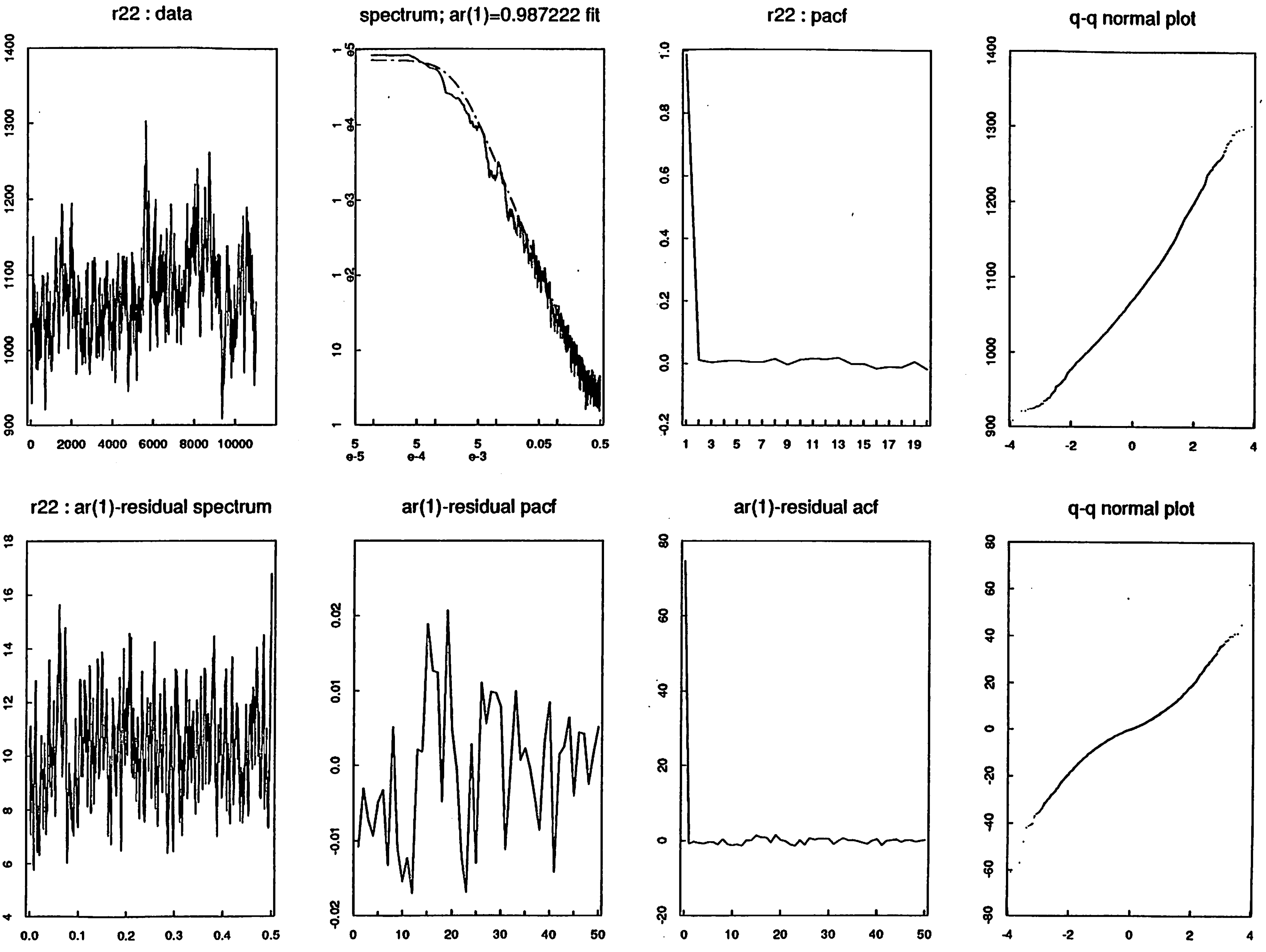

r23 : data
1100
1200
1300
1400
1500
1600
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.967534 fit
1
10
1e2
1e3
1e4
1e5
5e-5
5e-4
5e-3
0.05
0.5
r23 : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
7
9
11
13
15
17
19
q-q normal plot
1100
1200
1300
1400
1500
1600
-4
-2
0
2
4
r23 : ar(1)-residual spectrum
15
20
25
30
35
40
45
50
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.03
-0.02
-0.01
0.0
0.01
0.02
0.03
0
10
20
30
40
50
ar(1)-residual acf
-50
0
50
100
150
200
250
0
10
20
30
40
50
q-q normal plot
-100
-50
0
50
100
-4
-2
0
2
4

Fig 31



r22 : data

spectrum; ar(1)=0.987222 fit

r22 : pacf

q-q normal plot

r22 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

fig 32

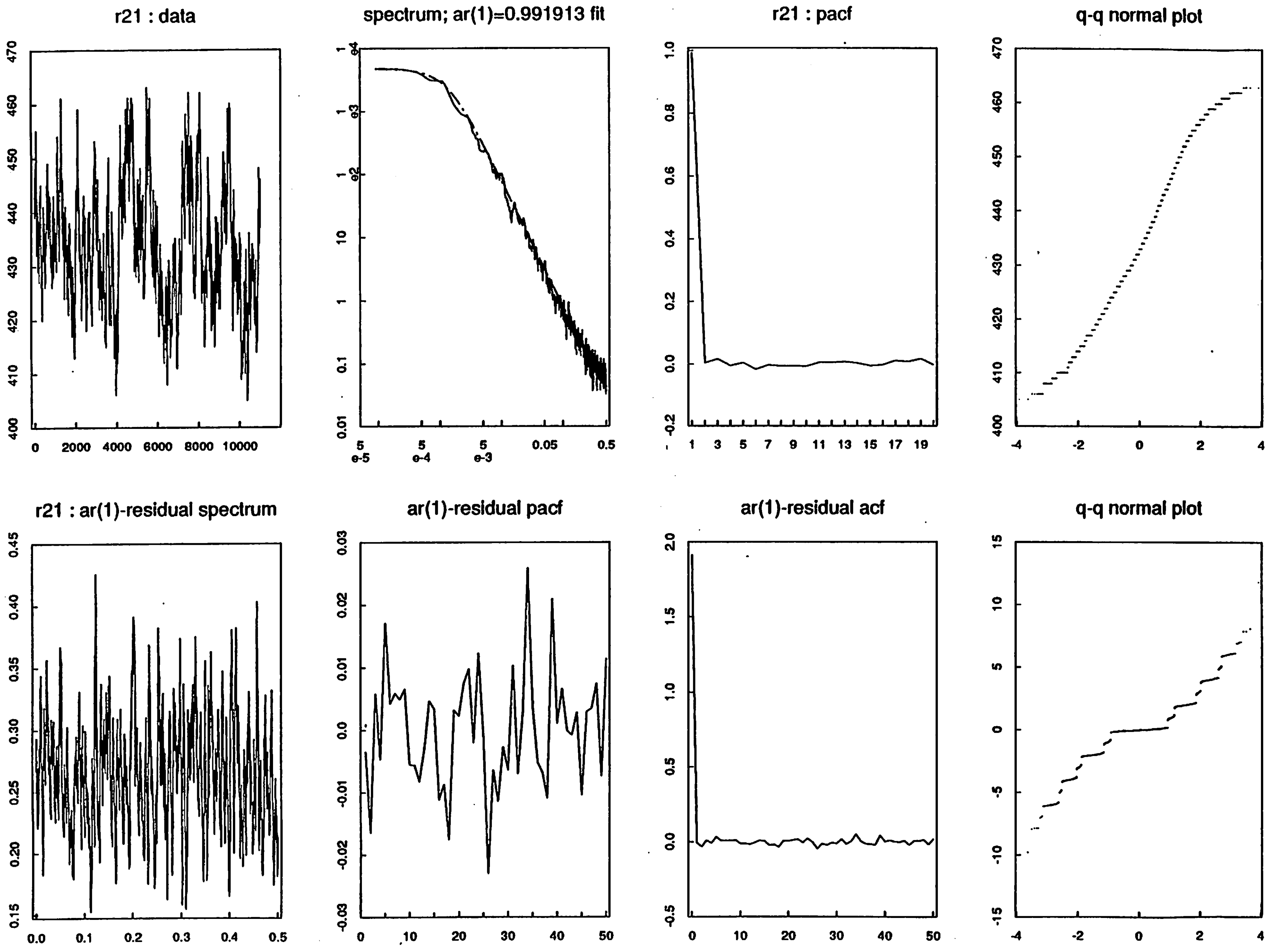
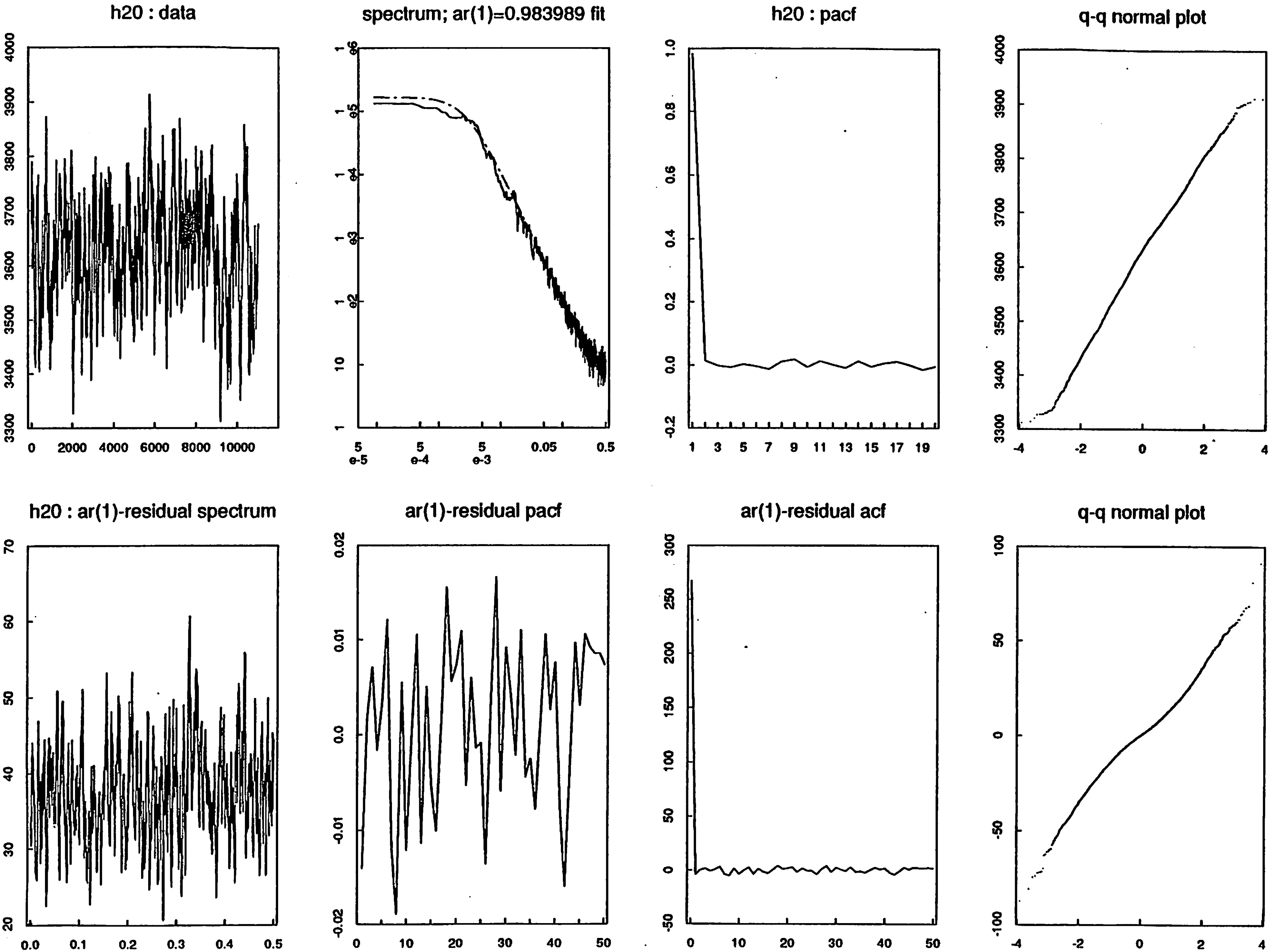

