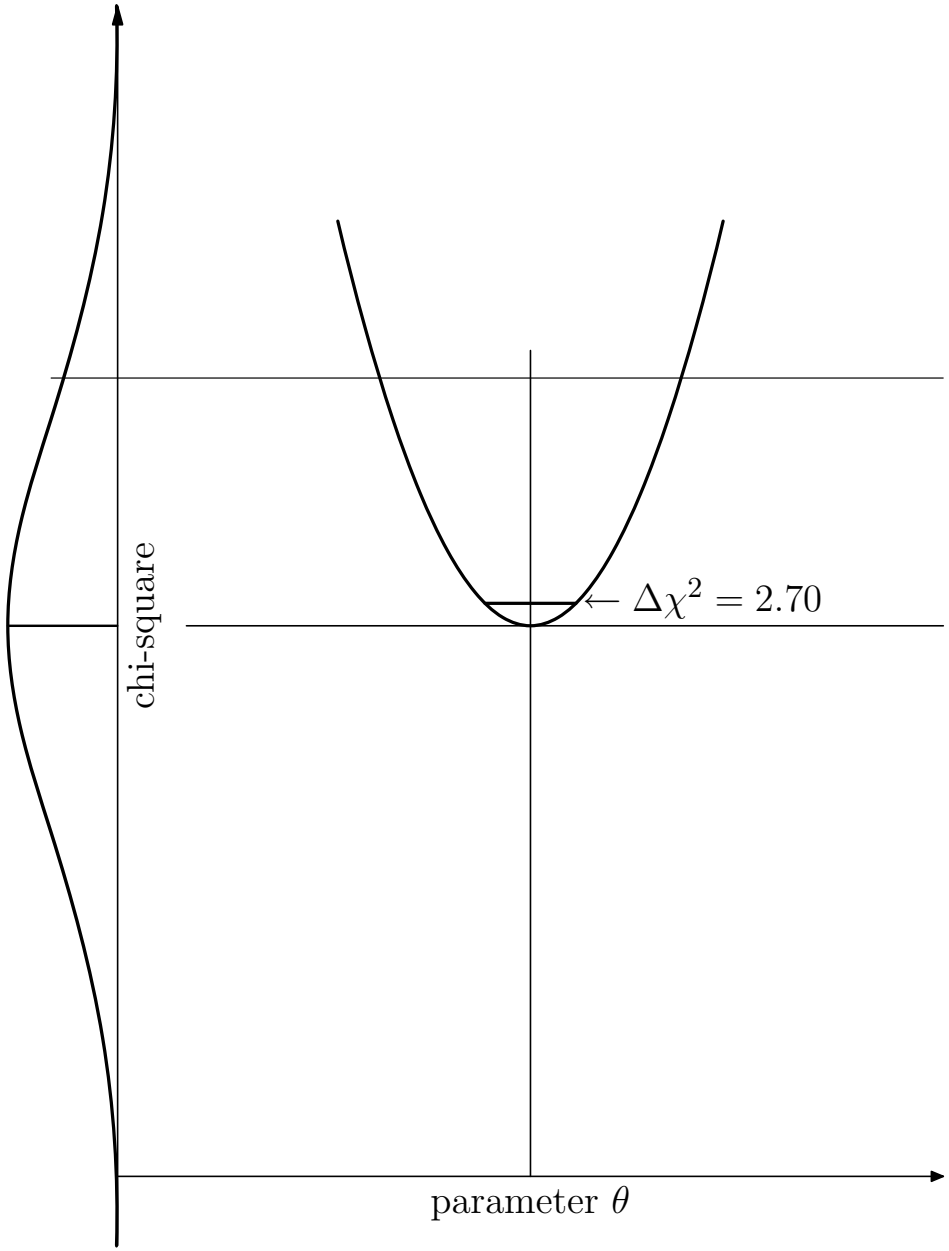
Confidence Intervals and Goodness-of-fit

When we introduced the "Five Classes of Problems", we said that some problems could be approached using the methods of different classes, giving different results.

Here is a good example of this effect, looking at a confidence interval on a fitted parameter as determined by:

1. The method of interval estimation: The ensemble of parameter values that includes the true value with probability 90%.
2. The method of goodness-of-fit: The ensemble of parameter values that gives a good fit to the data, with a P-value greater than 0.10.

In this example, we we use the chi-square method to fit a histogram with 51 bins and one unknown parameter θ.



