ELECTRONOTES
1016 Hanshaw Road
Ithaca, NY 14850

APPLICATION NOTE NO. 377
February 2012

FALSE IDEAS ABOUT RANDOM SEQUENCES

We all know how to generate random sequences of numbers. We could toss coins or throw dice, or even measure emissions from a radioactive source. For practical purposes (computer games, music synthesis, even Monte Carlo calculations), we are often quite comfortable with pseudo random generators. Baring a "dissection" of the pseudo-random algorithm, the numbers in a sequence are all a surprise. So here is what many people think they know:

(1) Samples of a random generated sequence are white noise.

(2) There is something "special" about such random sequences.

(3) The random sequence has a mean of zero.

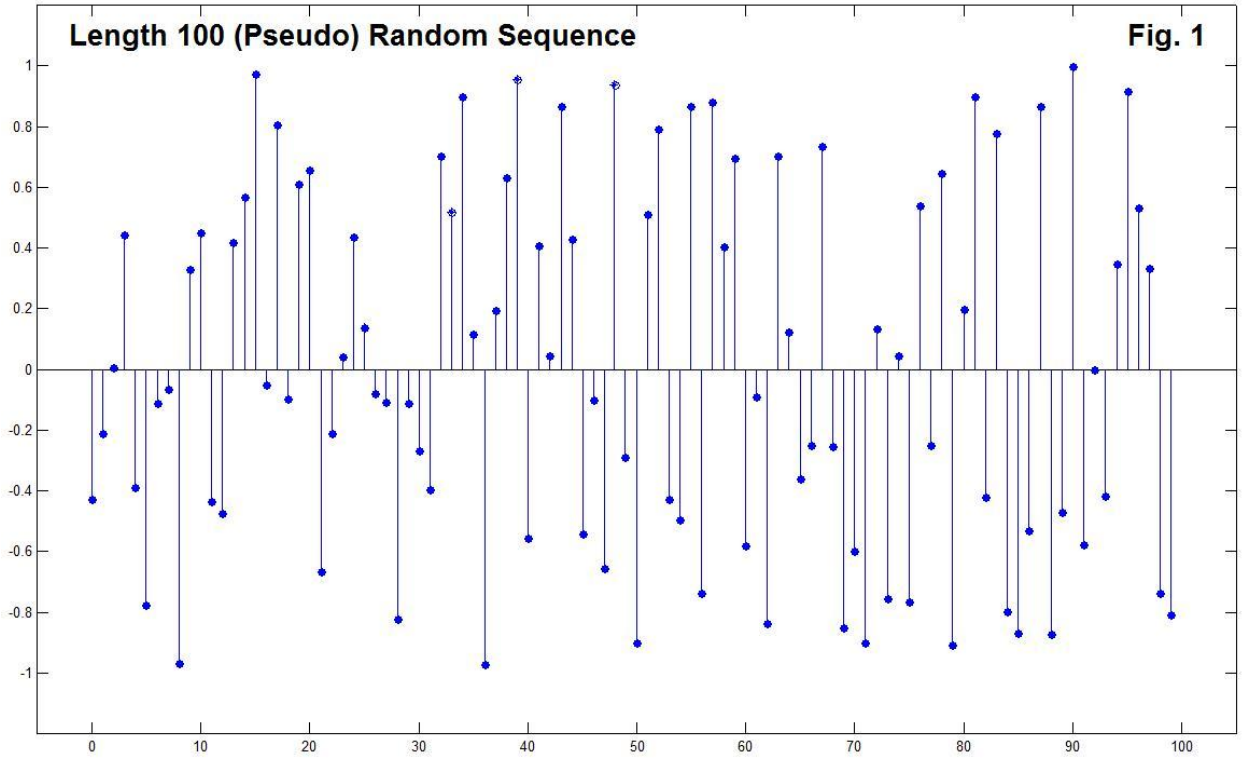
(4) The random sequence has no linear (or other) trend.

(5) The spectrum of our random sequences is flat.

(6) You can get a good idea about whether a sequence is random, or not, by looking at it. We are good at that judgment.

Let's try a sequence. Suppose we use Matlab to generate a pseudo-random sequence with their ***rand*** function. Specifically, since rand gives us a uniform distribution between 0 and 1, we usually offset this by -1/2 and double it for a sequence between -1 and +1 Choosing a length 100 sequence: x = 2*(rand(1,100)-0.5). This is the most commonly suggested approach. Fig. 1 shows the sequence as a ***stem*** plot. Overall, it looks pretty much like what we would expect of 100 random samples.

Things perhaps look a bit different in the connected plot of Fig. 2 where straight segments are plotted between samples with the stems gone. This is likely misleading, but the plot does much more closely resemble traditional graphs of random signals of the type often seen in physics and engineering books, at least before discrete rather than continuous-time signals were the usual finding.


Length 100 (Pseudo) Random Sequence
Fig. 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60
70
80
90
100

