General idea - to look at the rate of change of K compared with N to better distinguish chaos in borderline cases.

Old method was simply to look at values of K and to deduce that if large enough, probably chaos, and if small enough, probably not.

Weakness lies in the fact that it is unable to distinguish between cases which have a small value of K for small N but will have a large value of K for large N (or vice versa). For example, in values of $\mu$ slightly larger than $K_{crit}$, when N = 5000 values of K are typically around 0.01 in all cases (whether or not the system is chaotic), but if we increase N then some K values will increase towards 1 (indicating chaotic dynamics), whereas in other cases the value of K will drop towards 0. In theory, using large enough values of N should work in all cases, but: a) In many cases a very large value of $N$ is required, often over 1 000 000. b) We have no way of knowing whether the value of N is large enough. Earlier papers suggested comparing to earlier results by hand, but ideally this process could be made more rigorous and done automatically. c) Computational time increases proportional t $N^2$, so we wish to avoid such large values of N.

A well constructed test would involve only going to larger values of $N$ when there is a reasonable level of doubt, saving time and effort on the more clear cut cases.

But to determine 'clear cut', we suggest always looking at $K$ values from at least two values of $N$, to ensure no unexpected trends (eg N = 5 000, K = 0.001 to N = 10 000, K = 0.09 may both be small values of K (¡0.01), but might be deemed to be suspicious and merit further investigation).

Proposed general test:
For any value of $\mu$,
a) Calculate $K$ values for two values of $N$
b) Calculate relevant statistic(s) to determine if deterministic, chaotic or unresolved.
c) If unresolved, try new (larger) values of $N$ and repeat until result obtained. Possibly calculate statistics comparing more than just two data sets at a time.
d) If no certain result even for very large values of $N$, label as inconclusive and possibly resort to other methods.

It is proposed that we ensure the difference between $N$ values are prime to avoid problems with periodicity.

Suggested statistic: Pairwise comparison of two data sets, Median/Median Absolute Deviation (MAD).

Weaknesses: Statistic not ideally suited to convergent nature of process - example if $K$ values evenly distributed over [0 0.8] at N = 5 000 were to all to converge to 0.95 at N = 10 000, would seem a most significant result. However in this case because of nature of test (pairwise absolute (as opposed to relative) comparison) would return a weak result. Nevertheless, would return a positive result in most cases.

In later cases should statistic be computed between all difference pairs of $N$? How would you change the requirements, all pairs pass or at least one passes.

Suggested requirements:
If $K$ large enough and at least one rate of change negligible (in case of 0.999,

0.998 0.997801 for N = 5 000, 10 000 and 25 000 say) then implies chaos. If small enough and one rate of change negligible, deterministic.
Otherwise if $K$ not very large or very small but (one? all?) rate of growth significant, then implies trend and thus chaos or deterministic.
Inconclusive otherwise
Hence inconclusive if very large (small) and not possibly decreasing (increasing)
Not very large or small and not very much increasing (decreasing).

Proposed values:
Definitely increasing if all stat values > 5, at least one > 10
Possibly increasing if all stat values > 1

To be clarified:
Effect of noise
Efficiency
True distribution of median (if take new sample c values) and relationship with MAD, tail populations. Ensure that corresponds with results if large values of $N$ chosen, what if change N values in test (i.e. not 5000vs10000 etc).
Check results if take new set of $c$ values.

Example of results

$\mu \in [3.55 : 3.999]$, 450 increments. Edge of chaos defined to be $\mu \in [3.56995 : 3.6]$, all done with matlab seed = 3

Only 0-1 Test:
At $N = 10$K, 64 have $K < 0.01$, 360 have $K > 0.99$, 26 undecided (19 in edge of chaos)
At $N = 25$K, 4 extra $K < 0.01$, 4 extra $K > 0.99$, 18 undecided (14 in edge of chaos)
At $N = 50$K, 0 extra $K < 0.01$, 7 extra $K > 0.99$, 11 undecided (7 in edge of chaos)
At $N = 100$K, 0 extra $K < 0.01$, 3 extra $K > 0.99$, 8 undecided (4 in edge of chaos)
All undecided clearly trending up, but some values as low as 0.2 and so will take $N = 700$K to resolve(estimate), or 20 hours on university computers (compared 20 mins for 100K).
This gives total of 384 chaotic values, 1 off Jacobson result. Suggest error occurred at $\mu = 3.57$, merely 0.00005 after $\mu_{crit}$.