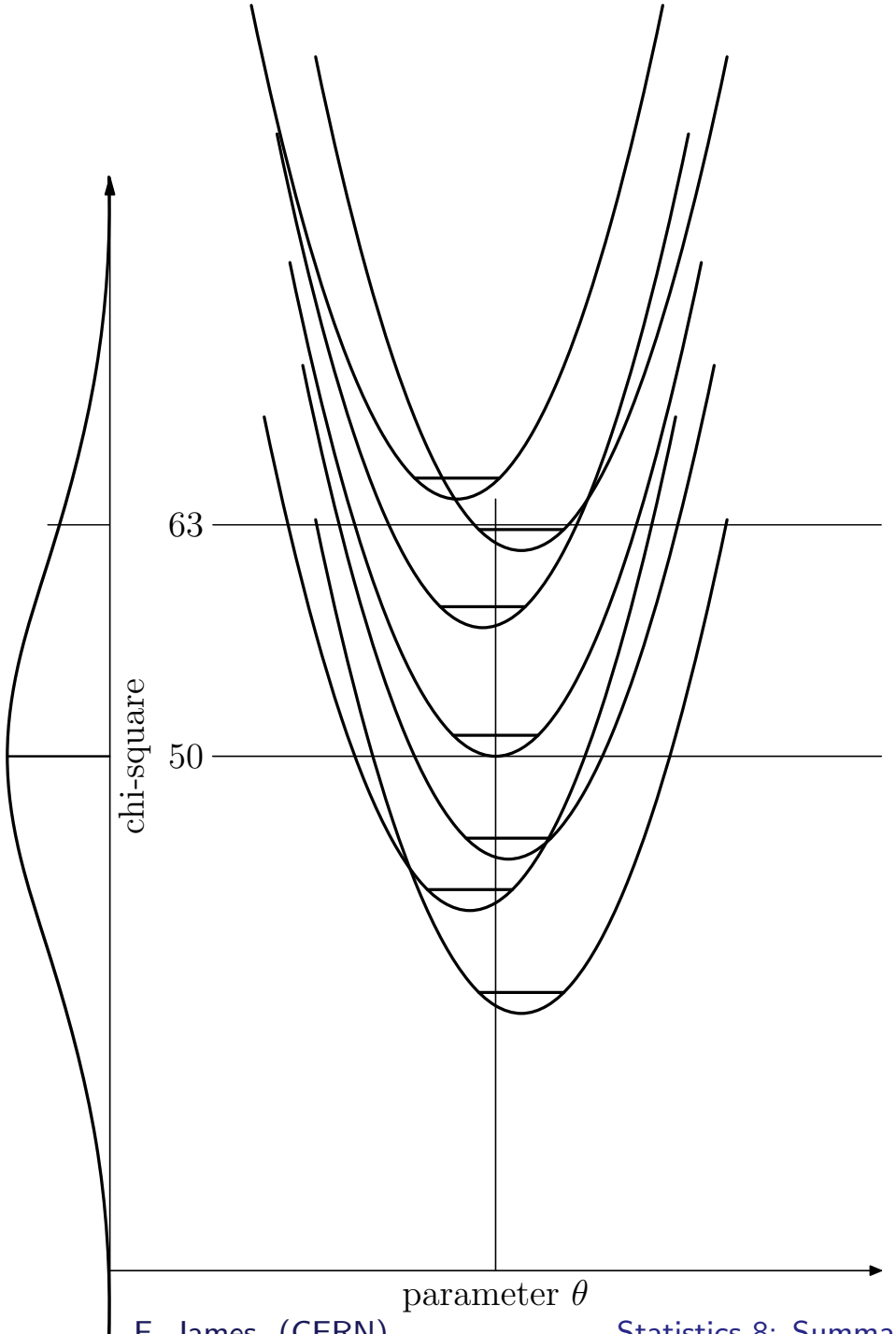
Chi-square as a function of the parameter θ.

At the 90% level, a good fit is defined by $\chi^2 < 63$ because there are 51 bins and one parameter θ.

But the 90% confidence interval for θ is given by $\chi^2 = \chi^2_{min} + 2.70$

How can we understand this?

Repeat the experiment!



Each time we repeat the experiment, we will get a parabola as shown here.

The confidence interval will always be (almost) the same width,
but the GOF intervals will vary wildly, sometimes even be empty.

However both methods will have 90 % coverage.

An experiment can always be repeated

Certainly in particle physics, and probably in most other areas of physics, an experiment can be performed a second time, and often is.

For a particle physicist,
the data from one experiment is a random sample from $P(data|hyp)$,
and if the expt is repeated, we obtain a second random sample
from the same underlying distribution. We will discuss randomness later.

For experiments that are repeatable, $P(data|hyp)$ can obviously be a frequentist probability, defined as a long-term frequency.

A decision cannot be repeated

In making a decision, you get only one chance.
If you can try it many times, to see which choice gives the best results,
it is not a decision, it is an experiment.

In situations that cannot be repeated exactly, it may be impossible to define frequentist probability,
so it makes sense to use Bayesian methods.

Summary of Frequentist vs Bayesian Methods

FREQUENTIST	BOTH	BAYESIAN
Probability is frequency		Probability is degree of belief
	Likelihood Function $P(observed\ data \vert hyp)$	
$P(all\ data \vert hyp)$ needed for confidence belts, coverage, Type I, Type II errors, Goodness-of-fit	enough for m. l. estimates, likelihood-based confidence intervals	Prior $P(hyp)$ needed for posterior density, comparing hypotheses, decision rules

Foundational questions about Frequentist and Bayesian Methods

In frequentist analysis, we assume that:

- Data are random. When are they random?
  When are they not random? How do we know?
  Does it matter?
- Experiments are repeatable Is that really necessary?