r21 : data
470
460
450
440
430
420
410
400
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.991913 fit
1 e4
1 e3
1 e2
10
1
0.1
0.01
5 e-5
5 e-4
5 e-3
0.05
0.5
r21 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
470
460
450
440
430
420
410
400
-4
-2
0
2
4
r21 : ar(1)-residual spectrum
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.03
0.02
0.01
0.0
-0.01
-0.02
-0.03
0
10
20
30
40
50
ar(1)-residual acf
2.0
1.5
1.0
0.5
0.0
-0.5
0
10
20
30
40
50
q-q normal plot
15
10
5
0
-5
-10
-15
-4
-2
0
2
4

fig 33



h20 : data
4000
3900
3800
3700
3600
3500
3400
3300
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.983989 fit
1 e6
1 e5
1 e4
1 e3
1 e2
10
1
5 e-5
5 e-4
5 e-3
0.05
0.5
h20 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
4000
3900
3800
3700
3600
3500
3400
3300
-4
-2
0
2
4
h20 : ar(1)-residual spectrum
70
60
50
40
30
20
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.02
0.01
0.0
-0.01
-0.02
0
10
20
30
40
50
ar(1)-residual acf
300
250
200
150
100
50
0
-50
0
10
20
30
40
50
q-q normal plot
100
50
0
-50
-100
-4
-2
0
2
4

Fig 34

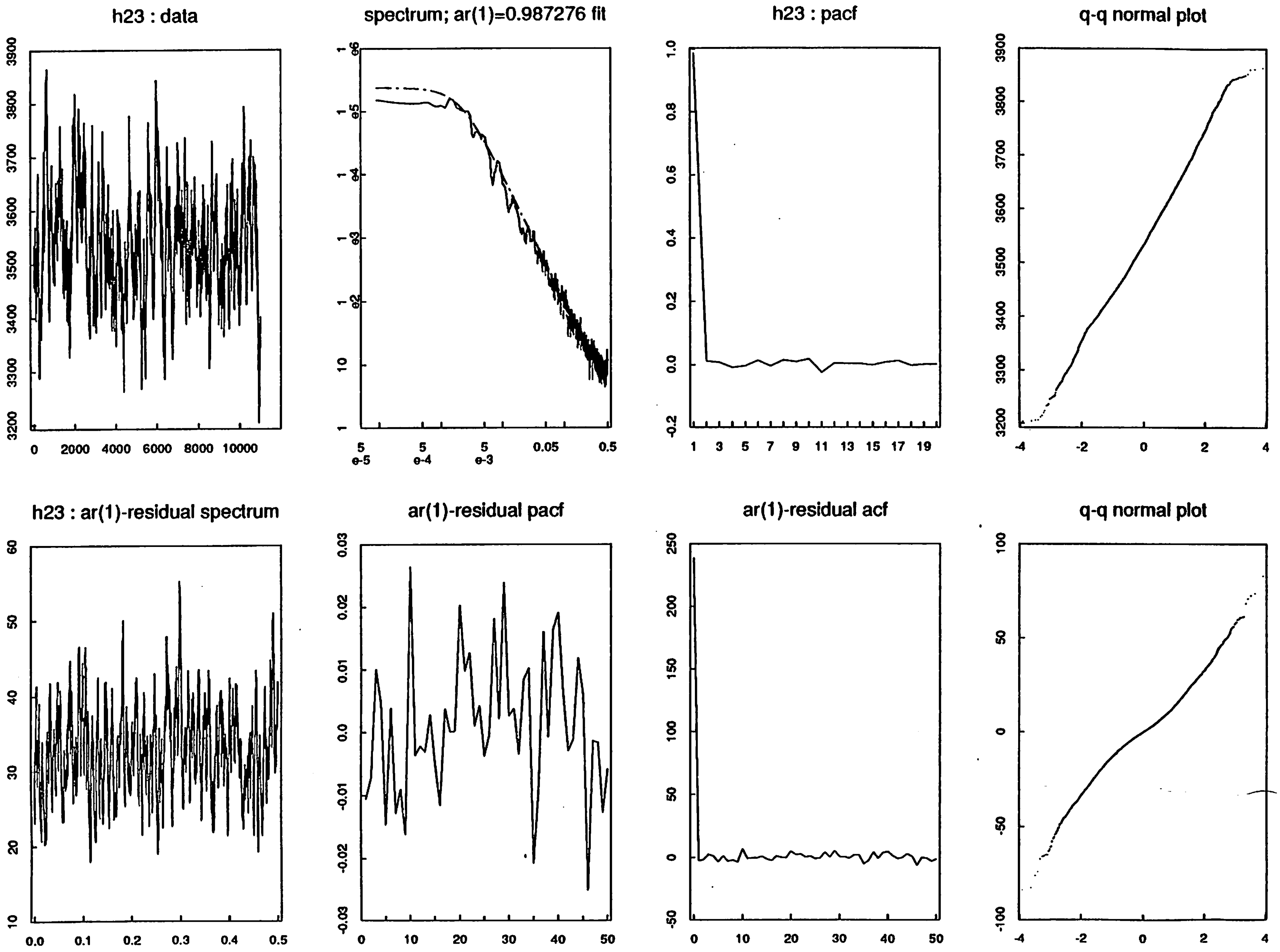

h23 : data
3900
3800
3700
3600
3500
3400
3300
3200
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.987276 fit
1 e6
1 e5
1 e4
1 e3
1 e2
10
1
5 e-5
5 e-4
5 e-3
0.05
0.5
h23 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
3900
3800
3700
3600
3500
3400
3300
3200
-4
-2
0
2
4
h23 : ar(1)-residual spectrum
60
50
40
30
20
10
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.03
0.02
0.01
0.0
-0.01
-0.02
-0.03
0
10
20
30
40
50
ar(1)-residual acf
250
200
150
100
50
0
-50
0
10
20
30
40
50
q-q normal plot
100
50
0
-50
-100
-4
-2
0
2
4

fig 35

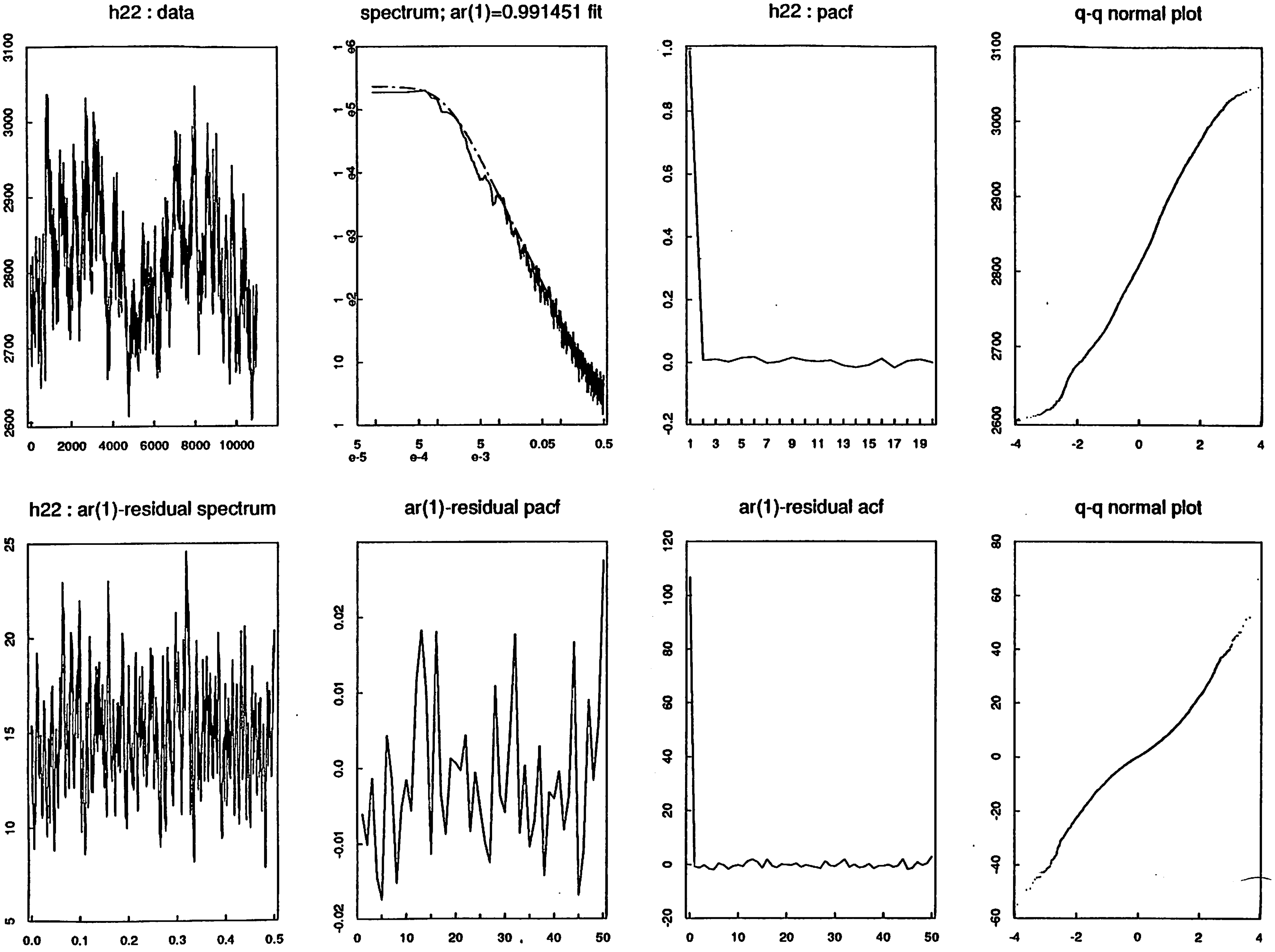

h22 : data
3100
3000
2900
2800
2700
2600
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.991451 fit
1 e6
1 e5
1 e4
1 e3
1 e2
10
1
5 e-5
5 e-4
5 e-3
0.05
0.5
h22 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
3100
3000
2900
2800
2700
2600
-4
-2
0
2
4
h22 : ar(1)-residual spectrum
25
20
15
10
5
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.02
0.01
0.0
-0.01
-0.02
0
10
20
30
40
50
ar(1)-residual acf
120
100
80
60
40
20
0
-20
0
10
20
30
40
50
q-q normal plot
80
60
40
20
0
-20
-40
-60
-4
-2
0
2
4

