Solutions to Homework Assignment 1

36-462

January 2009

1. First, note that the derivative of the logistic map is $4r(1-2x)$. Second, the "stability criterion" is that a cycle $x_1, x_2, \ldots x_p$ of period p is stable when $\prod_{i=1}^{p} |f'(x_i)| < 1$.

   (a) The fixed points are the solutions to the equation

   $$x = f(x)$$

   For the logistic map,

   $$x = 4rx(1-x)$$

   or

   $$0 = (4r-1)x - 4rx^2 = x((4r-1) - 4rx)$$

   Written in this form, it is easy to see that there are two solutions,

   $$x = 0$$

   and

   $$x_f = \frac{4r-1}{4r} = 1 - \frac{1}{4r}$$

   which of course is only valid when $r \geq 0.25$.

   ```
   > fixed.point = function(r) {(4*r-1)/(4*r)}
   > curve(fixed.point,from=0.25,to=1,xlim=c(0,1))
   ```

   A fixed point is stable when the magnitude of the derivative there is < 1. We know that the $x = 0$ fixed point becomes unstable when $r > 0.25$. We will find where the other, non-zero fixed point x_f becomes unstable, r_2, by plugging it into the derivative:

   $$\begin{aligned} 1 &= \left| 4r_2 \left(1 - 2\frac{4r_2 - 1}{4r_2} \right) \right| \\ &= |4r_2 - 2(4r_2 - 1)| \\ &= |-4r_2 + 2| \end{aligned}$$

   or $r_2 = 3/4$.

(b) To show that the fixed point is stable below r_2 and unstable above it, there are several possibilities. One is to numerically calculate and plot the derivative at x_f as r varies through r_2.

```
fixed.point = function(r) {(4*r-1)/(4*r)}

logistic.map.derivative = function(x,r) {4*r*(1-2*x)}

stability.logistic.map.fp = function(r) {
   abs(logistic.map.derivative(fixed.point(r),r))
}

curve(stability.logistic.map.fp,from=0.25,to=1)
```

Another possibility is to use the expression for the stability criterion at the fixed point found above, $|2-4r|$. (Notice that nothing in the derivation of that assumed any special properties of r_2.) If $1/4 < r < 3/4$, then $1 < 4r < 3$, and $|2-4r| < 1$. Similarly, if $3/4 < r < 1$, then $3 < r < 4$ and $1 < |2-4r| < 2$.

(c) We need to locate the 2-cycle. Recall from the slides that the points on the this cycle need to be solutions of the equation

$$x = f(f(x))$$

Explicitly,

$$x = 4r(4rx(1-x))(1-4rx(1-x))$$

Writing this out

$$\begin{aligned} x &= 16r^2x(1-x)(1-4rx+4rx^2) \\ &= 16r^2x(1-4rx+4rx^2-x+4rx^2-4rx^3) \\ &= 16r^2x(1-(4r+1)x+8rx^2-4rx^3) \end{aligned}$$

Now, this is a fourth-order (quartic) equation, which has four solutions. Finding them is tedious but not impossible. However, as remarked in the class and in the slides, we already *know* two of the solutions: they are the fixed points. After all, if $x = f(x)$, then $f(f(x)) = f(x) = x$, so fixed points automatically solve the 2-cycle equation. One of those fixed points is $x = 0$, and it's plain from the way I wrote the equation above that this is, in fact, a solution. Since we're not interested in that, we can divide both sides of the equation by x, leaving us with a cubic equation.

$$\begin{aligned} 1 &= 16r^2(1-(4r+1)x+8rx^2-4rx^3) \\ 0 &= (16r^2-1)-16r^2(4r+1)x+128r^3x^2-64r^3x^3 \\ 0 &= \frac{1-16r^2}{64r^3}+\frac{1}{4r}(4r+1)x-2x^2+x^3 \end{aligned}$$