Bayesian analysis treats data observed as fixed, because they are known.
and the true value of the hypothesis as random, because it is unknown.
Is this reasonable? Is it acceptable?

The logical conclusion:
Use frequentist statistics for experiments when data are random.
Use Bayesian methods for making decisions.

Example: Medical research and Medical practice.

1. You are a medical researcher.
You are studying the effects of administering Medication X.
You count how many people are treated, how many survive, how many are cured, etc.
You use frequentist statistics, publish P-values, confidence intervals with coverage, etc.

2. Doctor Y reads your publication.
He has to treat Patient Z and decide whether to use Medication X.
He can understand your results because they are objective.
But he should use Bayesian methods, including his prior knowledge about Patient Z, to make his decision about the treatment.

Randomness

1. True randomness

True intrinsic randomness is found in Quantum Mechanics.

Physicists know quantum mechanics, but most statisticians do not,
so they have no reliable model for randomness.
The statistician's expression for random is i.i.d. .
(independent and identically distributed)
In Bayesian statistics, there are no random variables, only unknowns.

Is true randomness found anywhere else in Nature?

We need a good (operational) definition of randomness. [Later ...]

A good definition of randomness could help us to find a computer algorithm that makes random numbers for Monte Carlo calculations.

2. Pseudorandomness.

In the 1940's, John von Neumann and friends were amazed to discover that simple arithmetic operations like integer addition or multiplication could produce sequences of numbers that appeared to be random.

They called them pseudorandom.
Nobody understood why they were (almost) random.

Fifty years later, this was still the case!

see, for example:
Donald Knuth, The Art of Computer Programming, Vol. 2, 3rd edition, 1994. Over 600 pages on pseudorandom number generators, but no hint about how they make randomness. A lot about how hard it is to define.

A different approach from the Russian school

The study of classical mechanical systems developed differently in the West (only interested in solvable systems) and in the Soviet Union (interested mainly in non-integrable systems because they are candidates for chaos.).

Mathematicians and physicists of the Russian school (Kolmogorov, Anasov, Rokhlin, Avez, Sinai, Arnold, and others) eventually developed a very elegant and general theory to describe the (asymptotic) behavior of non-integrable classical mechanical systems, including Kolmogorov mixing.

Representing the state of the dynamical system at the i^{th} time step by an array $\mathbf{X}_i$, the time evolution can be represented by transformations

$$\mathbf{X}_{i+1} = \mathcal{A}\,\mathbf{X}_i \mod 1$$

To obtain Kolmogorov k-mixing, the matrix $\mathcal{A}$ has to have

- Determinant $= 1$
- All eigenvalues complex and distinct
- No eigenvalue has modulus 1

Kolmogorov Mixing

Kolmogorov mixing is a hierarchical ordering defined as follows:

1. Zero-mixing is the same as ergodic motion. It means that the system will asymptotically sweep out all the available states.
2. One-mixing means that the coverage will be uniform, in the sense that asymptotically, the probability of finding the system in any given volume of state space is proportional to the volume.
3. Two-mixing means that the probability of the system being in one volume of state space at one time, and another volume at another time, is proportional to the product of the two volumes.
4. N-mixing means that the probability of finding the system in N different volumes at N different times is proportional to the product of the N volumes.
5. k-mixing is N-mixing for arbitrarily large N.

Mixing in Sinai's billiards

Another way of looking at chaos in classical dynamical systems is the trajectory of a ball on a frictionless billiard table, studied extensively by Ya. G. Sinai.

On a standard rectangular table, it can be shown that the motion is almost always ergodic (almost all trajectories will cover the whole table asymptotically), but there is no mixing, so these systems are not k-systems.

If the edges of the table are curved outward, there is a focussing effect, and the motion is not even sure to be ergodic. On an elliptical table, for example, the system is integrable, and not chaotic.

But if the edges of the table are curved inward, they defocus nearby trajectories, leading to Kolmogorov mixing.

Lüscher's algorithm: The Lyapunov Exponent

Around 1993, Martin Lüscher of DESY recognized that an existing RNG by Marsaglia and Zaman called SWB (or RCARRY) was closely related to a k-system. However, it was already known to fail some tests of randomness. But he knew why.

Answer: Because Kolmogorov mixing is only an asymptotic property . Such systems forget their past history, but only after a certain time.

K-systems diverge exponentially, in the following sense:
If a k-system is initiated from two different but arbitrarily close initial states, the two trajectories will diverge exponentially. The exponent describing the speed of divergence is called the Lyapunov exponent.