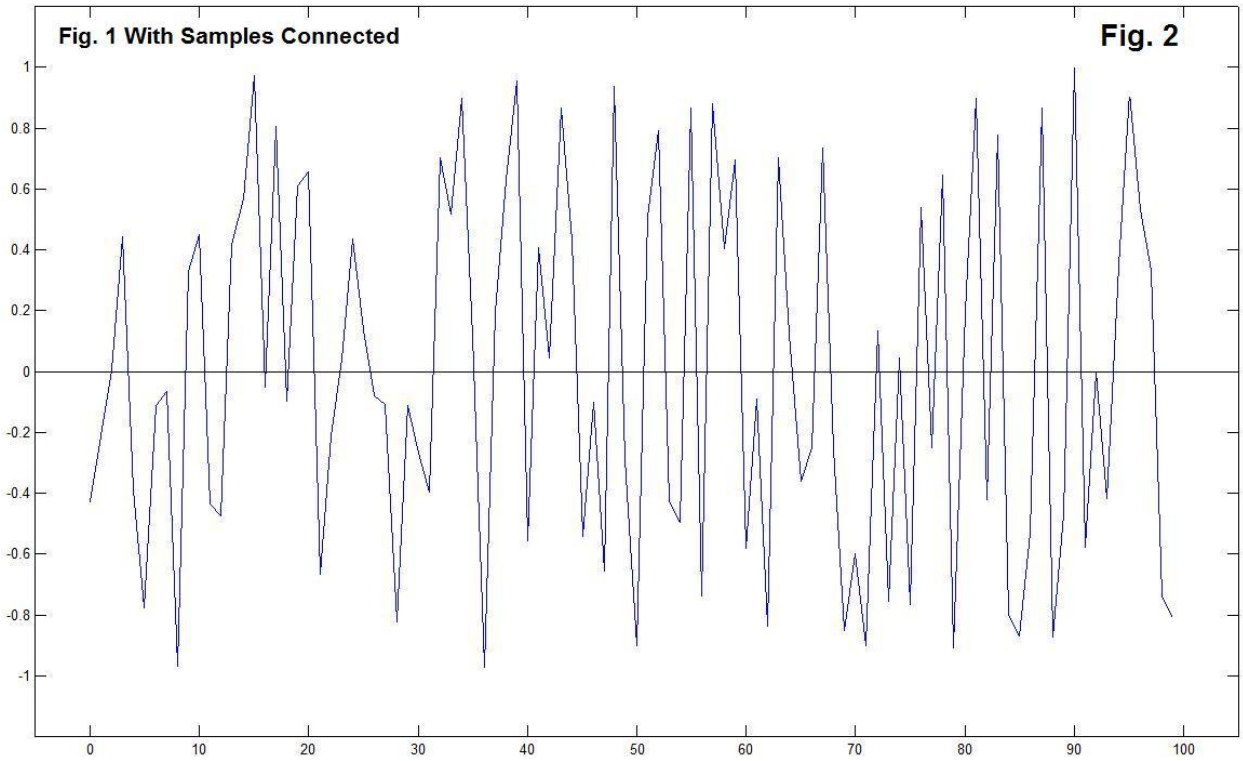


Fig. 1 With Samples Connected
Fig. 2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60
70
80
90
100

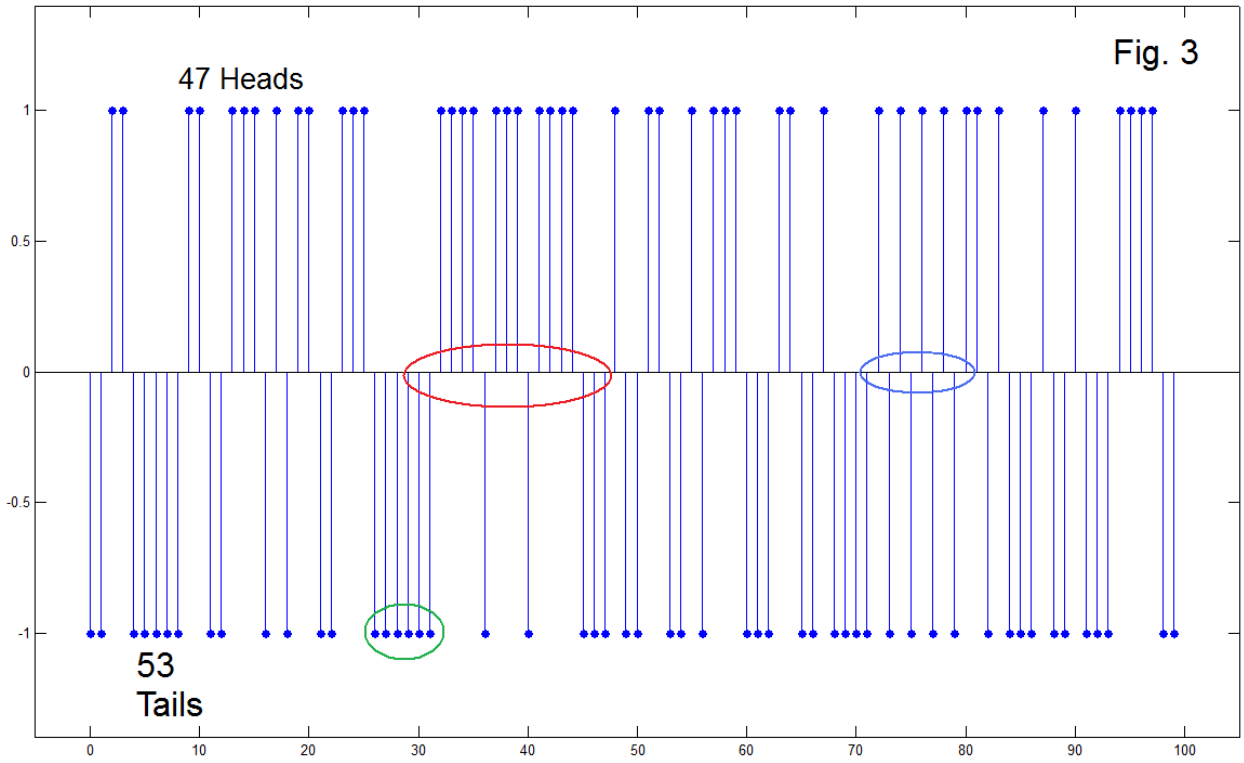


Fig. 3 is a revealing departure from Fig. 1 because here we plot (***stem***) just the signs (+1 or -1) of the samples in the sequence of Fig. 1. Accordingly we have a result that looks like (and is) analogous to the coin-flip experiments plotted in books on probability and statistics. We may immediately think we see problems at this point. The number of heads is not the same as the number of tails. Indeed there are six more tails. Well, we remember now our readings about this. It's NOT anticipated that they will be exactly the same. But in one place (green) there are six tails in a row, and that seems unlikely. In another place (red), there are 19 samples with exact symmetry in time. In yet another place (blue) there are 10 samples alternating. Way too much order!