Fig 36

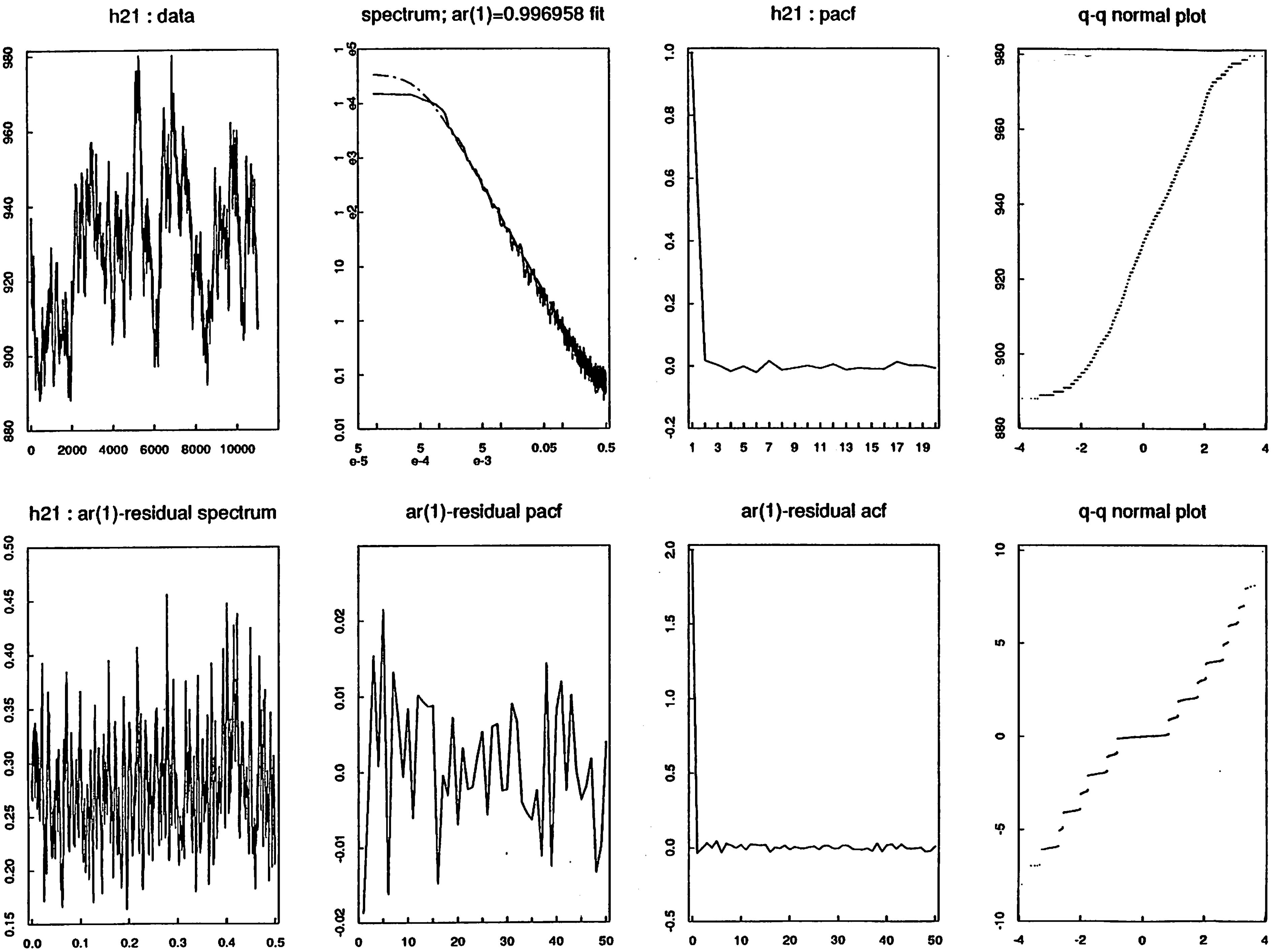

h21 : data
880
900
920
940
960
980
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.996958 fit
0.01
0.1
1
10
1e2
1e3
1e4
1e5
5e-5
5e-4
5e-3
0.05
0.5
h21 : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
7
9
11
13
15
17
19
q-q normal plot
880
900
920
940
960
980
-4
-2
0
2
4
h21 : ar(1)-residual spectrum
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.02
-0.01
0.0
0.01
0.02
0
10
20
30
40
50
ar(1)-residual acf
-0.5
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
50
q-q normal plot
-10
-5
0
5
10
-4
-2
0
2
4

Fig 37

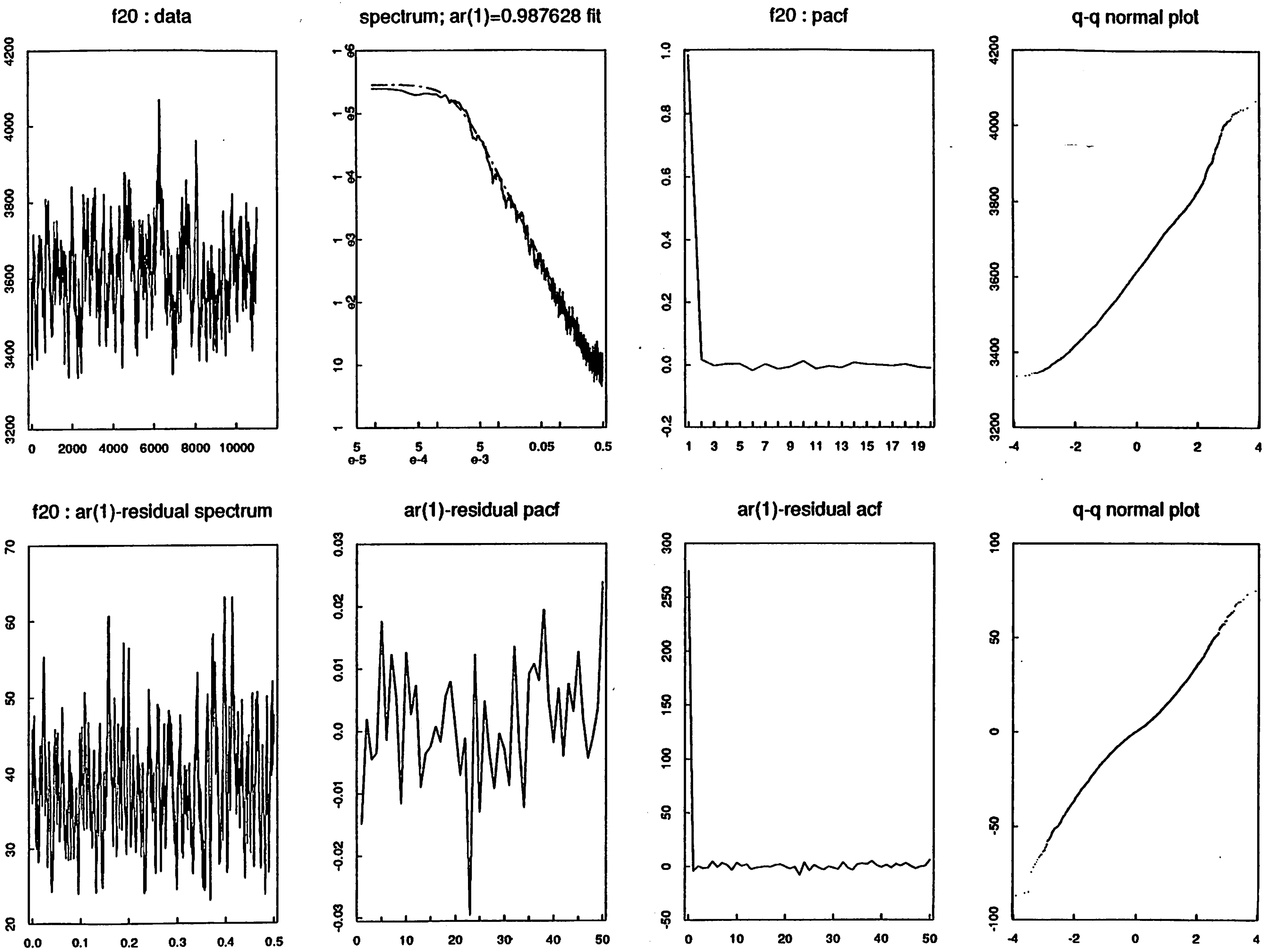
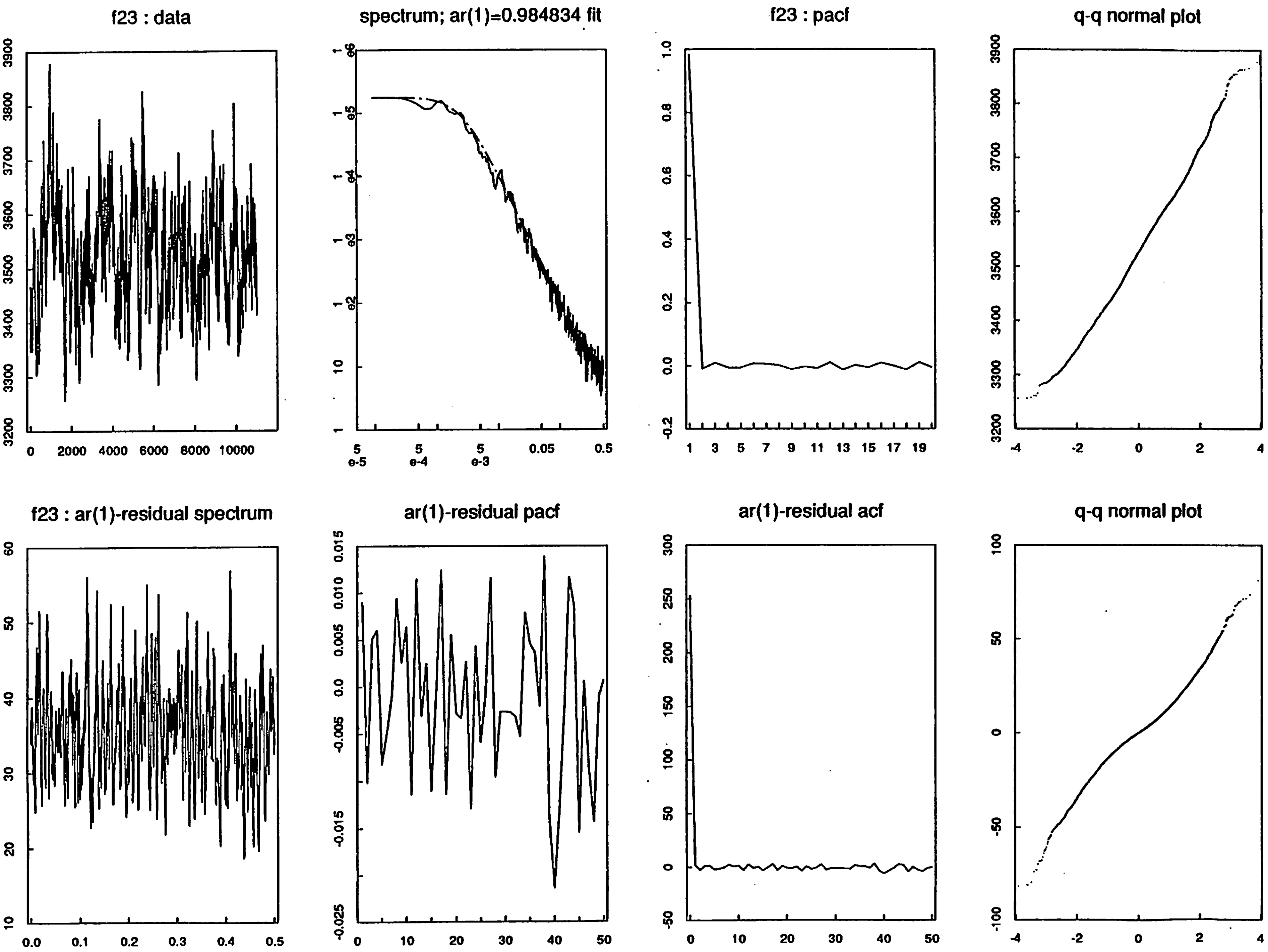