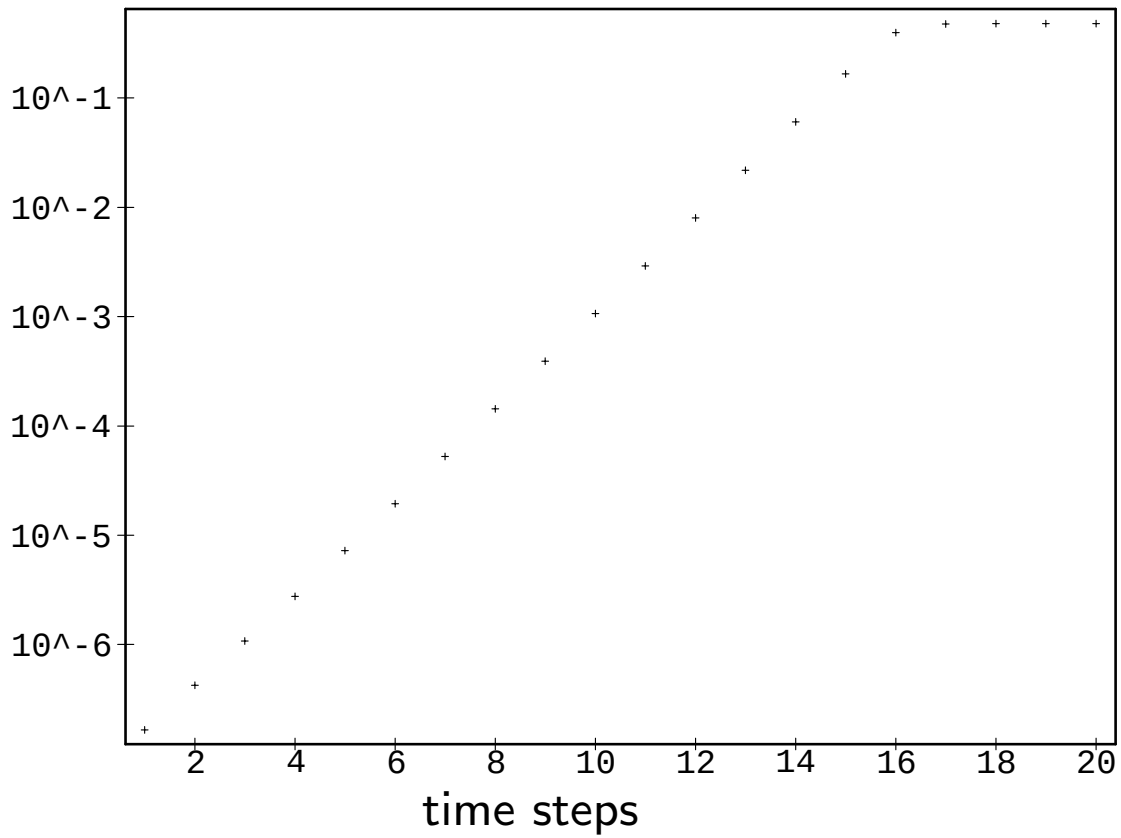
For the matrix $\mathcal{A}$, there are 4 eigenvalues with maximum absolute value $|\lambda|_{max} = 1.04299..$, and the Lyapunov exponent is $\ln |\lambda|_{max} = 1.01027..$

For this system, that means that full mixing (where the deviation between the trajectories exceeds 1/2) is attained only after 16 applications of $\mathcal{A}$.

Lüscher's algorithm: Decimation

This figure shows the "experimental" divergence of nearby trajectories after the number of time steps shown on the x-axis.

Average spacing between trajectories



After generating 24 numbers, you must reject 15×24 numbers before using the next 24, in order to attain complete mixing, or independence.

M. Lüscher, Comp. Phys. Comm. 79 (1994) 100

RANLUX: the Luxury Level

When I coded Lüscher's algorithm in Fortran, I called it RANLUX because it offered several different luxury levels:

1. Level Zero with no decimation,corresponds to RCARRY: fast, and good enough for many applications, but known to have easily detectable defects.
2. Level One with half the numbers rejected, is almost as fast as RCARRY, but with considerably better randomness.
3. Level Two with acceptance 1/4 is about half the speed of RCARRY, and much more random, but still has detectable defects.
4. Level Three with acceptance 1/8 has theoretically detectable defects, but none has yet been detected.
5. The Highest Level, corresponding to rejecting 15 steps, should have no defects detectable by any possible tests. Unfortunately it is about 8 times slower than RCARRY.

The idea was that if you can afford the luxury of a provably good PRNG, you can use RANLUX at the highest luxury level. Otherwise you should use the highest luxury level you can afford.

Continuous and discrete systems

The major theoretical problem with Lüscher's algorithm is that the Kolmogorov k-systems are continuous dynamical systems, but we are applying the theory to discrete systems.

An important difference between discrete and continuous systems is that a continuous system can follow an infinitely long trajectory in state space without ever encountering a previously occupied state.

A discrete system, on the other hand, has only a finite number of possible states, so it eats up its state space as it moves through it, in the sense that all those states it has already visited are removed from the space of states it has available for future visits. Otherwise it gets into a loop.

However it is easily seen that, since the period of RANLUX is $\approx 10^{171}$, even if it was used to generate all the random numbers that have ever been generated plus all those that will ever be generated on the earth, it would not eat up more than 10^{-130} of its total state space.