The other fixed point is $x = (4r-1)/4r$. This means that the cubic equation can itself be factored into a linear and a quadratic term[1].

$$0 = (x - \frac{4r-1}{4r})(x^2 + (-2 + \frac{4r-1}{4r})x + \frac{4r+1}{4r} - 2\frac{4r-1}{4r} + \left(\frac{4r-1}{4r}\right)^2)$$

The two remaining solutions, which make up the 2-cycle, are then given by a quadratic:

$$x_{\pm} = \frac{1}{2}\left(2 - \frac{4r-1}{4r} \pm \sqrt{-3\frac{(4r-1)^2}{16r^2} + 4\frac{4r-1}{4r} + 4 - 4\frac{4r+1}{4r}}\right)$$

To check that this monster is what we want, let's plot it:

```
x.cycle.p = function(r) {
   0.5*(2 - fixed.point(r)
        + sqrt(-3*fixed.point(r)*fixed.point(r)
               +4*fixed.point(r)+4 -4*(4*r+1)/(4*r)))
}

x.cycle.p = function(r) {
   0.5*(2 - fixed.point(r)
        + sqrt(-3*fixed.point(r)*fixed.point(r)
               +4*fixed.point(r)+4 -4*(4*r+1)/(4*r)))
}

curve(x.cycle.p,from=0.75,to=1,ylim=c(0,1))
curve(x.cycle.m,from=0.75,to=1,add=TRUE)
```

Rather than plug in to the algebra to determine stability directly — though that's possible! — let's use the computer to do it.

```
stability.lm.2cycle = function(r) {
  abs(logistic.map.derivative(x.cycle.p(r),r) *
      logistic.map.derivative(x.cycle.m(r),r))
}

curve(stability.lm.2cycle,from=0.75,to=1)
abline(h=1,lty=2)
```

(The last command draws a horizontal line at 1, for clarity.) This shows that the stability criterion for the 2-cycle starts at 1, at $r = r_2 = 0.75$, but then falls below 1 immediately, and stays below 1 until about $r \approx 0.86$. To give a more precise value of r_4, we can use the `uniroot` function, which solves one-dimensional equations.

```
uniroot(function(r) {stability.lm.2cycle(r)-1},interval=c(0.85,0.89))
```

[1]See, for example, Wikipedia, s.v. "cubic equation", section "factorization".

This gives $r_4 = 0.8623726 \pm 0.0000610$.

(d) In the previous part, we saw that the 2-cycle becomes unstable when $r > r_4 \approx 0.8623726$. It's enough to simulate here to see that it's attracted to a 4-cycle here.

Alternately, one could go through the route of heroic algebra. The points on the 4-cycle consist of the solutions to the equation

$$x = f(f(f(f(x))))$$

which are not also fixed points or points on the 2-cycle. The equation itself is an eighth order polynomial, with in general eight solutions, which could be obtained numerically via (say) `polyroot`. However, two of those solutions are the fixed points, and another two are the 2-cycle, which can be factored out by polynomial division, leaving a fourth-order polynomial for the 4-cycle proper. This would be a quartic, which can be solved by hand.

2. (a) A point θ is periodic if and only if

$$\theta = \theta + p\alpha \bmod 360 \tag{1}$$

for some integer p. This is equivalent to

$$p\alpha \bmod 360 = 0 \tag{2}$$

Suppose α is rational, i.e., $\alpha = m/n$ for integers m, n. Then the previous equation becomes the assertion that

$$pm/n \bmod 360 = 0 \tag{3}$$

for some p, but this is always true, e.g., for $p = 360n$. So, if α is rational, $p\alpha$ is always a multiple of 360 for some p, and so every point is periodic.

Suppose α is irrational. A periodic point would require an integer p such that $p\alpha \bmod 360 = 0$. But this would mean that $p\alpha$ was a multiple of 360, so that $\alpha = 360/p$. But then α would be the ratio of two integers, i.e., rational. Hence there can be no such p.