f20 : data
4200
4000
3800
3600
3400
3200
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.987628 fit
1 e6
1 e5
1 e4
1 e3
1 e2
10
1
5 e-5
5 e-4
5 e-3
0.05
0.5
f20 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
4200
4000
3800
3600
3400
3200
-4
-2
0
2
4
f20 : ar(1)-residual spectrum
70
60
50
40
30
20
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.03
0.02
0.01
0.0
-0.01
-0.02
-0.03
0
10
20
30
40
50
ar(1)-residual acf
300
250
200
150
100
50
0
-50
0
10
20
30
40
50
q-q normal plot
100
50
0
-50
-100
-4
-2
0
2
4

fig 38



f23 : data

spectrum; ar(1)=0.984834 fit

f23 : pacf

q-q normal plot

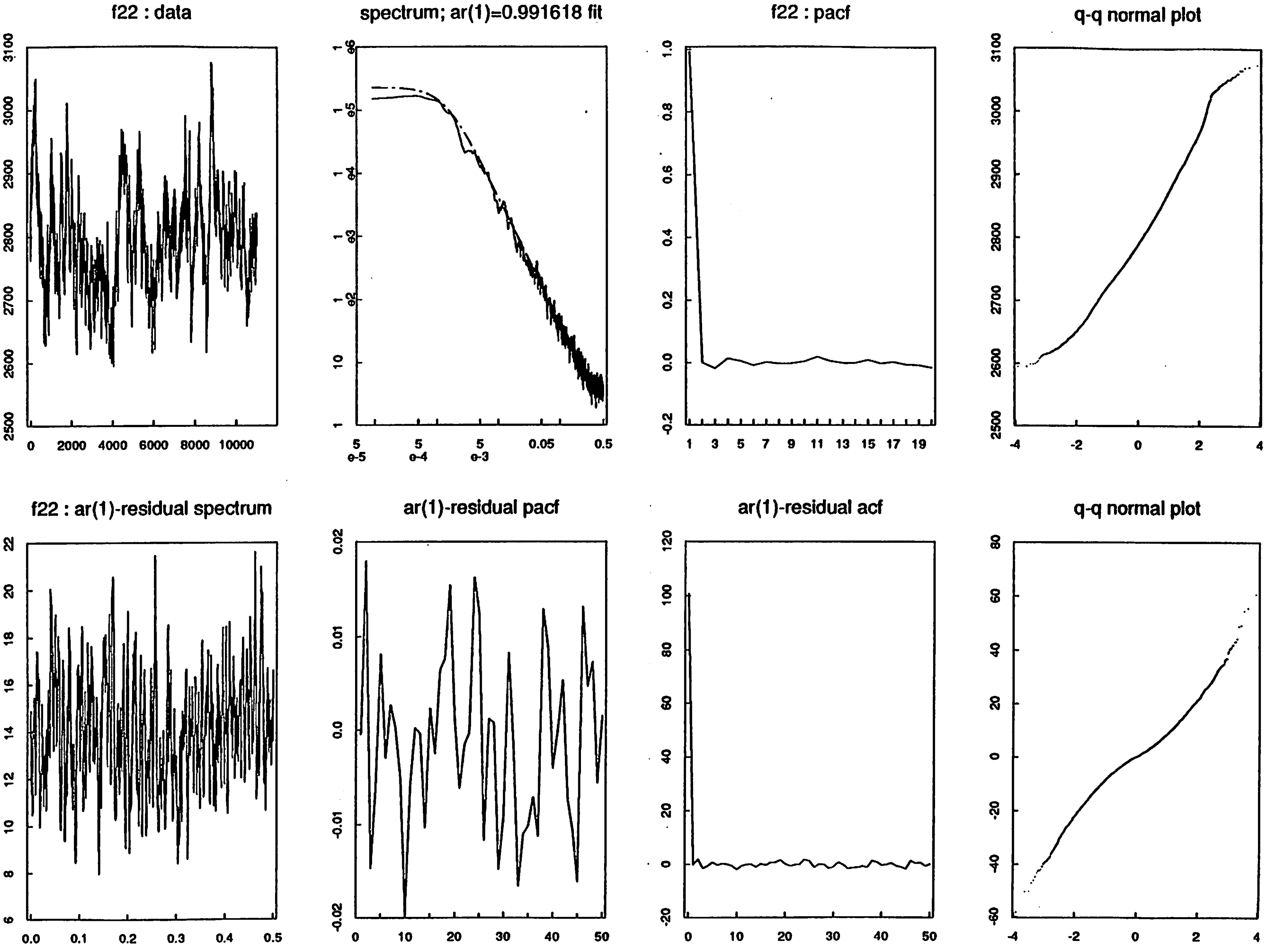
f23 : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

Fig 39



f22 : data
2500
2600
2700
2800
2900
3000
3100
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.991618 fit
1e6
1e5
1e4
1e3
1e2
10
1
5e-5
5e-4
5e-3
0.05
0.5
f22 : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
3
5
7
9
11
13
15
17
19
q-q normal plot
2500
2600
2700
2800
2900
3000
3100
-4
-2
0
2
4
f22 : ar(1)-residual spectrum
6
8
10
12
14
16
18
20
22
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.02
-0.01
0.0
0.01
0.02
0
10
20
30
40
50
ar(1)-residual acf
-20
0
20
40
60
80
100
120
0
10
20
30
40
50
q-q normal plot
-60
-40
-20
0
20
40
60
80
-4
-2
0
2
4