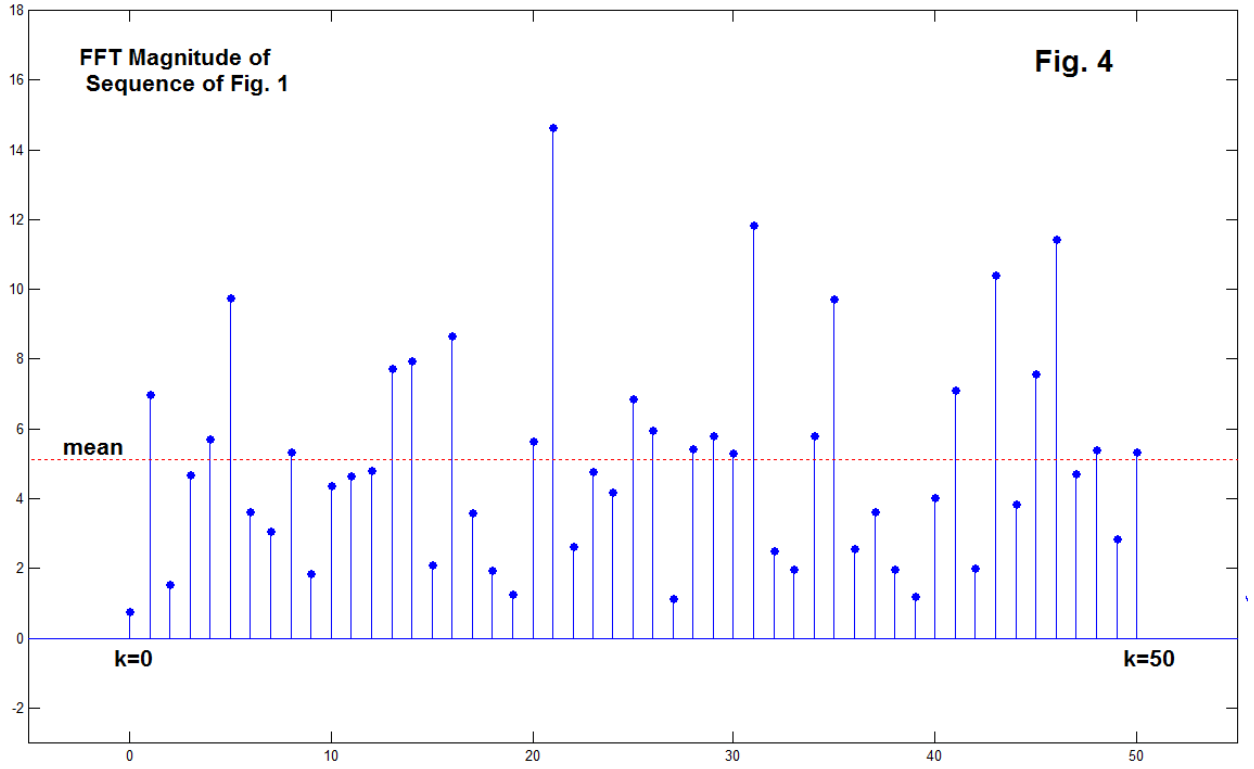
Yet no special effort was made to find this example. This was the fifth run of the program, the first four were expended getting the code right. Subsequent runs turned up different, but seemingly similar remarkable runs and symmetries - particularly as they were revealed by just plotting the signs (heads/tails). Are these things possible, or perhaps even expected? Of course. Most of us have read enough to know that no sequence with obvious patterns is any more or any less likely than one with less apparent order.

But isn't there a temptation run ***rand*** again, and perhaps again and again? We might get something that "looks" better. Or more simply, why not just go in and "edit" out the embarrassing accidents? In a long run of 6 tails, why not flip one or two of those tails to heads? And Just changing one sample would destroy symmetry. After all, isn't the sequence random anyway? Does it matter, therefore, if we change a few samples? They could have occurred without any intervention, after all. Aren't we just making it more random?

The tendency to manipulate a randomization sequence for an experiment (typically medical, psychological, or perhaps opinion sampling) is dangerous, particularly as outcomes are marginally significant. If we are using a random sequence to "surprise" a test subject, it is reasonable to suppose that the human tendency (to hold a run of six straight tails as "suspect", for example) applies to the test subject as well as the experimenter. To any attempt we make to arrive at a "better" randomizer (from selecting sequence that just seem better, to outright editing a head into a "too-long" run of tails) will serve to bias an experiment in favor of joint (false) expectations. Experimenter and subject both knew a "head was overdue" and adjusted accordingly. A miss becomes a hit.

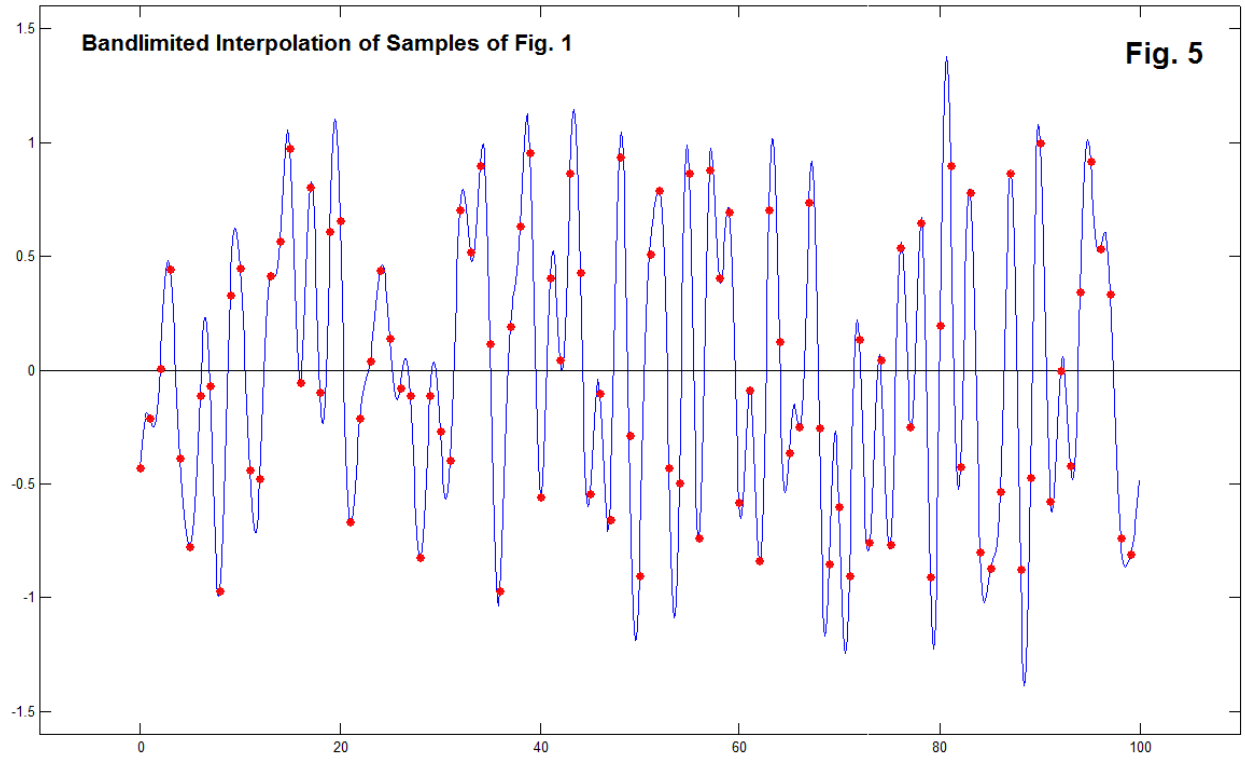
Continuing, the fact that there are more tails than heads in Fig. 3 tells us that that sequence is not zero mean. Well, perhaps the actual sequence of Fig. 1 is? Nope, the mean is -0.0078. Why would we have expected zero mean? Perhaps because we took a uniform distribution between 0 and +1 and subtracted (intentionally) 1/2. But this 1/2 is the mean of a very very long sequence – not of any more usual length. We then perhaps quickly accept a non-zero mean, a DC term, because we know we wanted the overall spectrum to be flat, so it had to have some non-zero DC. The argument starts to almost sound convincing.