Progress with MIXMAX

Meanwhile the Greek physicist George Savvidy, working in Armenia, also knew about the Kolmogorov Theory, and tried to make a matrix generator called MIXMAX, based on this theory. Unfortunately, matrix multiplication is very slow, taking $O(N^2)$ operations to generate N numbers. Even when he managed to reduce the time the computation time from $O(N^2)$ to $O(N \ln N)$, it was not fast enough to compete with other RNG's.

More recently, George Savvidy's son Konstantin (Savvidis) has continued this work and has reduced the computation time to $O(N)$, which finally makes it comparable in speed with the fastest generators.

MIXMAX is now available from HEPFORGE and is also installed at CERN. The default matrix size is now 256×256, and the generator produces extended precision reals or 60-bit integers. A lot of work has gone into assuring long periods and good mixing (eigenvalues as far as possible from the unit circle to obtain a big Lyapunov exponent and avoid decimation). Even without decimation it passes the Big Crush tests.

The matrix used in MIXMAX

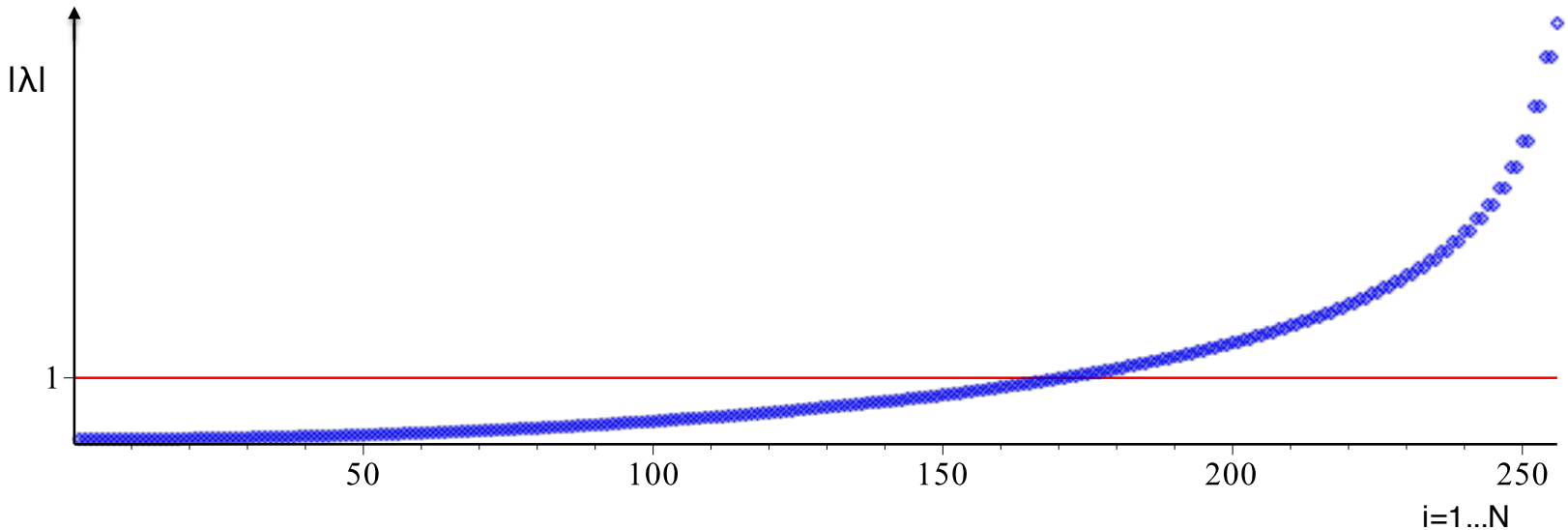
1	1	1	1	1	1	...	1	1
1	2	1	1	1	1	...	1	1
1	3+s	2	1	1	1	...	1	1
1	4	3	2	1	1	...	1	1
1	5	4	3	2	1	...	1	1
	...							
	...							
1	N	N-1	N-2		...		3	2

where the current default value for N is 256.
The integer s is zero for some values of N, but for other values a small non-zero integer is required to avoid an eigenvalue of modulus one.

This matrix is chosen to have eigenvalues as far as possible from one, which is difficult because the determinant (= the product of all the eigenvalues) must be $= 1$.

Eigenvalues and Mixing

The mixing properties of the linear system (in particular, the speed of mixing) depend on
The Lyapunov exponent, which is a function of $|\lambda|_{max}$, and
The Kolmogorov entropy , depends on all the $|\lambda| > 1$.