fig 40

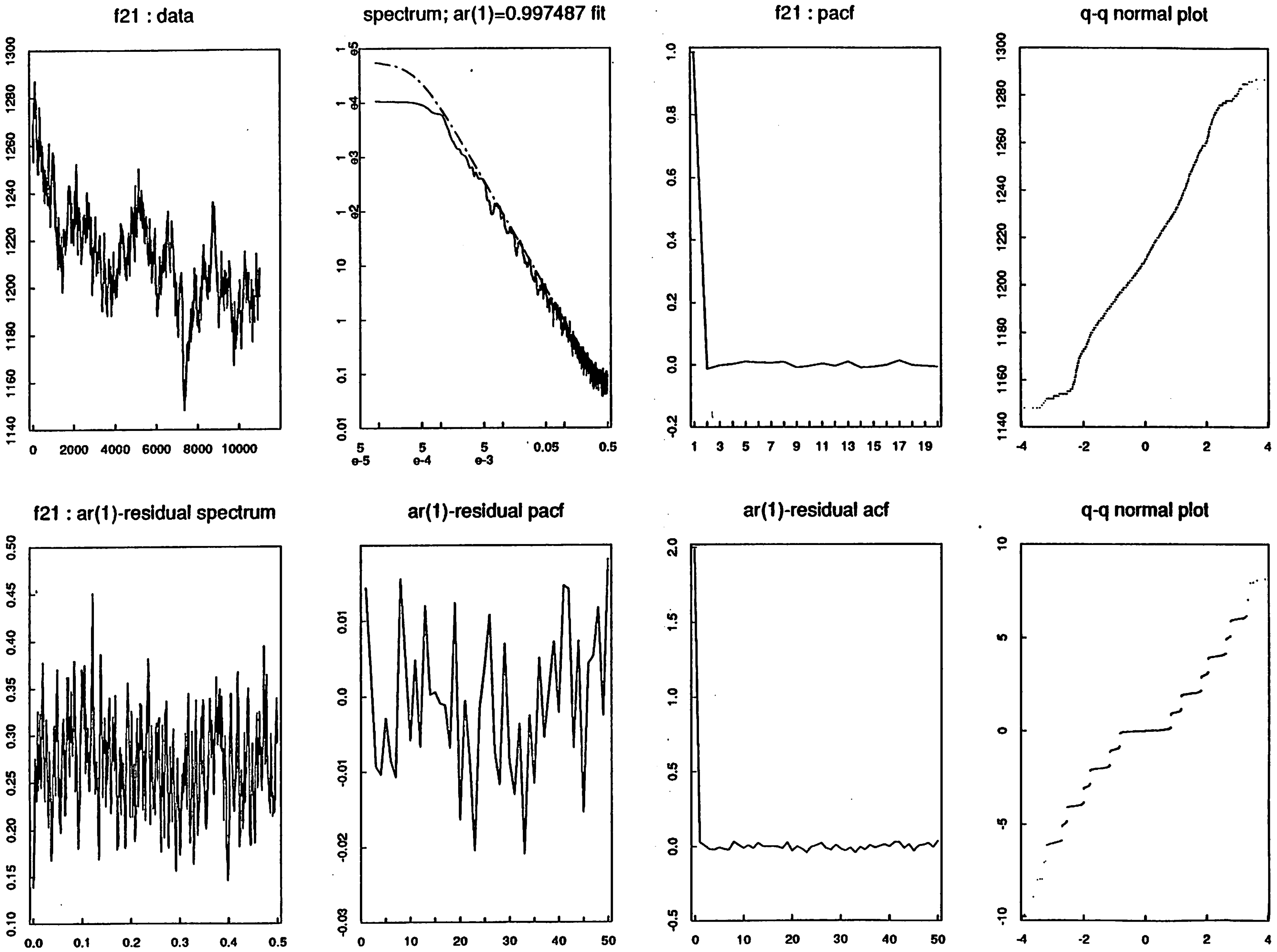
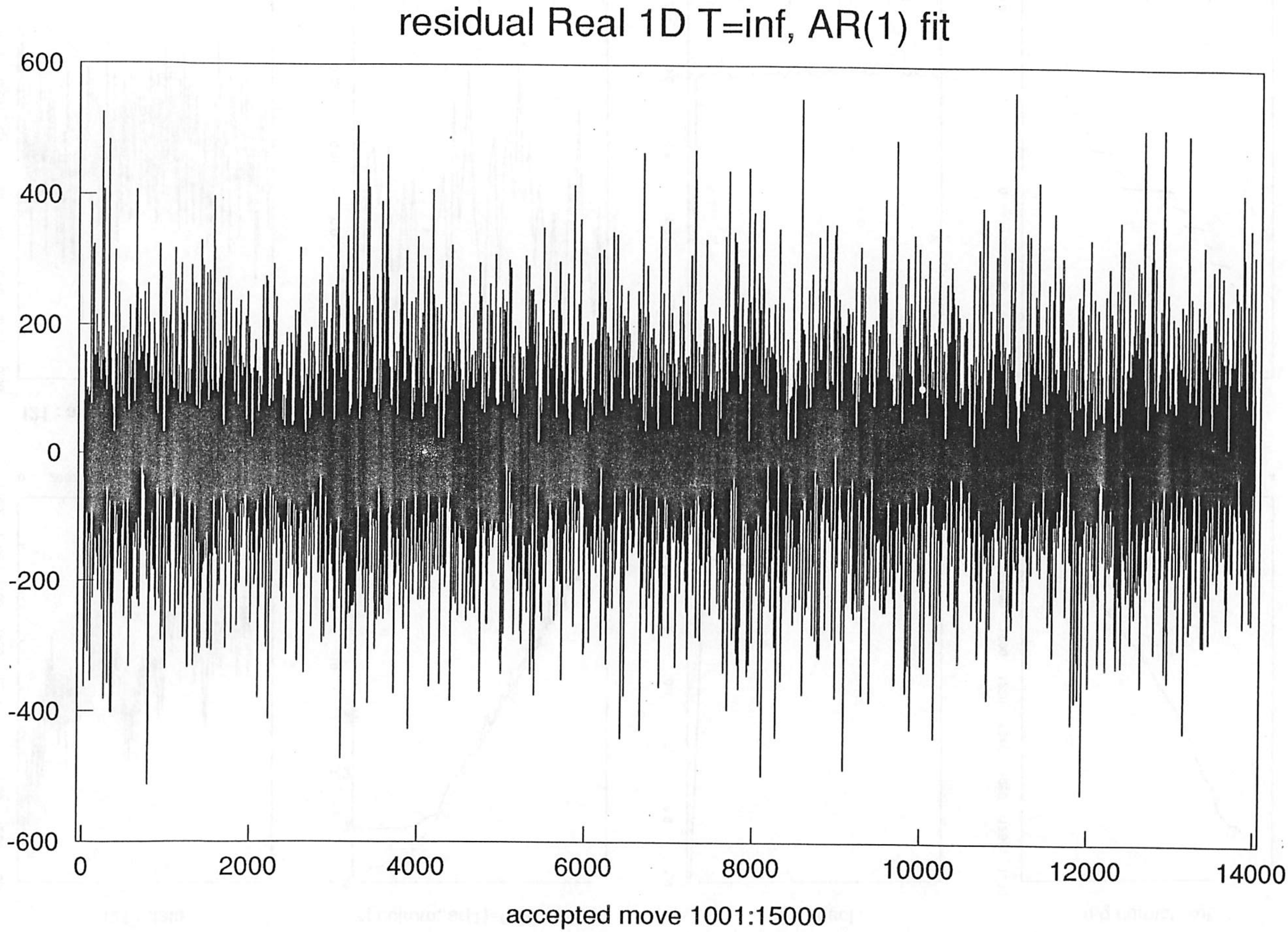
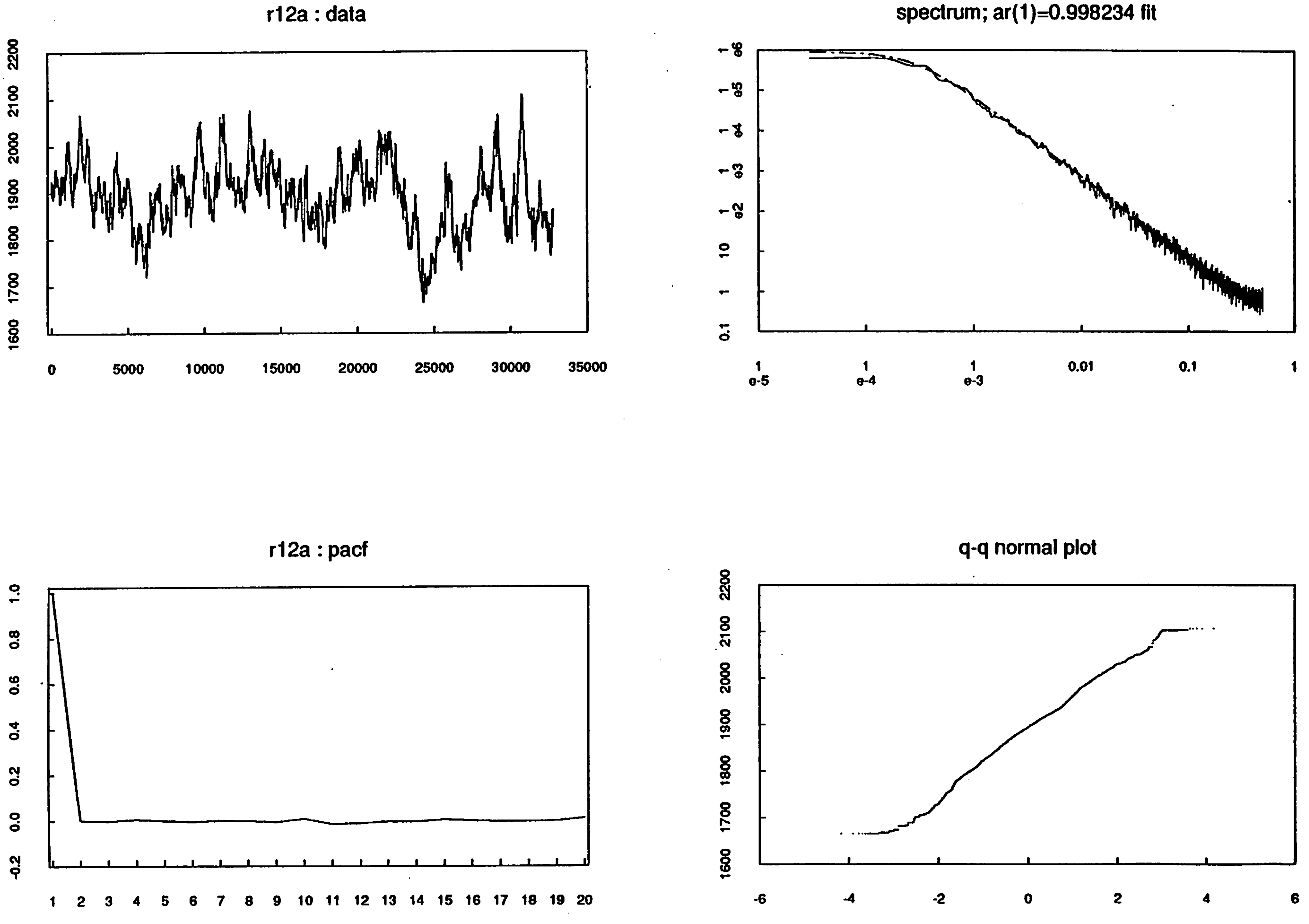

f21 : data
1140
1160
1180
1200
1220
1240
1260
1280
1300
0
2000
4000
6000
8000
10000
spectrum; ar(1)=0.997487 fit
1e5
1e4
1e3
1e2
10
1
0.1
0.01
5 e-5
5 e-4
5 e-3
0.05
0.5
f21 : pacf
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
1
3
5
7
9
11
13
15
17
19
q-q normal plot
1140
1160
1180
1200
1220
1240
1260
1280
1300
-4
-2
0
2
4
f21 : ar(1)-residual spectrum
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
0.01
0.0
-0.01
-0.02
-0.03
0
10
20
30
40
50
ar(1)-residual acf
2.0
1.5
1.0
0.5
0.0
-0.5
0
10
20
30
40
50
q-q normal plot
10
5
0
-5
-10
-4
-2
0
2
4

fig 41



residual Real 1D T=inf, AR(1) fit
600
400
200
0
-200
-400
-600
0
2000
4000
6000
8000
10000
12000
14000
accepted move 1001:15000

