Stat 248, Spring 2021
Final Project: Circularly Coupled Markov Chain Sampling
May 3, 2021
Author: Seth Billiau, Christine Cai, Michael Yin

1 Paper Summary

1.1 Introduction

Circular coupling is a method of coupling that circularly reuses common random numbers. In Neal's words, circular coupling is in fact a "refinement" of the common random numbers coupling described in a paper by Johnson (1995). Circular coupling is designed specifically to avoid discarding burn-in times, which can alow for bias to remain: instead of running the Markov Chain process and discarding an arbitrarily selected first few terms, we can just take a "wrapped around" chain because it will be approximately distributed as the Markov chain's stationary distribution. Neal's proposed diagnostic for convergence, auxiliary chains, also allow for more rigorous checks of convergence speed than just using one sequence of common random numbers with the same starting point, as in Johnson. However, circular coupling has its shortcomings, and quite a few at that. Most notably, it can only guarantee a rough upper bound of total variation distance from the stationary distribution (not even a particular loose bound). This upper bound is also near-impossible to verify computationally in the first place, which is why one must approximate with diagnostic methods (that are sometimes quite computationally expensive). The coupling is also only successful if the wrap around chain converges by some fixed time N, but it is difficult to determine if it will even converge within a fixed time N – and it is also difficult to determine how much to increase N if it does not coalesce, so many bounds can seem somewhat arbitrary.

Hence, in practice, circular coupling does not actually seem to provide a significant improvement over standard Markov Chain sampling methods. The idea of coupling to past states may be reminiscent of more efficient and more recent forms of coupling, such as coupling lagged chains (Glynn and Rhee 2014), which avoids the issue of coalescence with the property that $X_t \stackrel{d}{=} Y_t$, and provides a much more deterministic bound on the total variation distance.

1.2 Circular Coupling

Let us formalize our treatment of circular coupling: take some (ergodic) Markov chain with stationary distribution π, and denote transitions on this chain by the deterministic function $\phi(\cdot, \cdot)$, which generates the state at a given point in time from the prior state (at time $t-1$) and a randomly generated number $u_{t-1} \sim U$ (for some fixed distribution U).

To generate a sequence with stationary distribution π with circular coupling, we first generate a chain x of N elements in the usual MCMC fashion, where x_0 is sampled from some starting distribution p_0 and we generate ensuing terms $x_t = \phi(x_{t-1}, u_{t-1})$ for the transition function ϕ defined above. (Assume that $u_t \overset{i.i.d}{\sim} U$.) Once we generate up until x_N, we "wrap around" like a circle and start a new chain y such that $y_0 = x_N$. We use ϕ to generate the remaining terms $y_t = \phi(y_{t-1}, u_{t-1})$ with the same samples u_t that we used to generate x, much like in common random number coupling. Notice that once we have some $y_t = x_t$, then because $\phi(\cdot, u_t)$ is deterministic and u_t is fixed to be the same for both y_t and x_t, the two chains will take on the same values for the remaining terms

until time N.

If this y chain coalesces with the x chain before y_N (equivalently, $x_N = y_N$), then we can say that the y chain approximately has distribution π. If this y chain does not coalesce, then the coupling process has failed and we choose a bigger N.

Formally:

Algorithm 1. Practical Circular Coupling

1. Similarly to many standard Markov Chain simulations, sample terms $u_t \overset{i.i.d.}{\sim} U$ for $t \in \{1, 2, ..., N\}$. Sample $x_0 \sim p_0$ (independently of the u_t terms), and generate $x_t = \phi(x_{t-1}, u_{t-1})$ for $t \in \{1, 2, ..., N\}$.
2. Define a new chain y starting at $y_0 := X_N$.
3. For $t \in \{1, 2, ..., N\}$, if $y_{t-1} \neq x_{t-1}$, then generate $y_t = \phi(y_{t-1}, u_{t-1})$.
4. Once $y_t = x_t$, set the remaining terms equal to the corresponding terms in the x chain. If this doesn't happen, repeat with larger N.

For the rest of this paper, we will refer to y as the "wrap around chain" and x as the "original chain".

In theory, this algorithm sounds promising: because we can directly use y as samples from π, we eliminate the need for initial burn-in states. In addition, we are not introducing more bias by using $y_0 = x_N$. But how can we verify that circular coupling actually converges to the desired distribution π?