(b) A map is ergodic when time-averages converge on expectations under the invariant distribution. Or, said differently, the histogram we get from an *individual* time-series needs to converge on the histogram of the invariant distribution. (Look back to lectures 2 and 3.) So we want to modify the examples of histograms from the individual logistic map trajectories to get histograms from an individual circle-map trajectory. (This code is also in the accompanying R file.)

```
# Do one iteration of the rotation map
  # Notes:
```

```
  # 1. Through the magic of R vectorization, if given a vector of initial
  #    conditions, it will iterate them all in parallel
  # 2. Added optional argument "circle" for the measure of a full circle, so
  #    you can use radians (or whatever) rather than degrees if you want to
# Inputs: vector of angles (theta), angular increment (alpha), measure of
#         complete circle (circle, defaults 360)
# Output: vector of new angles
rotation.map <- function(theta,alpha, circle=360) {
  new.theta = (theta + alpha) %% circle  # "%%" is the modulus operator
  return(new.theta)
}

# Produce a rotation map time series
# Inputs: number of steps (timelength), angular increment (alpha), initial
#         condition (initial.cond, default is uniform random), measure of
#         complete circle (circle, defaults 360)
# Calls: rotation.map()
# Output: vector of length timelength
rotation.map.ts <- function(timelength,alpha,initial.cond=NULL,circle=360) {
  theta = vector(mode="numeric",length=timelength)
  if (is.null(initial.cond)) {
    theta[1] = runif(1,0,circle)
  } else {
    theta[1] = initial.cond
  }
  for (t in 2:timelength) {
    theta[t] = rotation.map(theta[t-1],alpha,circle)
  }
  return(theta)
}

# Evolve an initial ensemble according to the rotation map
# Inputs: number of time steps (timelength), angular increment (alpha),
#         vector of initial conditions (theta), measure of a full circle
#         (circle, default 360)
rotation.map.evolution <- function(timelength,alpha,theta,circle=360) {
  for (t in 1:timesteps) {
    theta <- rotation.map(theta,alpha,circle)
  }
  return(theta)
}
```

Figures 1 and 2 show, respectively, a single time series of length 10^4 from a random initial condition with $\alpha = 100\pi$, and the histogram obtained from that series. It is extremely close to uniform; in fact, a Kolmogorov-Smirnov test against the uniform distribution has a p-

value of 1 (to machine precision; the actual KS distance is 4×10^{-4}). (EXERCISE: what does the following code do?

```
hist(replicate(100,ks.test(rotation.map.ts(1e4,100*pi),punif,0,360)$statistic))
```

What should its results be if this map is ergodic with the uniform density as its invariant distribution? [You may want to look at `help(ks.test)`.])