fig 42a

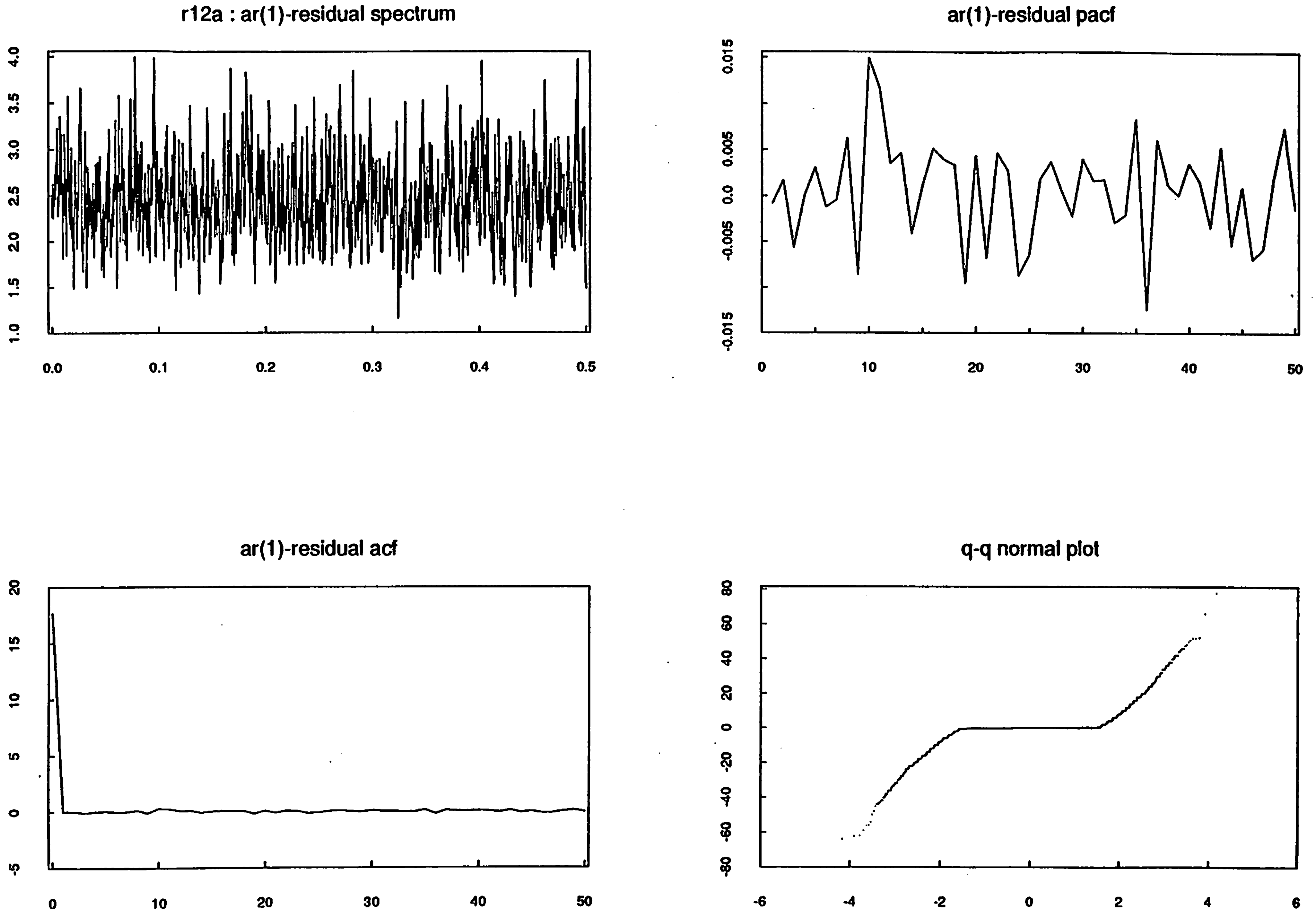

r12a : data

spectrum; ar(1)=0.998234 fit

r12a : pacf

q-q normal plot

Fig 42b



r12a : ar(1)-residual spectrum

ar(1)-residual pacf

ar(1)-residual acf

q-q normal plot

fig 43a

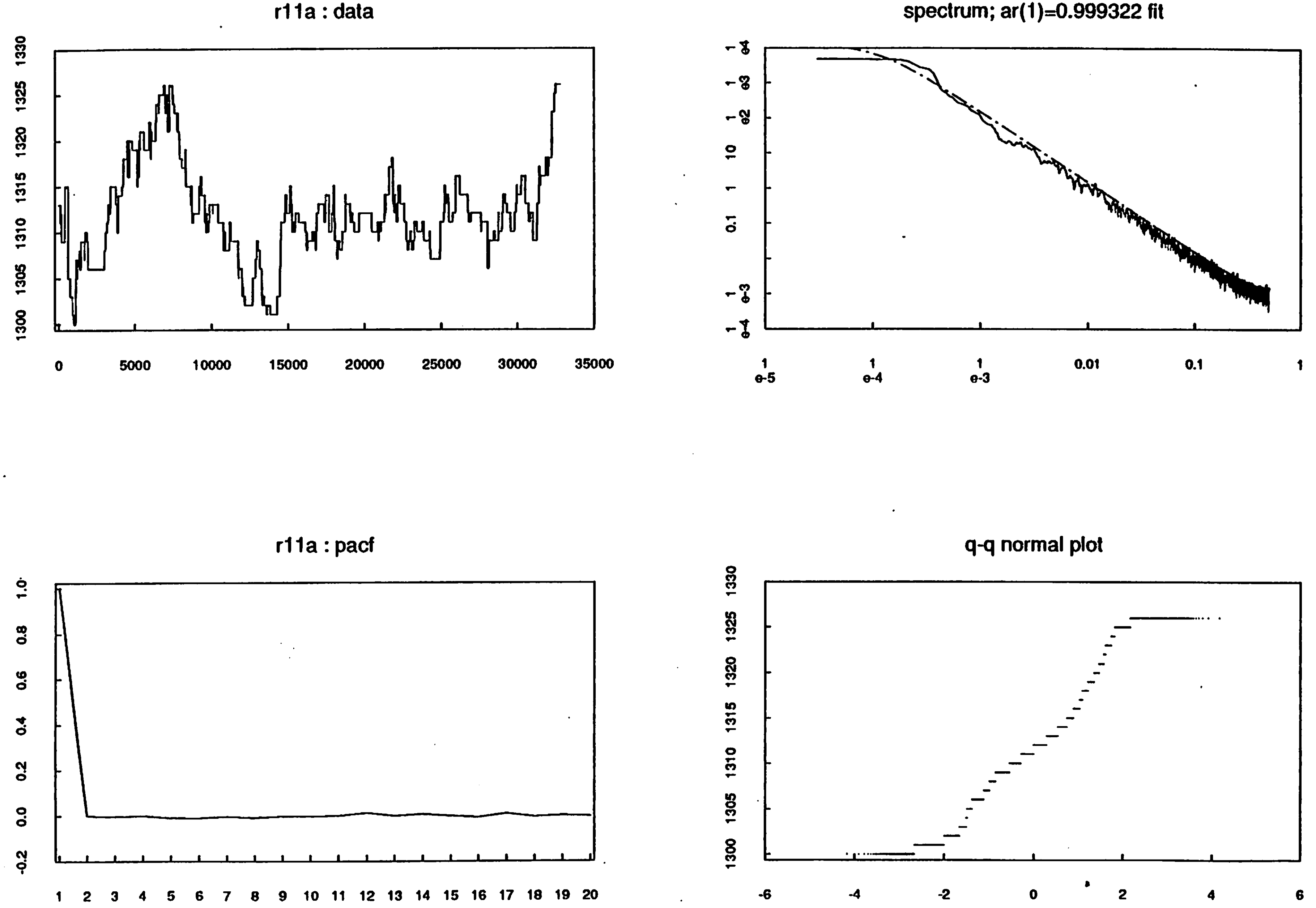

r11a : data
1300
1305
1310
1315
1320
1325
1330
0
5000
10000
15000
20000
25000
30000
35000
spectrum; ar(1)=0.999322 fit
1 e4
1 e3
1 e2
10
1
0.1
1 e-3
1 e-4
1 e-5
1 e-4
1 e-3
0.01
0.1
1
r11a : pacf
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
q-q normal plot
1300
1305
1310
1315
1320
1325
1330
-6
-4
-2
0
2
4
6

Fig 43b

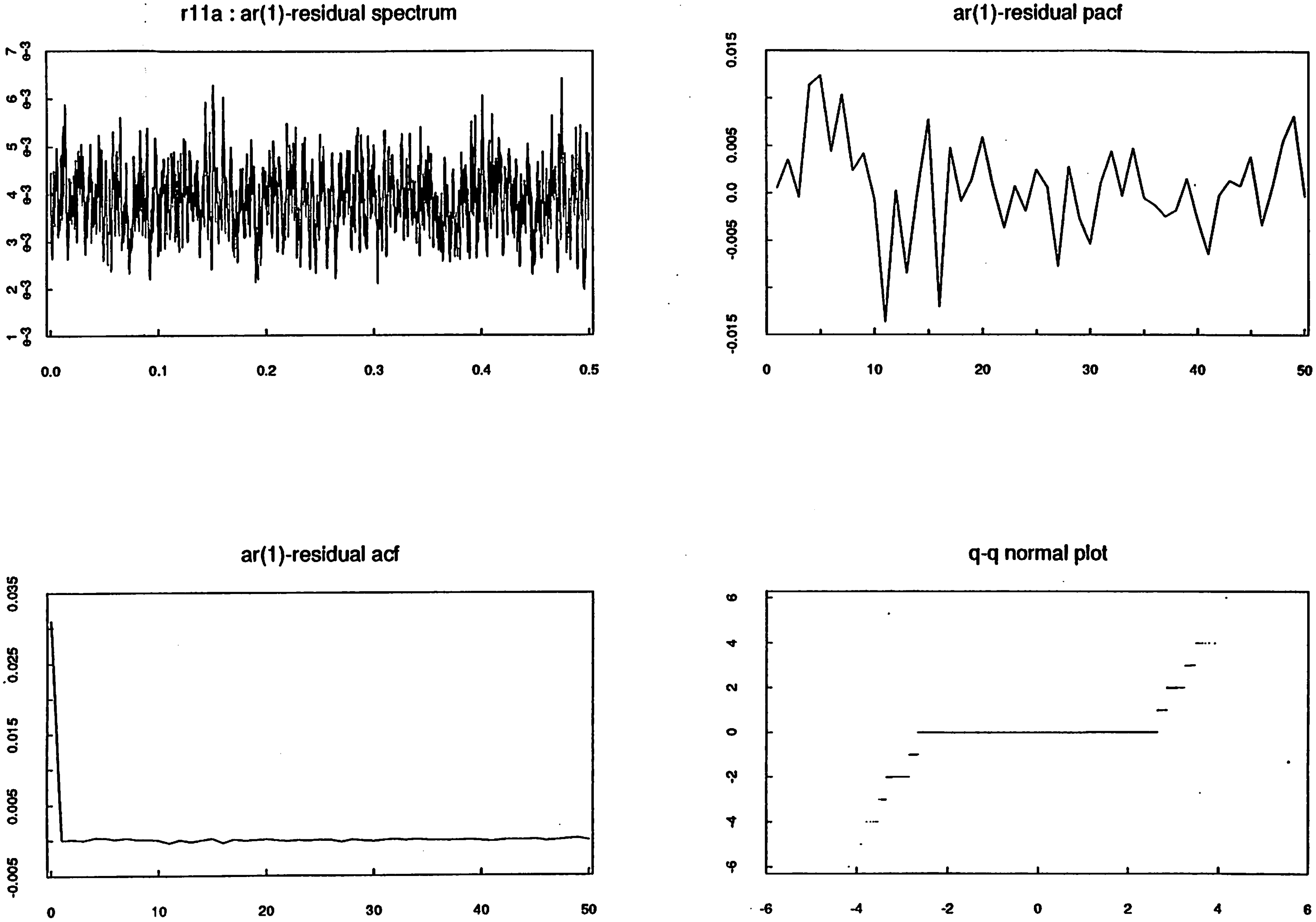

r11a : ar(1)-residual spectrum
1 e-3
2 e-3
3 e-3
4 e-3
5 e-3
6 e-3
7 e-3
0.0
0.1
0.2
0.3
0.4
0.5
ar(1)-residual pacf
-0.015
-0.005
0.0
0.005
0.015
0
10
20
30
40
50
ar(1)-residual acf
-0.005
0.005
0.015
0.025
0.035
0
10
20
30
40
50
q-q normal plot
-6
-4
-2
0
2
4
6
-6
-4
-2
0
2
4
6