So, is the spectrum really white (flat)? In Fig. 4 we look at the sequence in the frequency domain by using the FFT. Here we need to plot only the first half of the FFT magnitude (k=0 to k=50), since the sequence is real and its magnitude is symmetric. The result is not flat. Have we failed? We can try another sequence, and it too will not be flat, but it will be different from Fig. 4, and soon we recognize that the notion of a flat spectrum relates to expectation drawn from an ensemble, not to any one example. If we want to see flat, we need to average. More about this later.



We know the samples of Fig. 1 were obtained using a pseudo-random algorithm. Could they have come from anywhere else? Is it possible they could have been obtained from a bandlimited waveform – in fact, one with a finite number of discrete frequencies? The use of the FFT answers these questions in the affirmative. The FFT can be viewed as N equations in N unknowns (N being the length of the sequence). The FFT converts N given time domain samples to N frequency domain samples. The amount of information is conserved. [The FFT is in general complex, so it is really 2N numbers. But the time-domain is real, so the FFT has an even symmetric real part and an odd symmetry imaginary part, so that's N unique numbers. Just had to mention that.]

Can we easily view the bandlimited waveform corresponding to Fig. 1? Yes, by using the FFT to interpolate additional values between the 100 samples we have. This is done by zero padding the FFT in the middle and taking the inverse FFT. Fig. 5 shows the result – zero-padding by 10:1 (900 zeros added to the center of the FFT). Here instead of plotting samples with stem, we just use plot to better suggest a continuous waveform (and to avoid clutter). At the same time, the original samples are shown as circles. This is very interesting to see it this way.


Bandlimited Interpolation of Samples of Fig. 1
Fig. 5
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100

.

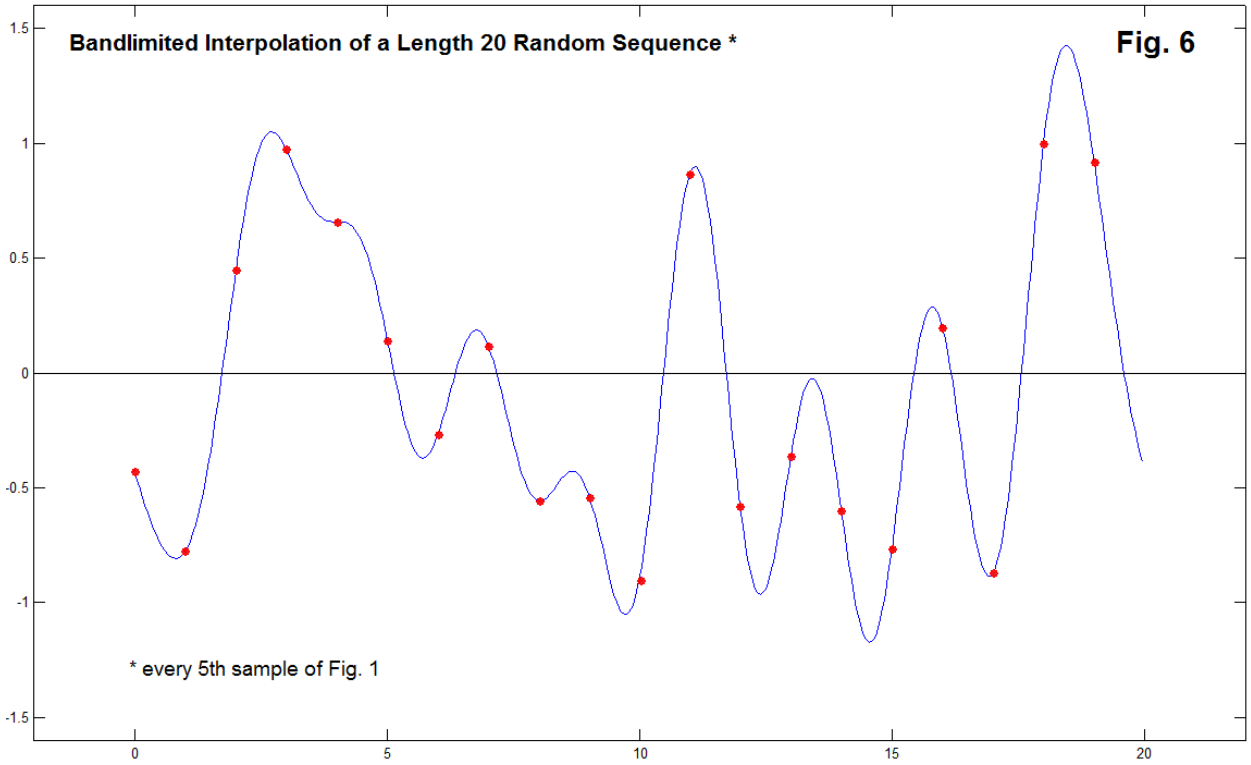

Bandlimited Interpolation of a Length 20 Random Sequence *
Fig. 6
* every 5th sample of Fig. 1
1.5
1
0.5
0
-0.5
-1
-1.5
0
5
10
15
20

Fig. 5 is still a bit cluttered. Consider instead the length-20 random sequence, the circles shown in Fig. 6. These are not new samples, but just the first and every fifth sample from Fig. 1, These we interpolated with the FFT method by 20:1. In this case, we see how remarkably smooth a bandlimited waveform looks, although it came from random samples. Note that a rough structural similarity to Fig. 5 remains here.

Speaking of two sides of a coin, we can compare Fig. 1 with Fig. 5, and perhaps suppose that they are very different. One (Fig. 1) is supposed to be random, while the other (Fig. 5), being itself an inverse FFT, is composed of sinusoidal components. In fact, the two are very much alike. For example, we can play them using Matlab's ***sound*** and they sound basically the same (a short "burp"). So, on the one side of the coin, we are less impressed with the randomness of a so-called white noise sequence, and on the other side, a signal composed of (short) sinusoidal components may be more random than we suppose.