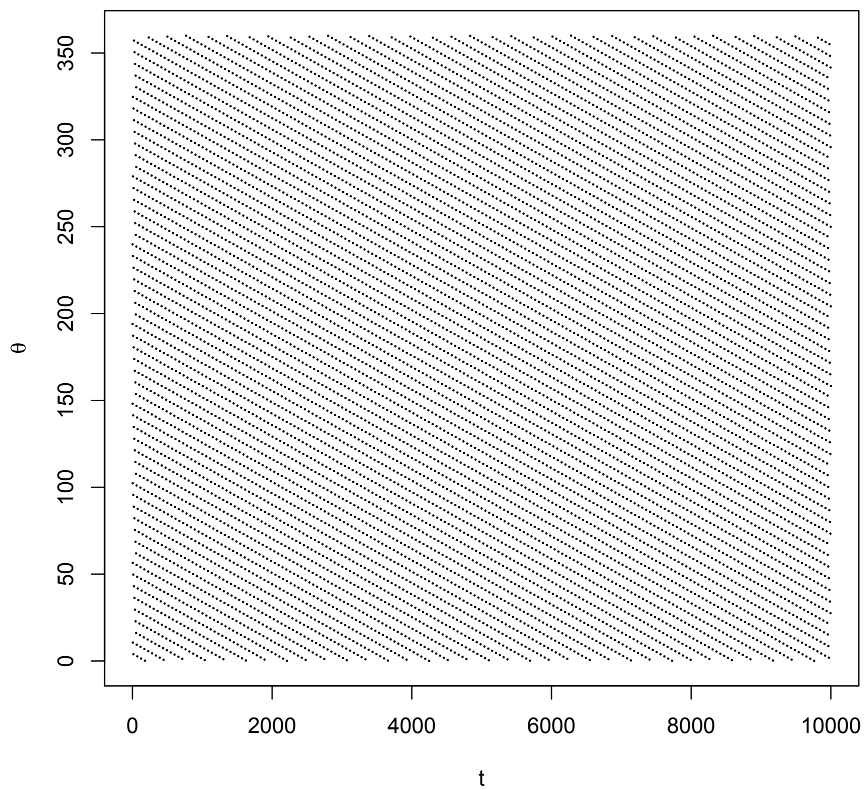
(c) A map is mixing when the correlations of all bounded, continuous functions go to zero. Alternately, any initial ensemble needs to converge on to the invariant distribution. To show it is not mixing, one can either show that correlations do not decay, or that different initial ensembles do not converge on a common invariant ensemble.

It's easier to *see* the second way. The effect of the map is to rotate every point by the same angle. If we start with a non-uniform distribution, all we will get is that same distribution shifted around the circle, never coming any closer to uniform. For example, if we start with points with a Gaussian distribution around 45 and a s.d. of 10 degrees (Figure 3), setting $\alpha = 100\pi$ and evolving for one step gives us Figure 4; evolving for a thousand steps gives Figure 5. The shape of the distribution is clearly not changing or becoming more uniform.

Alternately, on can check the autocorrelation function (Figure 6, and see that it is definitely not decaying to zero!

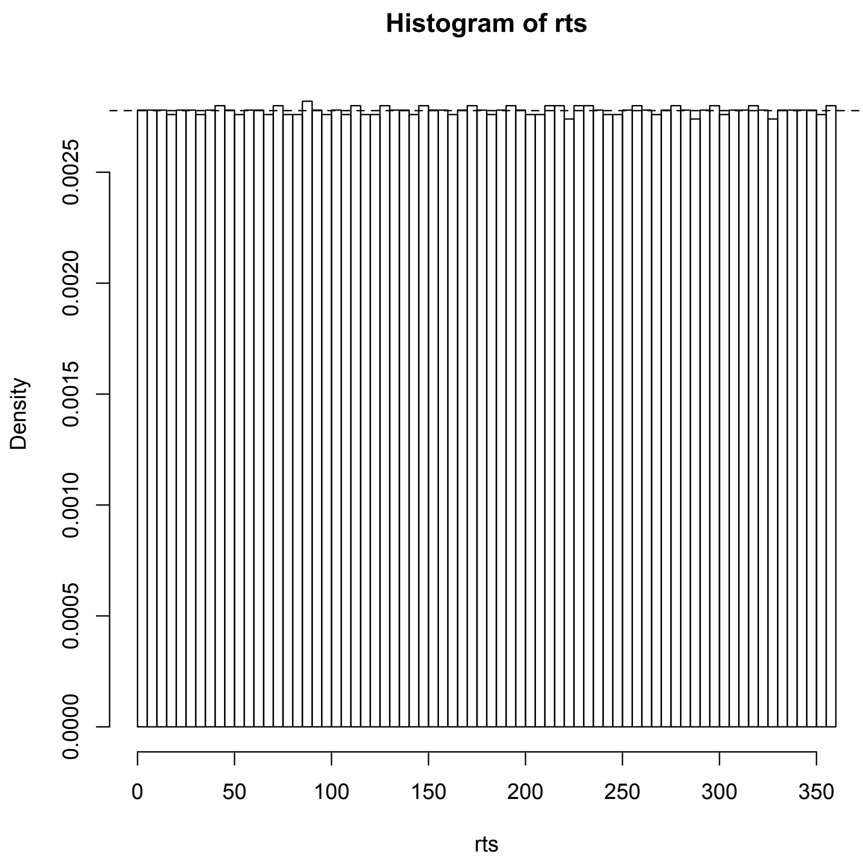
Finally, there is an easy way to see *analytically* that the circle map is not mixing. Pick any ensemble of initial conditions you like, so long as it is restricted to only a part of the circle — say a 10^{deg} arc. Clearly, at the next time step the ensemble will still be confined to a 10^{deg} arc, though a different one. And this will remain true for an arbitrary number of iterations. But the invariant distribution is uniform on the whole circle, so it's not confined to any 10^{deg} arc. Since this initial distribution does not converge on the invariant distribution, the map is not mixing.