Including some gradient test
Interesting has grad ¿ 3, conclusive has grad ¿ 10:
Comparing 5K v 10K
64 have $K < 0.01$, all grad less than 3 (indeed all less than -5)
354 have $K > 0.99$ with grad greater than -3 (the 6 other grads between -3 and

-6)
14 have $0.01 < K < 0.99$, grad greater than 10
1 has $0.01 < K < 0.99$, grad less than -10
17 undecided (including 6 doubtful ones which were 'decided' under above test)

Using 25K
The 6 from earlier $K > 0.99$ with grads less than -3 now have at least one grad (either 5 vs 25 or 10 vs 25) > -3 (indeed all > -0.1), $K$ still > 0.99 - implies chaos.
3 extra with $K < 0.01$, grad less than 3 (all less than -8)
1 extra with $K > 0.99$, grad > -3 (grad = 10.4)
2 extra with $0.01 < K < 0.99$, grad greater than 10
5 undecided

Using 50K 1 extra with $0.01 < K < 0.99$, grad greater than 10
4 undecided

Using 100K 1 extra with $0.01 < K < 0.99$, grad greater than 10
3 undecided (with $K$ values of 0.195, 0.644, 0.895 and grads of 7.5, 8.1, 9.5)

If we were to relax conclusive condition to 6 (still over 3 sigma out, bearing in mind that to get false positive with this data set would need conclusive condition of -6 - ridiculous), we would have 8 undecided at 5v10k (including 6 doubtful ones...), 1 at 25k and none at 50k.

All 0-1 tests seem to return same result eventually.

Computationally more efficient to use grad method (bearing in mind 100k takes 400 times longer than 5k, so even though grad method requires 450 extra 5k runs, better as it saves at least 4 100k runs (ignoring 50k runs saved etc)). And return results sooner (with no known disagreement - could even drop both possible and definitive stat values to less than 1 and get same results [not that I am suggesting anything this foolish, just test seems very robust]).

Examination of $\mu = 3.57$:
For 5K, 10K, 25K, 50K, 100K
$K = 0.015, 0.01, 0.006, 0.006, 0.01$
So good chance there is an upward turn, but very slow and weak
Could detect if we increased 5k to 25k, but would take a lot longer (and would have to increase even further for values between 3.5669 and 3.57). Present method with gradient returns result for values of $\mu$ beyond 3.5701 if start at 10k instead of 5k, and from 3.5704 with 5k.
Alternatively could change 0.01/0.99 to 0.005/0.995. All values tested do eventually seem to make their way into these areas, but would also take longer. Will run test to confirm - might just lead to more 'unresolved' cases. In this case

would suggest only looking at most recent stat, as first ones could be negative in dip (but hopefully not large negative) before turning around and going positive.

Alternatively depending on initial value $N$ could just say test works for chaotic and deterministic of period less than $N/5$ (this value estimated, not tested). Always going to be undetected cycle of one billion hiding amongst chaotic states for values of $N$ less than one (or two) billion.

Tests to find density of chaotic states (results don't depend on method used - in all cases 5K value for $N$ is either above 0.99 or below 0.02, and does what expected (either rising or dropping) )

Test 1: Run over a large number of $\mu$s, and look at average for estimator. Improvements: Take random $\mu$s instead of equidistant ones so that statistic is more random, arguably now each $\mu$ value has the same probability of being chaotic (and that probability is the same as the percentage of parameter values which give chaos). Note that this is an unbiased estimator
This leads to a binomial distribution, added advantage in that it gives an estimator to the variance so we can put error bars on expected proportion. For this we use a Binomial proportion confidence interval, the standard way is to treat $\hat{p}$ as if normal with variance $\hat{p}(1-\hat{p})/n$. Apparently valid so long as $N$ is larger than 50 and $n * \hat{p}$ is larger than 5 (so 1000 definitely acceptable, larger will narrow error bars)

Test 2: Run over increasing number of $\mu$s (spaced equidistant over interval), look for convergence
Main weakness: Not automated when 'convergence' is achieved. Useful for examining patterns and check validity of error bars.

Two plots.
3p8to3p81convergence.png shows convergence over increasing $N$. Error bars are drawn, with $\hat{p}$ value used being the $\hat{p}$ value from the last (1000) simulation.
3p8to3p81hist.png shows a histogram for number of $\mu$ values which lead to chaos with 1000 random $\mu$ values chosen for each one. We expect these to be binomially distributed - 10 not enough to see full pattern (try 100, will take two weeks). Together (10 000 $\mu$ values sampled) these simulations give an average of 3.29% chaotic, with $\sigma$ of 0.18%, which agrees with convergence results. Note that for $N = 1000$, i.e. for each individual trial, we expect a $\sigma$ of 0.58%, with 8/10 values within 1 $\sigma$, and all within 2 $\sigma$ (normal distribution suggests 68% within 1 $\sigma$, 95% within $2\sigma$, so broadly agrees).

So I suggest run simulations for $N = 10000$, using test 1 and simple binomial confidence interval.