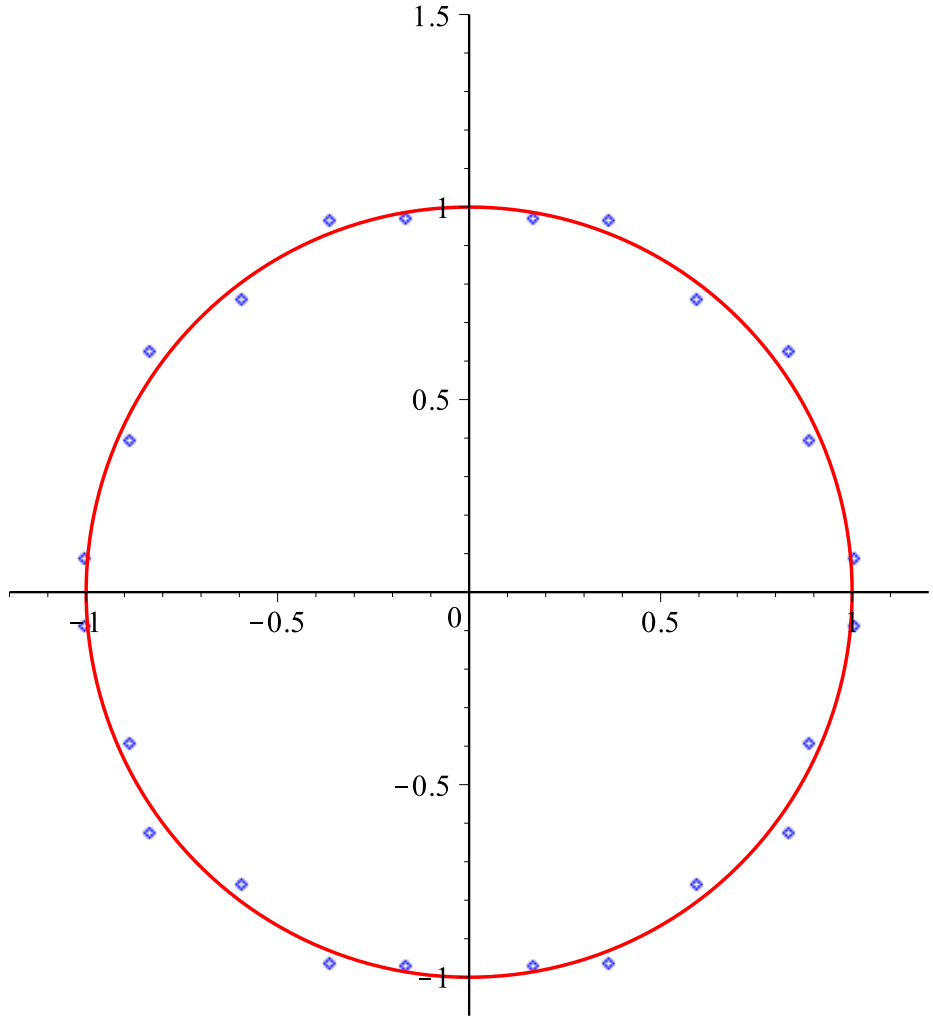


The spectrum of eigenvalues for the MIXMAX matrix with $N = 256$.
The scale of $|\lambda|$ is logarithmic from 0.25 to 256.
There is no eigenvalue with $|\lambda| = 1$ and very few close to one.

Mixing in other RNG's – RCARRY-RANLUX

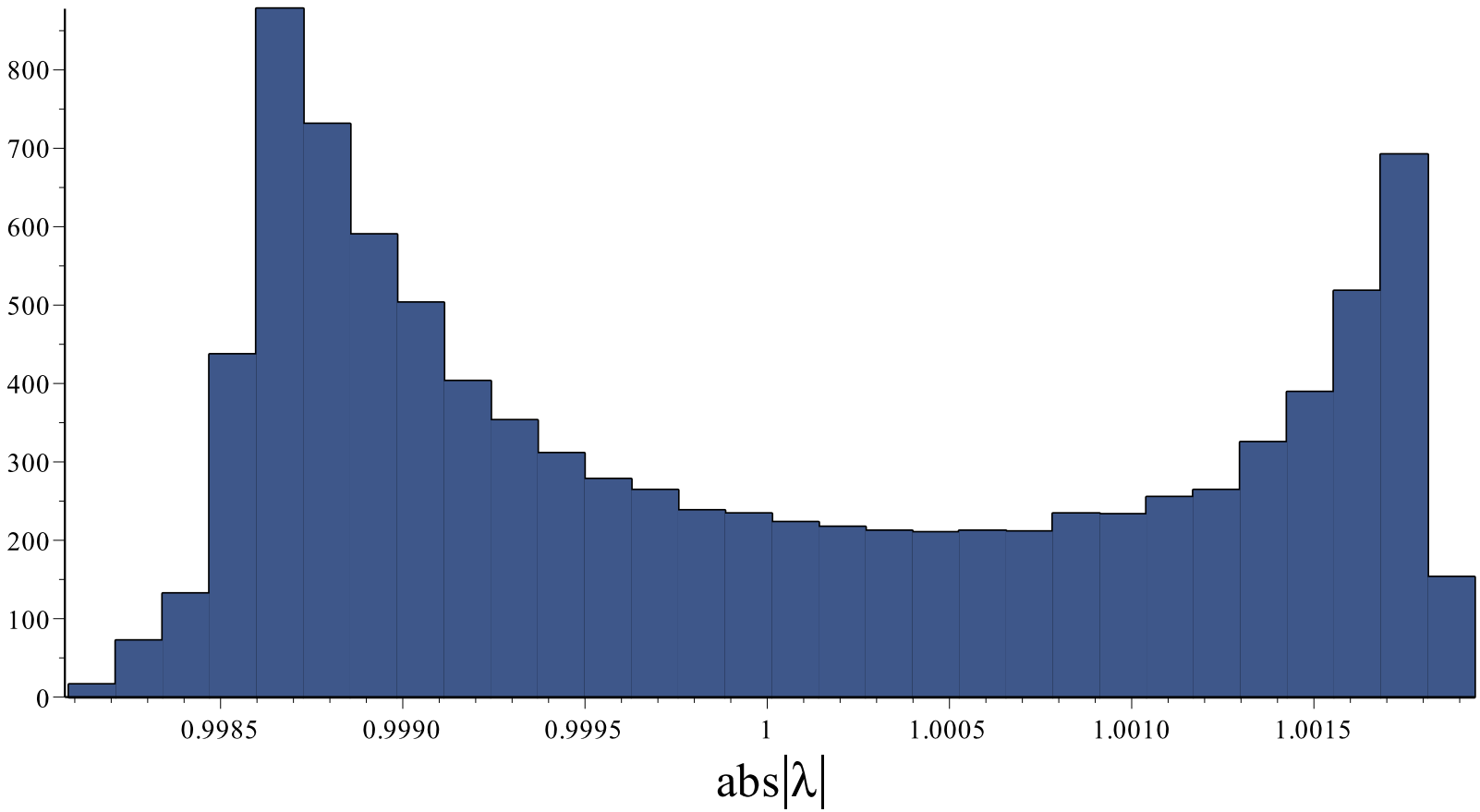
Martin Lüscher has calculated the eigenvalues of the effective matrix of RCARRY (and RANLUX), which are plotted below.