(Notice that this argument works because there is no stretching to the dynamics — every interval gets mapped to an interval of exactly equal length. What happens with the map $\theta_t = 2\theta_{t-1} + \alpha \bmod 360$?)



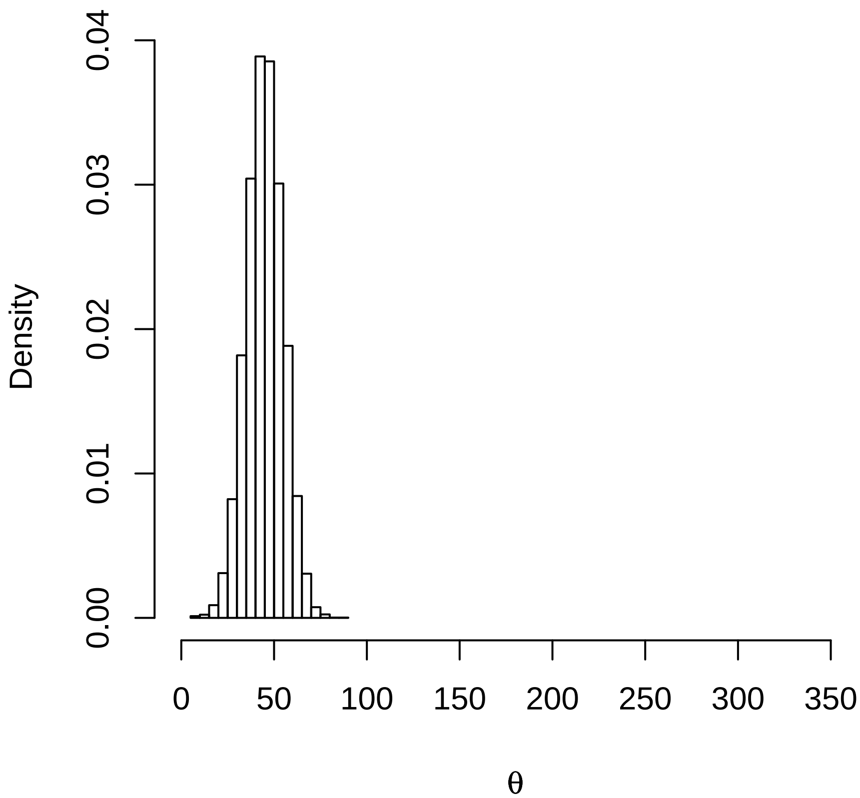

```
> rts = rotation.map.ts(1e4,100*pi)
> plot(rts,cex=0.1,xlab="t",ylab=expression(theta))
```

Figure 1: Typical time-series from the rotation map, with $\alpha = 100\pi$.