But let's be clear about this. The 20 samples (red circles) in Fig. 6 are random and independent. They were in fact obtained as every fifth sample of Fig. 1. [Equally well, we could have obtained a test sequence by using 2*(rand(1,20)-0.5)]. Here we preferred to maintain the connection with Fig. 1. If on the other hand we had taken the FFT data of the sequence, and pretended that we just made that up as the sum of 11 sinusoidal components (counting dc), and sampled it, we would have the same 20 samples. What we actually see as most interesting in Fig. 6, is probably the blue curve which we see as smooth (bandlimited) – and it seems to be continuous. It's not of course. We interpolated the random samples 20:1 and then plotted the 400 points as connected. The interpolated points are of course not random, as they are highly correlated with each other, and with the original samples.

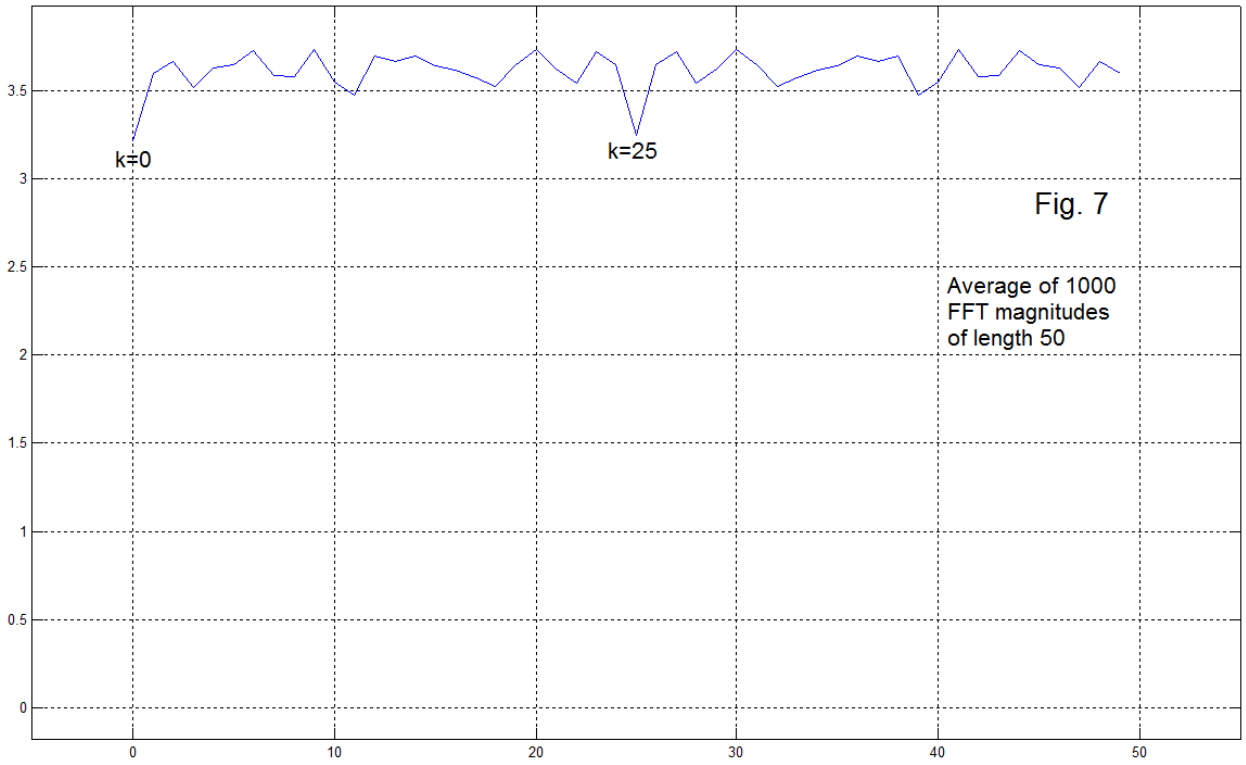
An additional point is that it is not possible to sample a white noise signal and obey the sampling theorem restriction. We understand a white noise spectrum to be flat from 0 (dc) to infinity. Accordingly, it can't possibly be bandlimited, so any sampling frequency is not high enough. The sampled spectrum would be folded over and over, back and forth, forever. True enough, it would remain flat in the added overlaps. Any non-white spectrum with a systematic roll-off could likewise average much flatter than the original.

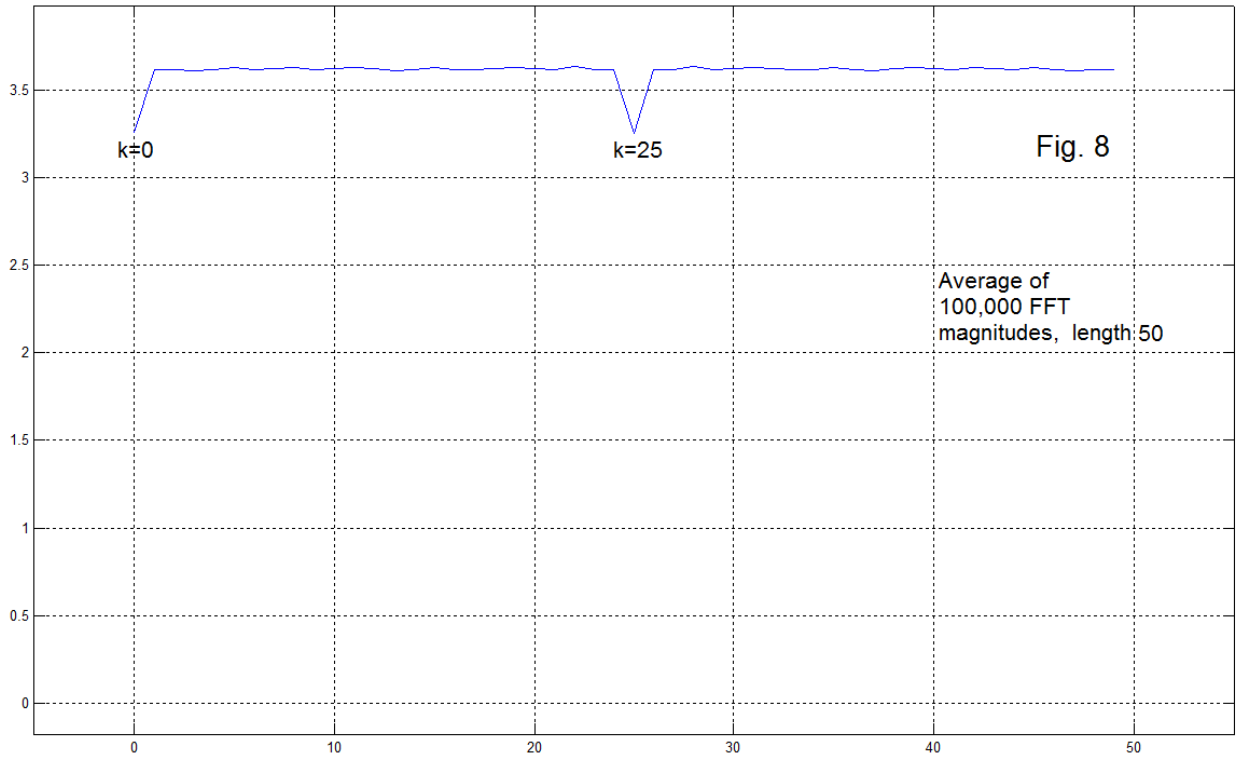
IS THE SPECTRUM FLAT?