fig 44

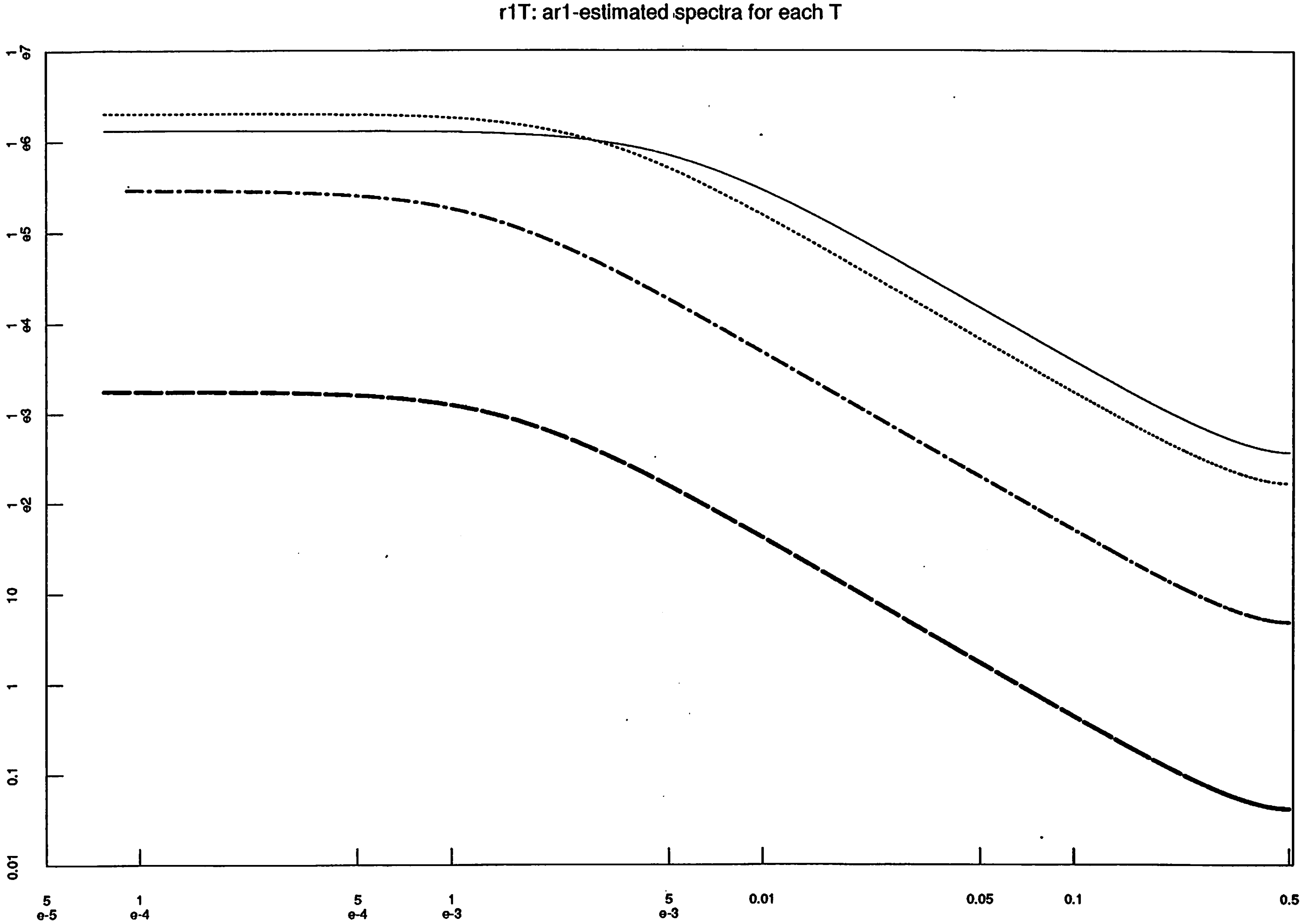

r1T: ar1-estimated spectra for each T
1e7
1e6
1e5
1e4
1e3
1e2
10
1
0.1
0.01
5e-5
1e-4
5e-4
1e-3
5e-3
0.01
0.05
0.1
0.5

fig 45

h1T: ar1-estimated spectra for each T

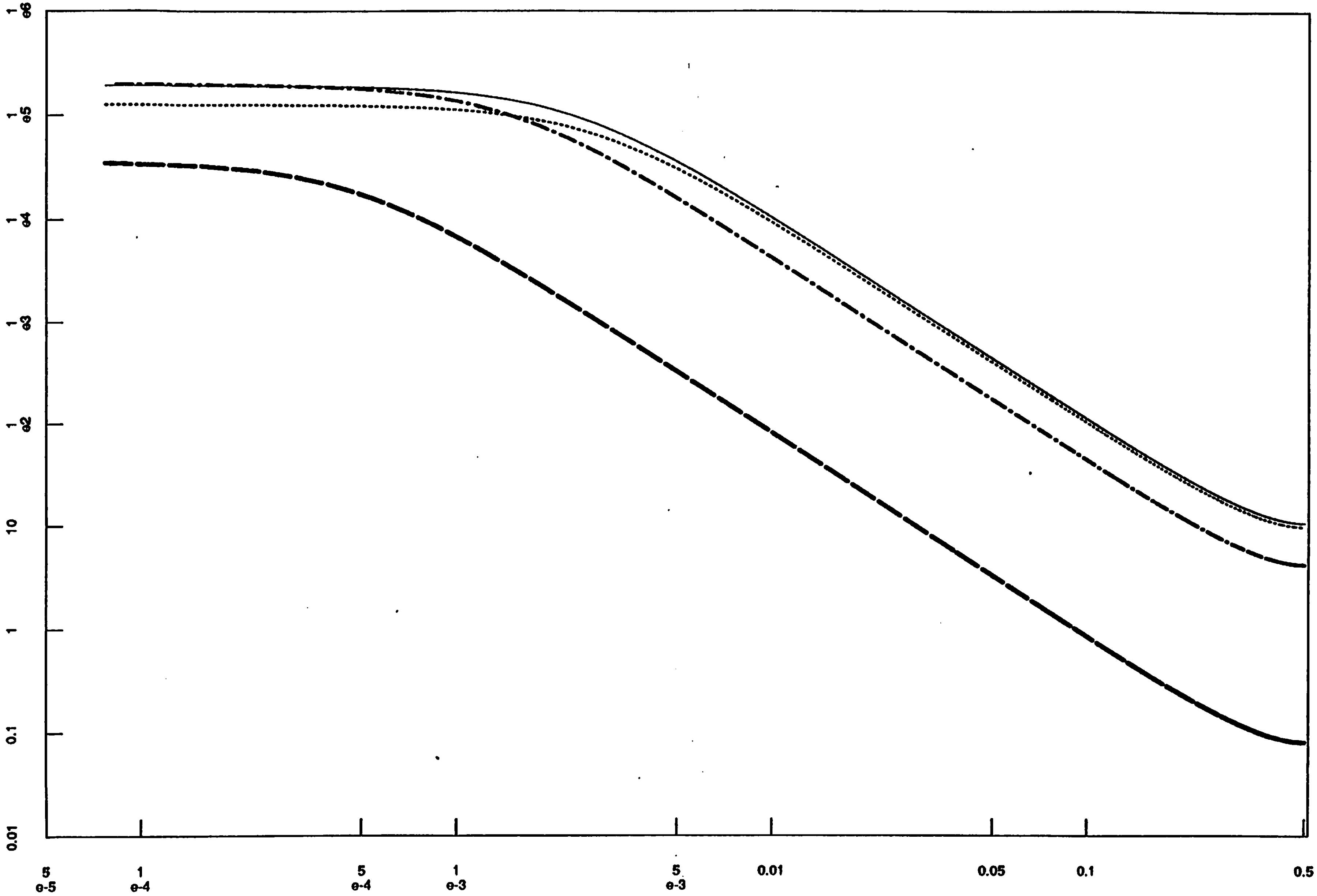


fig 46

f1T: ar1-estimated spectra for each T

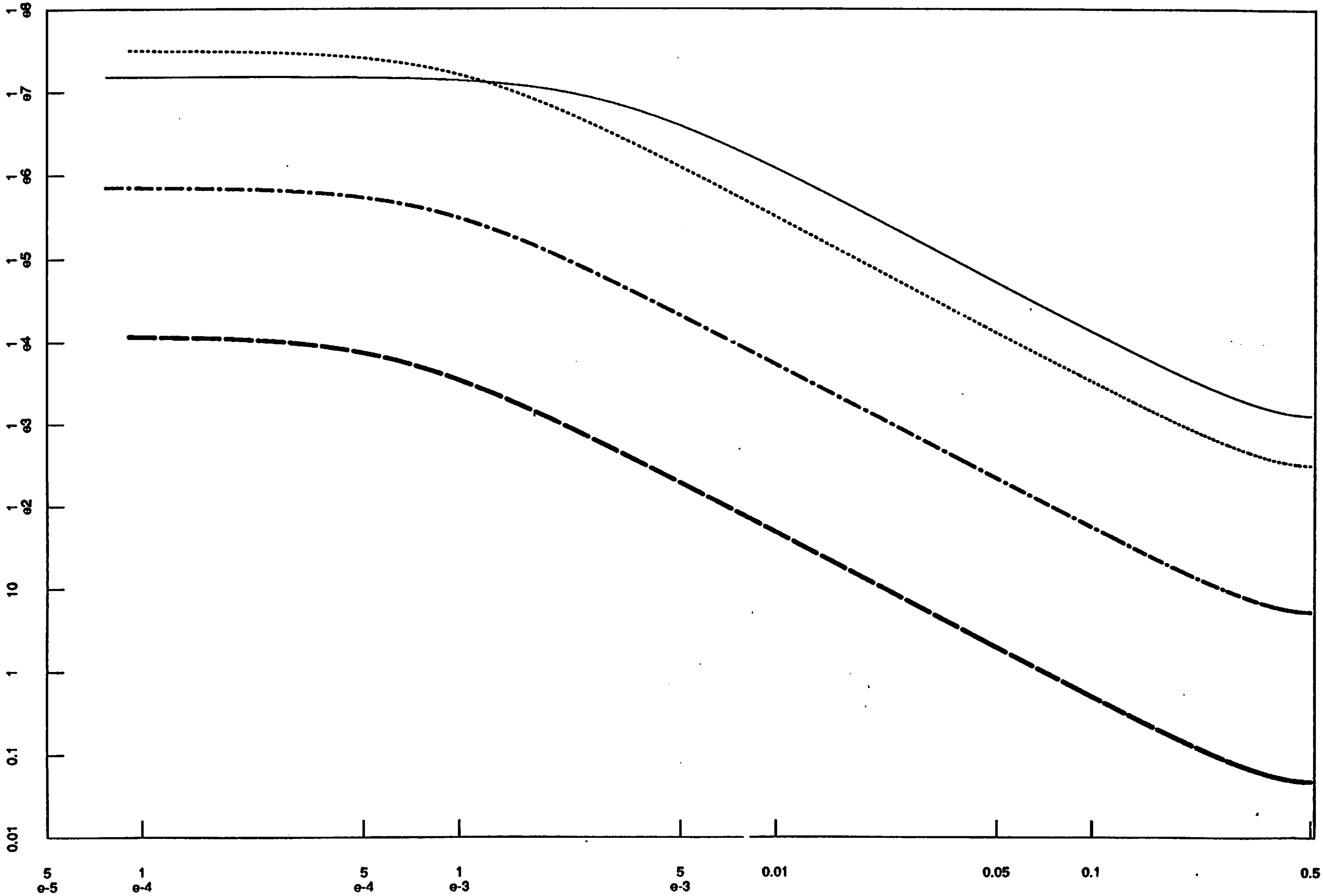
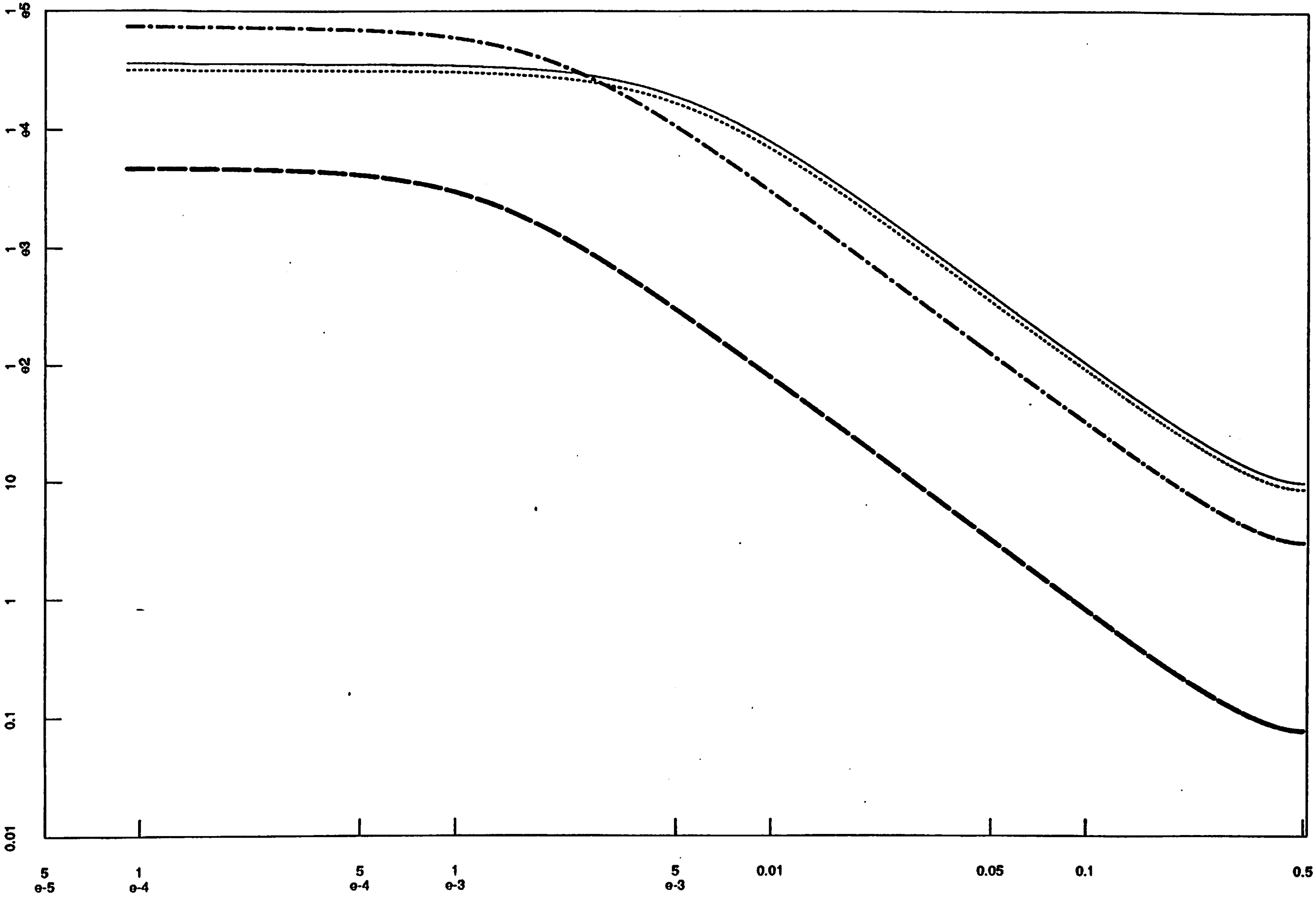