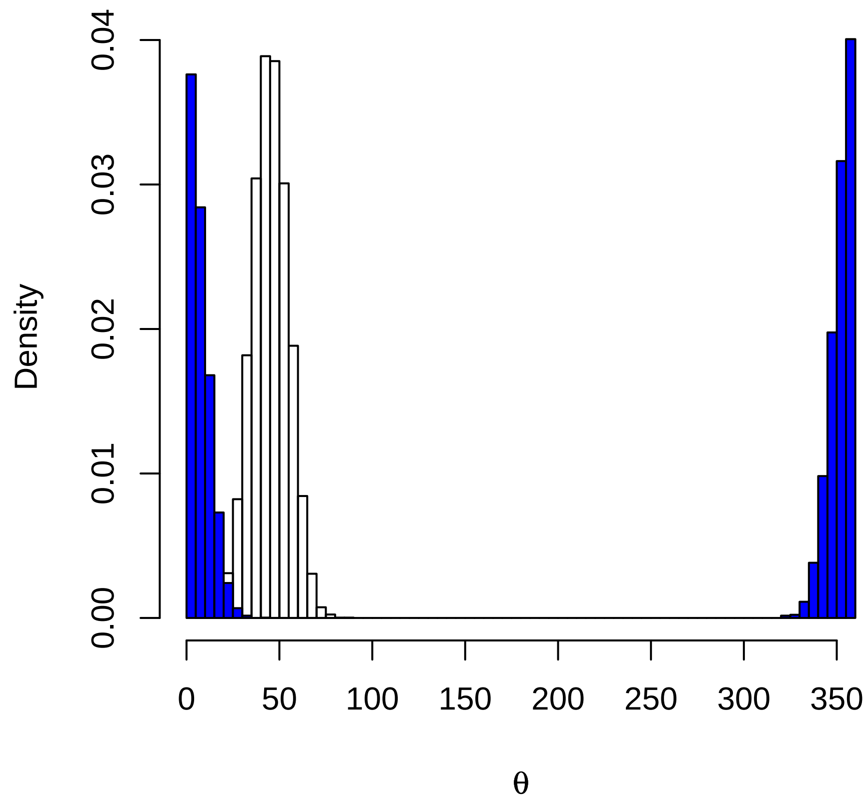

```
> hist(rts,n=101,probability=TRUE)
> abline(h=1/360,lty=2)
```

Figure 2: Histogram from the time series in the previous figure. The dashed horizontal line shows the uniform density over the circle.




```
> theta.0 <- rnorm(1e4,45,10)
> hist(theta.0,xlab=expression(theta),prob=TRUE,main="",xlim=c(0,360))
```

Figure 3: Gaussian ensemble of initial conditions for the rotation amp




```
> alpha = 100*pi
> theta.1 = rotation.map.evolution(1,theta.0,alpha)
> hist(theta.1,add=TRUE,col="blue",prob=TRUE,n=101)
```

Figure 4: As in the previous figure, but adding the first iterates of the initial conditions in blue.