As noted in a recent Electronotes issue:

http://electronotes.netfirms.com/EN208.pdf

what would seem to be an obvious procedure to demonstrate the "flat spectrum" of a random sequence contained a surprise. The original idea was that, first of all, we do not expect any single FFT (magnitude) of a white noise sequence to be flat. We would however expect an average of magnitudes of many FFT's to be flatter and flatter as more examples are dawn from the ensemble. Our finding was that the spectrum average, in generals, becomes flatter. Unexpectedly however, there were two frequencies that remain lower than the plateau achieved by the others. These are the frequency 0 (k=0, or dc) and half the sampling frequency (k=N/2 which is only present as a DFT frequency when N is even. Fig. 7 shows the average of 1000 spectra while Fig. 8 shows the average of 100,000 spectra. The dips below the plateau comes out to $2^{3/2}/\pi$ or about 90%. The referenced EN#208 presents this idea and offers explanations. More information will appear in the App Note to follow this one.





TREND LINES ?

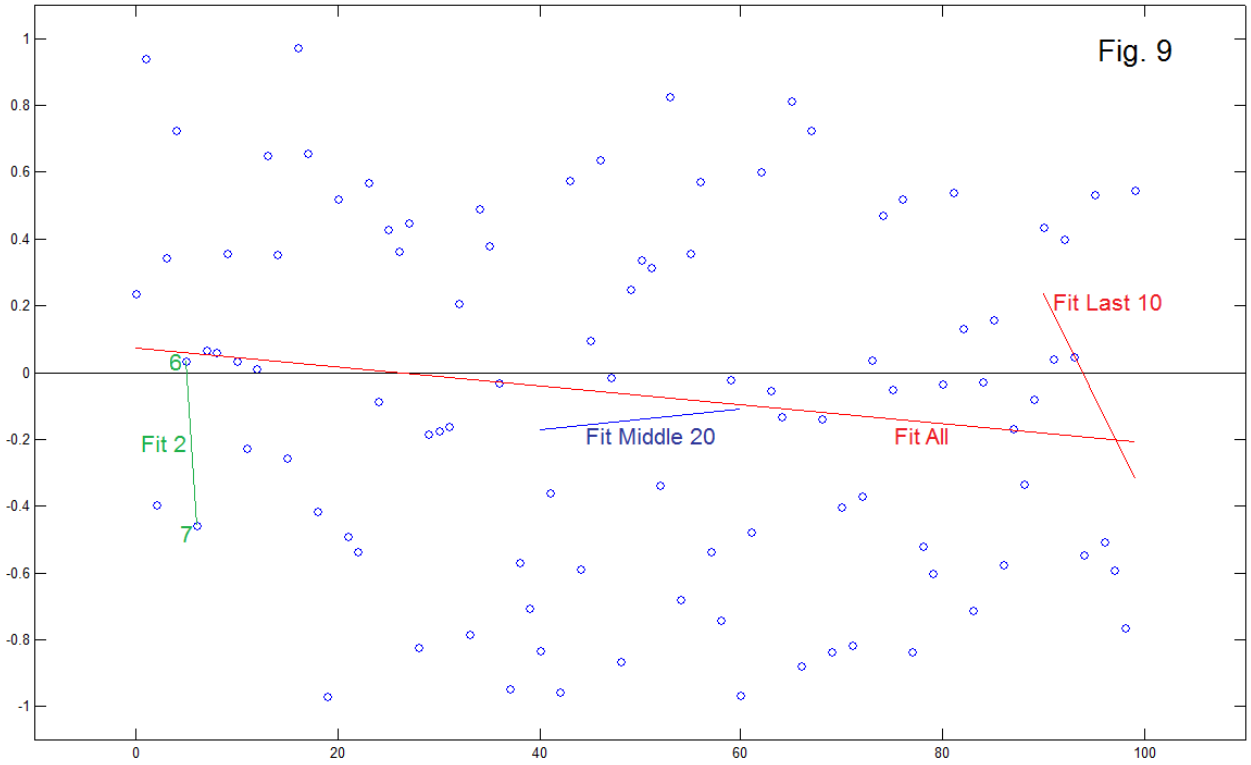
We can take two views of a random noise. The first is that we want a random signal for some purpose. Perhaps we want it as raw sound material for generating a snare drum sound for music synthesis. Perhaps we want it as a pure chance element for a computer game. Perhaps we want to use it to make many many random trials to estimate a probability we can't easily calculate directly. Perhaps we are processing unknown sequences and want to process random signals as a baseline test. We want random numbers – a valuable product.

The other view is that we don't actually want a random signal, but that a random signal has intruded in our real world. For example we have measurement errors, and choose to treat the errors as random (there are many kinds of measurement errors however). In such a case, the errors are thought of as "noise", and we seek ways of removing the noise from the "signal". It should go without saying that if there is actually no signal, a proper noise removal procedure should yield zero. Types of noise removal include averaging (including moving averages, for examples), which is filtering; and curve fitting (such as least squares), which is more a modeling of the signal. These do not always yield clear results.

So there might be something “real” that is going on. It might be stock market plots, temperature plots, weekly sales plots, or something like that. Let’s assume the data sequence has an underlying (although not obviously visible – like visible “by eye”) linear trend that is well-hidden by an apparently random noise. Can we dig this signal out? Perhaps we decide to try a “linear regression” or first-order least squares fit. The question is: does such a result mean anything? Quite possibly not.