It can be seen that the eigenvalues all lie very close to the unit circle, which makes the Lyapunov exponent only a little bigger than one. This can be seen as the reason why it is necessary to reject so many vectors before RANLUX attains asymptotic mixing.



Mixing in other RNG's – The Mersenne Twister

Konstantin Savvidis has calculated the eigenvalues of the effective matrix corresponding to the Mersenne Twister. There are an enormous number of eigenvalues, but they are all very close to the unit circle, which helps to explain why MT has been observed to fail many tests.



Testing Random Number Generators

Since there has been (until recently) no theory of randomness that could be applied to RNG, the only way to have some evidence for randomness was by statistical testing of the results.

A test of randomness is any function of the output of a RNG for which the expectation and variance are known (under the hypothesis of complete randomness of course). There are an uncountably infinite number of such tests possible. No test can prove a RNG to be good, but if a test is failed, that is proof that it is not good. Since these are Goodness-of-Fit tests, the power of the test is undefined, and no test is necessarily better than any other, but with experience, we can have some idea.

The DIEHARD test suite of Marsaglia was used for many years, but it is not extendable for testing long-period generators. The current standard is TestU01 by L'Ecuyer and Simard of the University of Montreal. The heart if this excellent software is the ensemble of tests affectionately known as Big Crush.

Coverage in Practice

Frequentist coverage is mainly a mathematical concept, in the sense that, if we use the Neyman procedure, we are mathematically guaranteed to have exact coverage (or overcoverage in the case of discrete data).

But of course it should also be observable in the real world. The problem is that we don't know the true value when we do the experiment, so we don't know how many experimental error bars actually cover the true value.

But if we wait a little while, the true value may become known, or at least known much better than when the experiments were performed.

So let us consider coverage from the practical side.

How does coverage work?

Consider some set of 68 % Confidence Intervals from different experiments.

For example look at the figure in the Introduction to RPP (the same figure has appeared for several editions). For our purposes, it doesn't matter what quantity is being measured here, but it happens to be $Re(X)$, where X is a constant parameterizing the violation of the $\Delta S/\Delta Q$ Rule in leptonic K^0 decays.

Because physics has made some progress in 30 years,
we now know the true value: $X \approx 0$.

In the figure, there are 17 Confidence Intervals of 68% CL,
and 12 of them include the true value (zero).

Coverage works well here.

Physicists generally consider coverage a required property of confidence intervals. (See R. D, Cousins, Am. J. Phys. **63**, 5, May 1995)