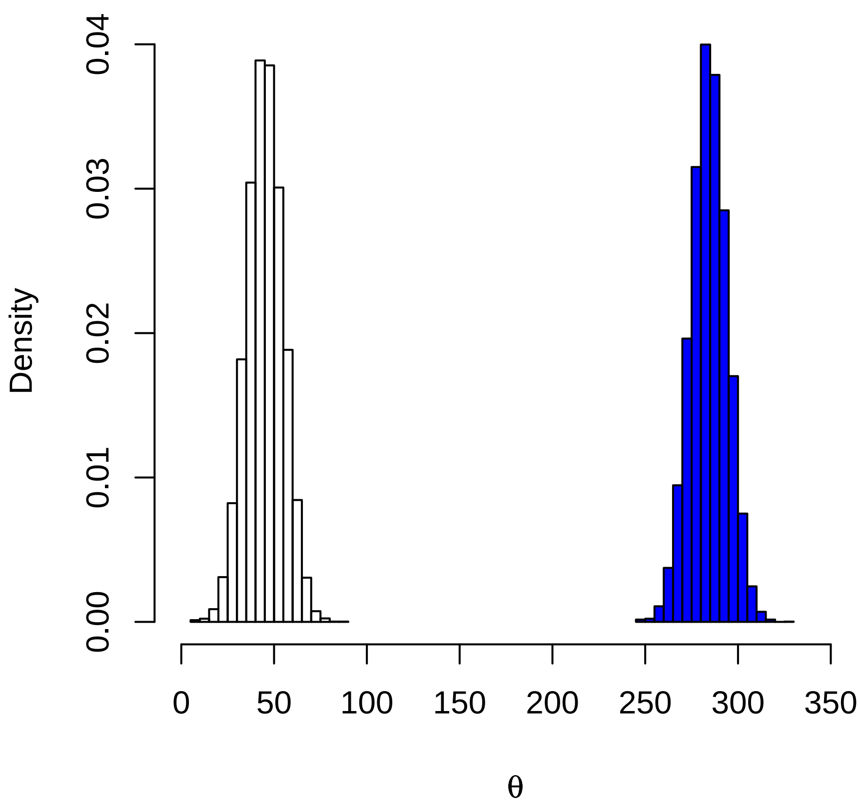
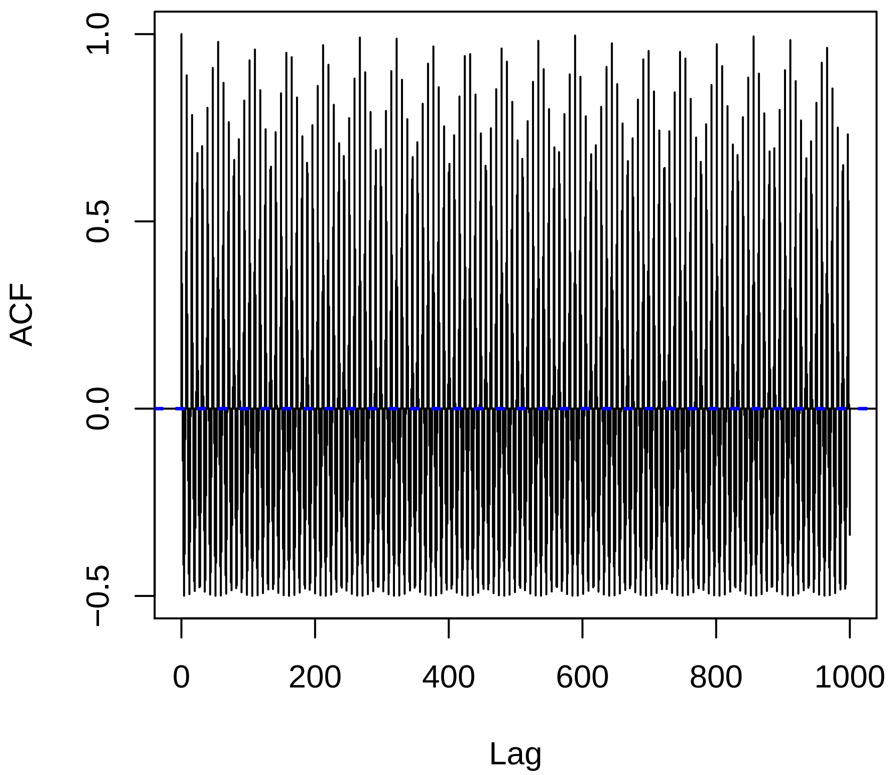


Figure 5: The ensemble of Figure 3 after 1000 iterations.




```
> theta.ts = rotation.map.ts(1e6,runif(1,0,360),alpha)
> acf(theta.ts,lag.max=1000,main="")
```

Figure 6: Autocorrelations out to lag 1000 from a rotation map time-series of length 1 million. Since θ is bounded, if the map were mixing this should be decaying to zero.