It is often wise to test any ideas by using simple test cases. If we propose an experimental test that a dropped ping-pong ball fall to the ground, perhaps we start by dropping ball bearings to see if gravity is working today. If we have a signal processing tool that is supposed to give us a spectrum, we might ask what happens if we test it with a sinewave signal. If we are looking for a linear trend, we might well start with a signal that we know has a particular trend (a straight line with a non-zero slope) or one which we suppose has no trend (thinking of a random signal).

We will not be overly surprised to find that a random signal has a linear trend. Clearly a length-two random signal has a linear trend: the straight line that connects the two points. If we have many points in our random signal, in general the least square



best fit would NOT go through any one of the points, let alone all of them. How badly could we be misled? One answer might be found by investigating how much a trend line varied for different starting and stopping points and/or among different random examples. If we find trend lines having slopes that are themselves random and equally likely to be positive or negative, we might well conclude that random signals possess trend lines, but they don't mean anything. If we found groupings of similar trend lines, we might conclude something is wrong with our supposed random signals.

Fig. 9 shows a length 100 random signal from 2*(rand(1,100) – 0.5) along with four choices of trend line. Keep in mind that any trend lines we find are real, but spurious. In Fig. 9 we see the long red line, which corresponds to the full 100 points. So the 100 points taken as a whole do have a trend (a negative slope in this case). Another interesting case is the short green line, the trend between samples 6 and 7. In this case, since there are only two points, the squared error can be made to be zero by going exactly through the two successive points. We can envision similar high slope trend lines between all successive points, with both positive and negative slopes. In general, a longer sequence will have shallower slopes. Also in Fig. 9 we show a length 20 sequence in the middle (blue) and a length 10 sequence (short red line at end). Clearly any trend line that does not involve a full sequence, but only those near the end, is suspect. We already know that the slopes will be (in general) greater for shorter segments, and the sign obtained often changes with the choice of starting points, not just the ending point.

The fitting of trend lines to noisy data is often used predictively. As such, we need to make use of the most recently available data. In addition, we use older data, but how do we decide how far back to go? This is a most difficult question, unless we want to manipulate a result, in which case, a starting point could be obtained that gives a result favorable to our preferred prediction! A choice that is not so far back as to include what might not be reliable data, but far enough back that trend lines may be relatively stable, would be a happy circumstance. Fig. 10 shows the same random sequence as in Fig. 9, but now shows trend lines starting at samples 0, 5, 10, 15, 20, 25, 30, 35, 40, and 45. Fig. 11 shows just the trend lines, and Fig. 12 plots the slopes. This is a useful demonstration example, but is in no sense typical of anything. It would be easy to have different runs with all upward or all downward trends, not the mixture of trends shown here. The lesson here is that there can be a great deal of variability. Here we are talking about test signals known to not intentionally include any trends.

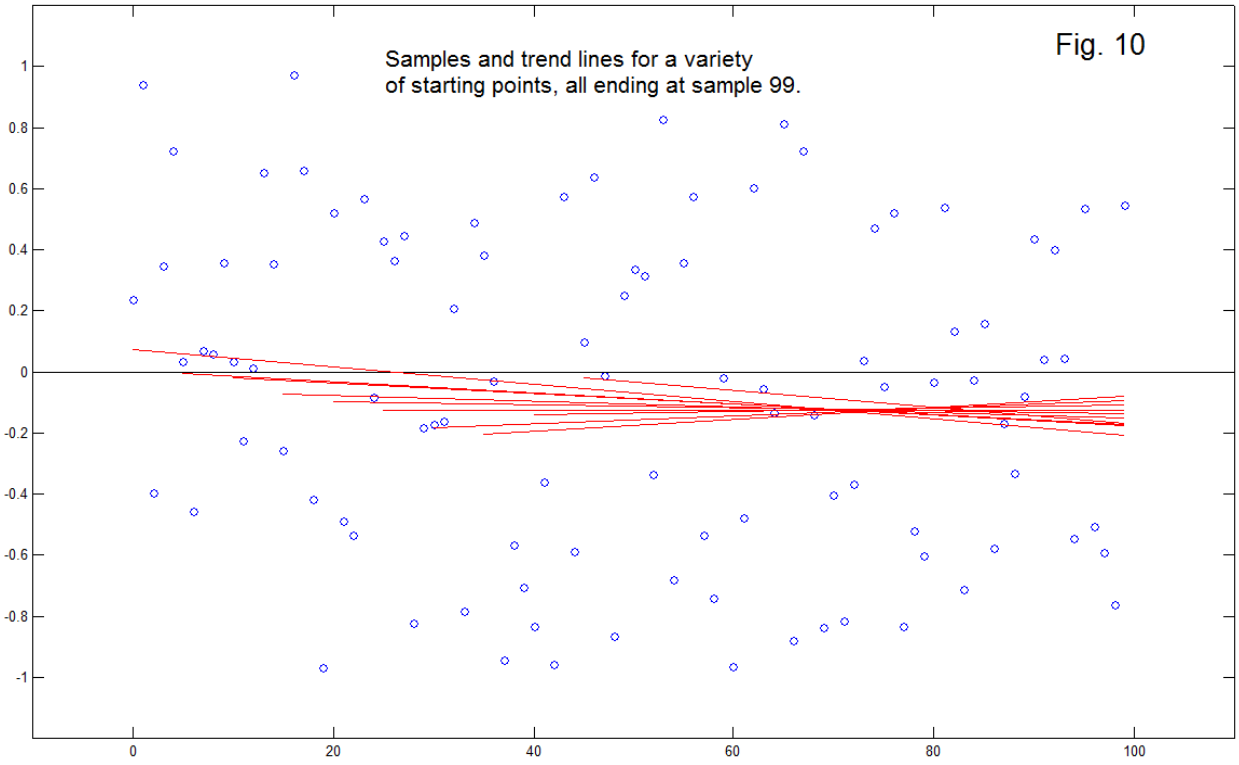


Fig. 10
Samples and trend lines for a variety
of starting points, all ending at sample 99.
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
20
40
60
80
100