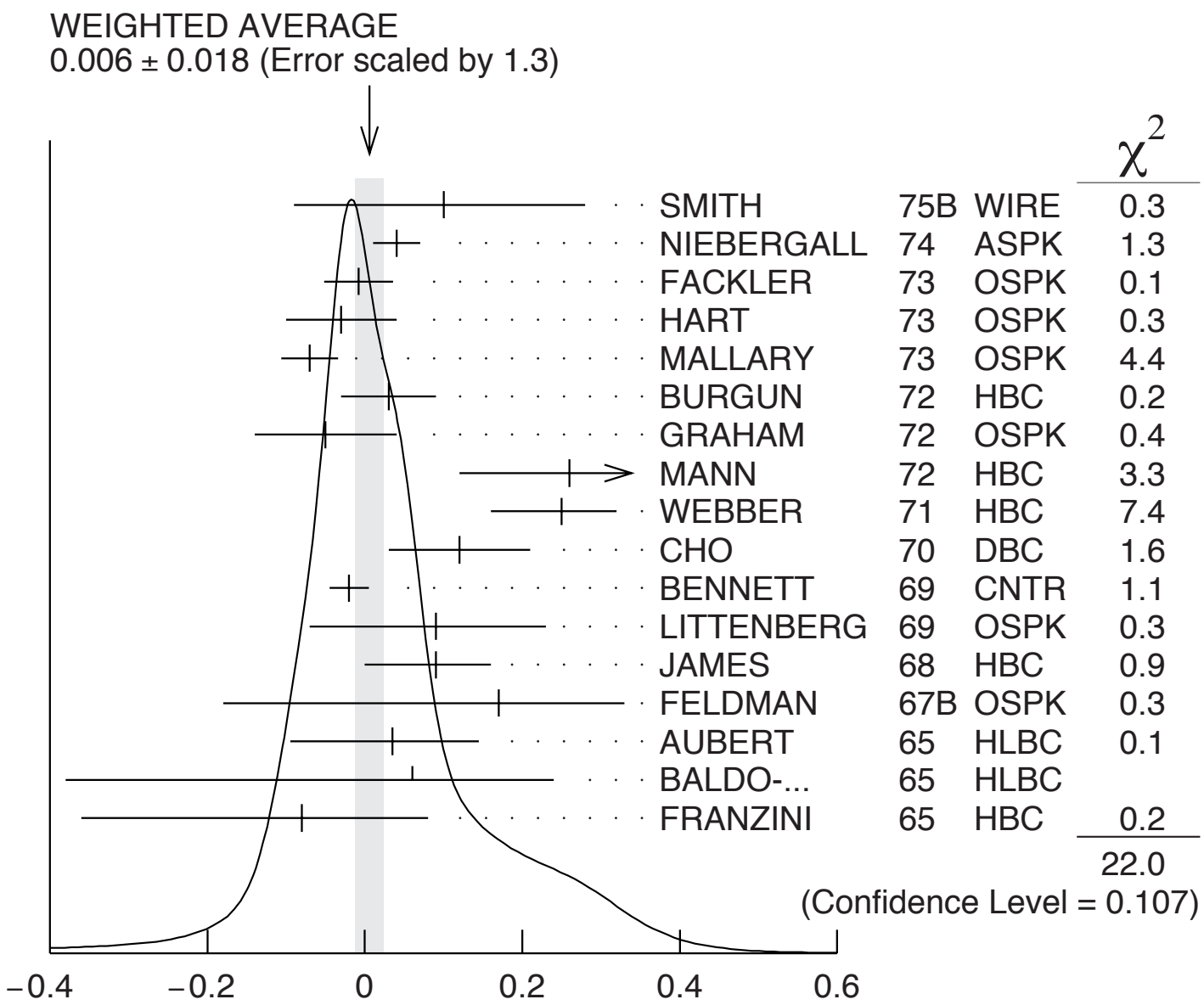


Figure 1: A typical ideogram. The arrow at the top shows the position of the weighted average, while the width of the shaded pattern shows the error in the average after scaling by the factor S. The column on the right gives the χ^2 contribution of each of the experiments. Note that the next-to-last experiment, denoted by the incomplete error flag (|), is not used in the calculation of S (see

Effects which can interfere with coverage

In practice, there are several effects that can make the apparent coverage wrong:

1. The file drawer effect. If several expts look for an unexpected new result, the ones that don't observe any effect will not publish their results, so we expect an overabundance of significant P-values in published papers.
2. Flip-flopping and other mistakes can cause physicists to publish incorrect confidence intervals that do not cover.
3. Embarrassing results such as **signal greater than physically allowed or predicted by any theory**, would most likely be "massaged" before publication, even if they only resulted from an unusual fluctuation. Suppressing such results modifies the global coverage.
4. The stopping rule for corrections. Most experimental results require several corrections (for systematic errors, calibration, etc) before they can be published. There is a tendency to stop applying corrections as soon as one attains the expected result. [W mass at LEP]

You have measured $M_{\text{top}} = 175 \pm 5$ GeV.

What statement can you make about the 68 % confidence interval (170,180) ?

1. The method I have used produces confidence intervals, 68 % of which include the true value of M_{top}.
2. The probability that M_{top} will lie inside my confidence interval is 0.68.
3. The probability that M_{top} lies inside the interval (170,180) is 0.68.
4. The probability that the interval (170,180) includes M_{top} is 0.68.
5. The probability that M_{top} lies inside the interval (170,180) is 0.68, in the sense of Neyman.

Note that if you make any of the above statements to a journalist, he will report No. 3 in his newspaper.

One last detail

Is there a difference between:

1. The probability that the true value lies inside an interval, and
2. The probability that an interval covers the true value ?

Formally, NO, since $P(A > B) = P(B < A)$.

but I think most people prefer to say $P(X > 5)$ rather than $P(5 < X)$.