fig 41



r2T: ar1-estimated spectra for each T

Fig 48

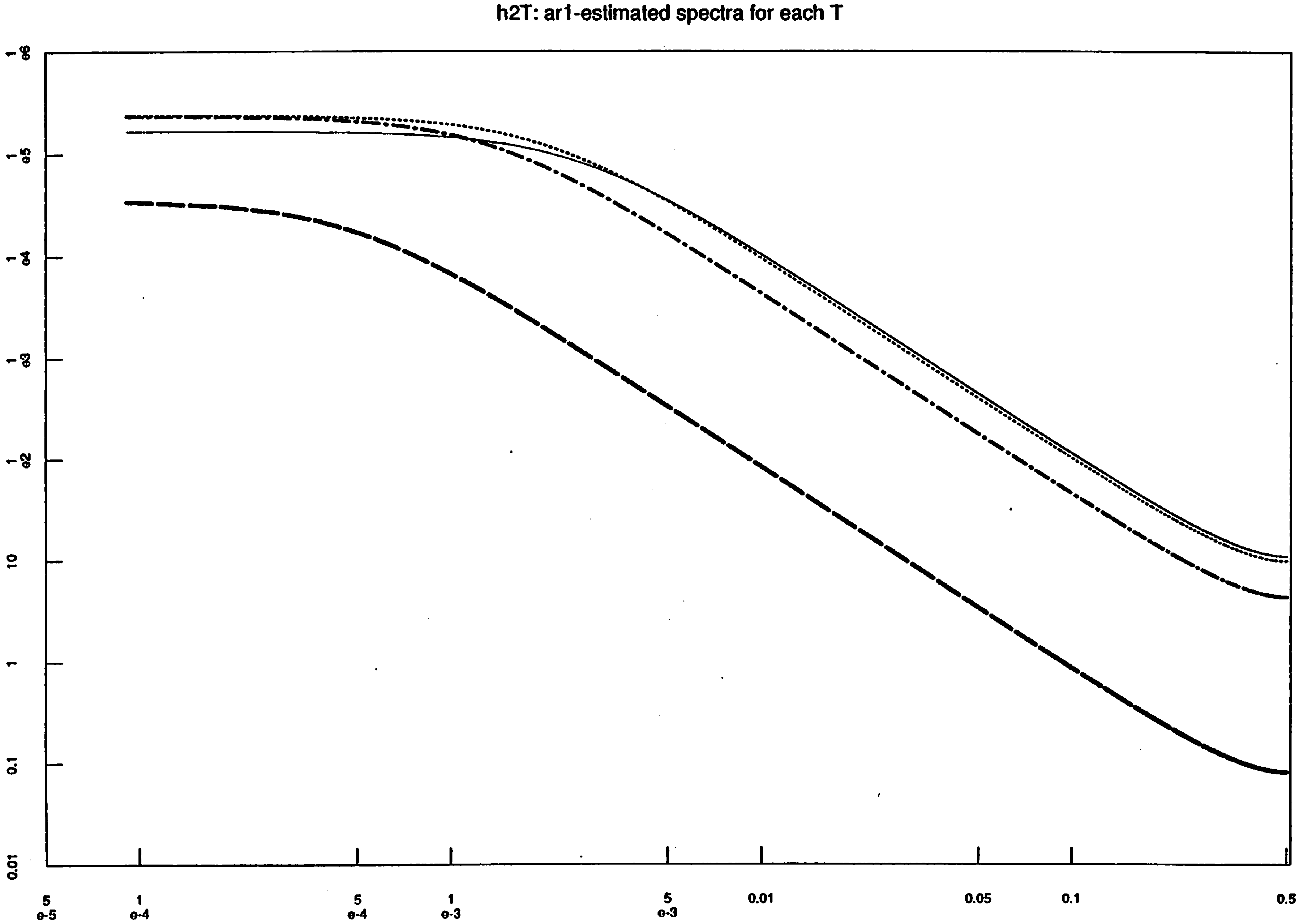
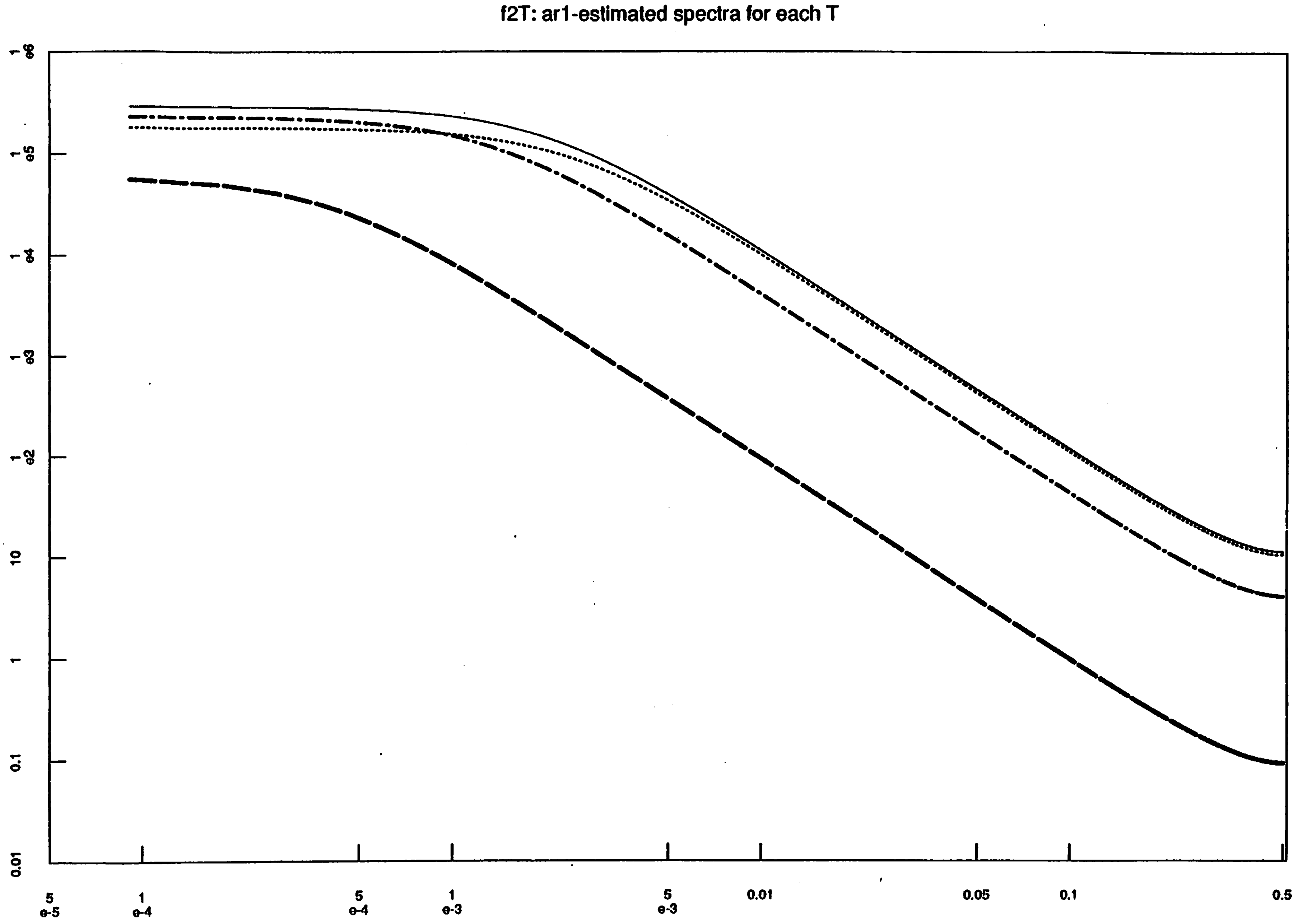

h2T: ar1-estimated spectra for each T
1e6
1e5
1e4
1e3
1e2
10
1
0.1
0.01
5e-5
1e-4
5e-4
1e-3
5e-3
0.01
0.05
0.1
0.5

fig 49



f2T: ar1-estimated spectra for each T