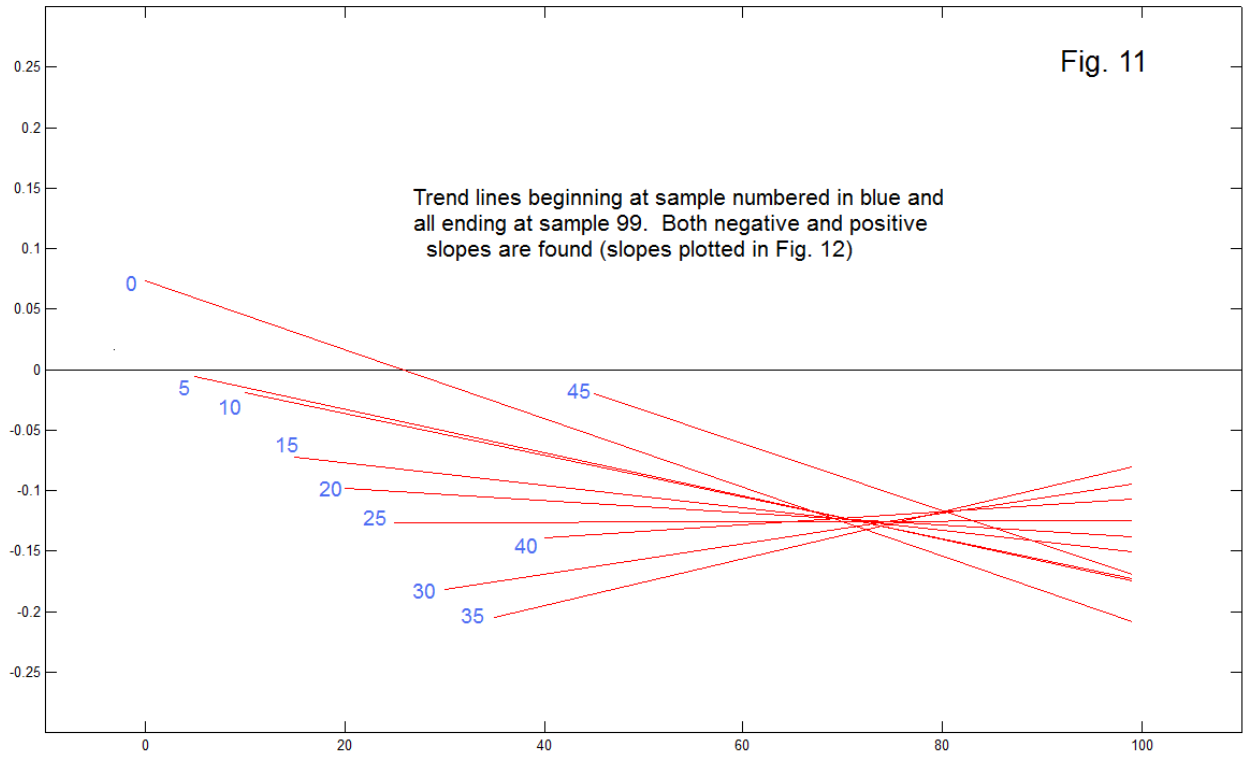
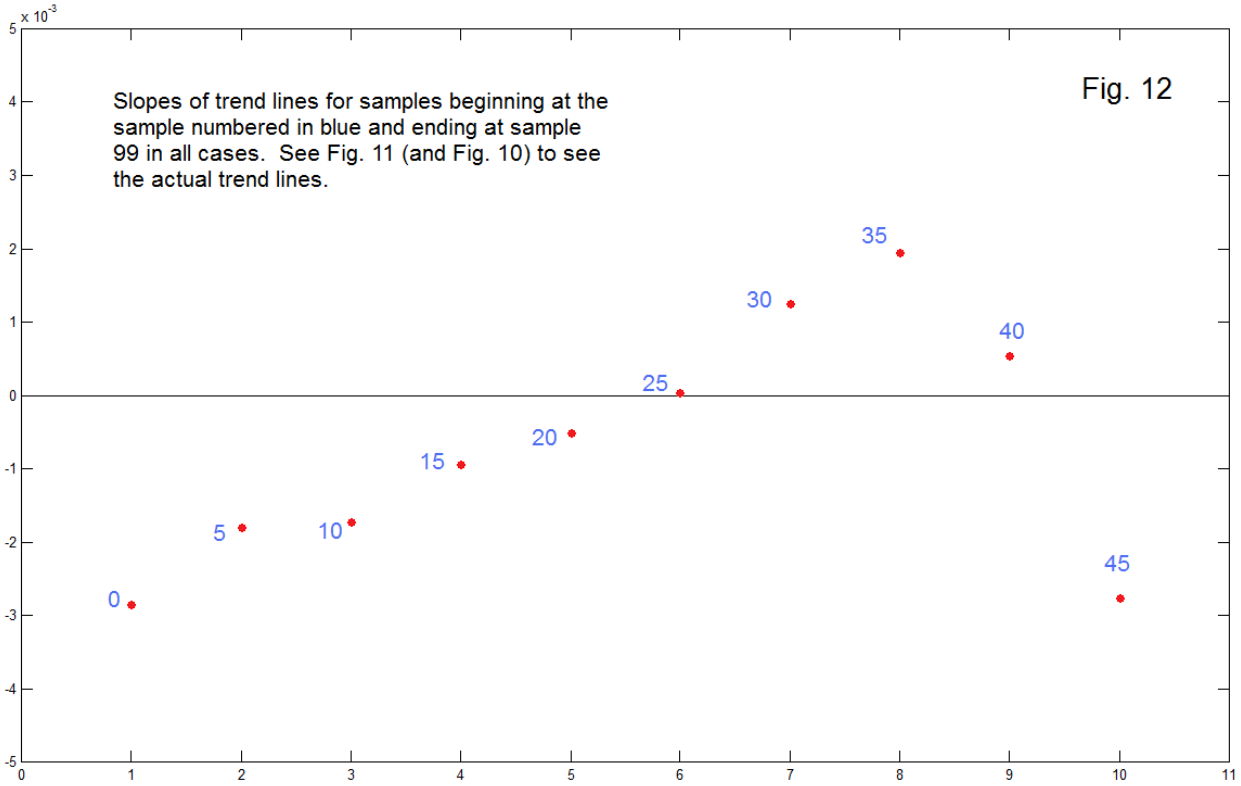


Fig. 11
Trend lines beginning at sample numbered in blue and
all ending at sample 99. Both negative and positive
slopes are found (slopes plotted in Fig. 12)
0
5
10
15
20
25
30
35
40
45
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
20
40
60
80
100



On the flip side, if we are trying to extract a linear model for noisy, real data, we might be quite happy if we already "see" the trend in the plotted data, and if we consistently calculate roughly the same slope for various lengths and endpoints. Otherwise – otherwise.

To summarize, we can and should test our model fitting (trend lines) to see how much of a trend we get accidently with random data. When we know or strongly suspect that trends are real, we can make fits, and using standard statistical methods, make calculations of how much confidence we place in the result.