1.2.1 Theoretical Approximate Correctness

Unfortunately, circular coupling cannot guarantee exact convergence. In Neal's paper, he shows that we can attain "approximate correctness". Because this is the only theoretical measure of efficacy provided, we will review this proof in a more in-depth manner:

Theorem 1 (Theorem of Approximate Correctness)**.** *Say that we have some $x \sim \pi$ and $u \sim U$. The theorem states that if we have some means of sampling from equilibrium distribution π, $\phi(x, u) \sim \pi$, then each point y_t ($t \in \{0, ..., N\}$) has a distribution within $2\epsilon + \delta$ of π in TV distance, where we define:*

1. *$1 - \epsilon$ is the lower bound of the probability that if two chains are started from states drawn from π (independently of u_t), they will coalesce within N/2 iterations.*
2. *$1 - \delta$ is the lower bound of the probability that if one chain is started from a state drawn from π (independently of u_t) and one chain is started from a state drawn from the initial state distribution p_0 (independently of u_t), they will coalesce within N iterations.*

Proof. Assume that conditions 1 and 2 of the theorem hold.

In order to bound the total variation distance of our wrap around chain from π, we want to compute the probability of a general wrap around chain being an "extreme" theoretical example of circular coupling in which each component of the wrap around chain already has distribution π.

Neal describes an algorithm to generate this theoretical chain as follows:

Algorithm 2. Theoretical Coupling

1. Similarly to the standard circular coupling algorithm, sample terms $u_t \overset{i.i.d.}{\sim} U$ for $t \in \{1, 2, ..., N\}$. Sample $x_0 \sim p_0$ (independently of the u_t terms), and set $x_t = \phi(x_{t-1}, u_{t-1})$ for $t \in \{1, 2, ..., N\}$.
2. Independently sample $v_0 \sim \pi$ and $w_{N/2} \sim \pi$.
3. For $t = 1, ..., N/2$: define $v_t = \phi(v_{t-1}, u_{t-1})$.
   For $t = N/2 + 1, ..., N$: define $w_t = \phi(w_{t-1}, w_{t-1})$.
4. Set $v^*_{N/2} = v_{N/2}$ to continue the v chain.
   Set $w^*_0 = w_N$ to wrap around the w chain.
5. For $t = N/2 + 1, ..., N$: define $v^*_t = \phi(v^*_{t-1}, u_{t-1})$.
   For $t = 1, ..., N/2$: define $w^*_t = \phi(w^*_{t-1}, w^*_{t-1})$.
6. Define our theoretical chain y^*_t.
   For $t = N/2, ..., N$: define $y^*_t := v^*_t$.
   For $t = 1, ..., N/2 - 1$: define $y^*_t := w^*_t$.

This is just a theoretical algorithm – presumably, actually sampling from π is difficult enough to warrant circular coupling and MCMC in the first place. Because π is the stationary distribution of our random sampling method and $v_0, w_{N/2} \sim \pi$, the successive elements of the chains v, v^* and w, w^* also have distribution π. Therefore y^*, being composed of v^* and w^*, also has distribution π, as desired.

Step 1 is identical to how one would generate the original chain in our practical circular coupling procedure (Algorithm 1). However, notice that y^*_t is only a successfully circularly coupled chain that can be generated from our practical circular coupling algorithm if the following three conditions hold:

I. Coupled draw condition: $y^*_t = \phi(y^*_{t-1}, u_{t-1}) \quad \forall t \in \{1, 2, ..., N\}$.

II. Convergence condition: $y^*_N = x_N$.

III. Wrap around condition: $y^*_0 = x_N$.

The coupled draw condition is already true for most elements of y^*. However, there is a "break" in the chain when it switches over from $w^*_{N/2-1}$ to $v^*_{N/2}$. Hence this condition is true if and only if

$$\begin{aligned} y^*_{N/2} = \phi(y^*_{N/2-1}, u_{N/2-1}) &\Leftrightarrow v^*_{N/2} = \phi(w^*_{N/2-1}, u_{N/2-1}) \\ &\Leftrightarrow v_{N/2} = w^*_{N/2}. \end{aligned} \tag{1}$$

Next, the convergence condition is true if and only if

$$\begin{aligned} & y^*_N = x_N \\ \Leftrightarrow\ & v^*_N = x_N. \end{aligned} \tag{2}$$

Finally, the wrap around condition is true if and only if

$$\begin{aligned} y^*_0 = x_N &\Leftrightarrow w^*_0 = x_N \\ &\Leftrightarrow w_N = x_N. \end{aligned}$$

If condition (2) is true, then we get the condition

$$w_N = v_N^*. \tag{3}$$

Hence we can conclude that y^* can be generated through a practically circularly coupled chain if all three of the numbered conditions above are true:

I. $v_{N/2} = w_{N/2}^*$.
This condition is true if we have convergence of v and w^* by time $N/2$. They both have distribution π by stationarity, but it is not immediate that w^* was sampled independently from u_t used in this chain, because w_0^* is a function of many u_t terms:

$$w_0^* = w_N = \phi(w_{N-1}, u_{N-1}) = \phi(\phi(w_{N-2}, u_{N-2}), u_{N-1}) = ...$$

However, this is why Neal chose to use two separate chains v_N and w_N and break at the point $N/2 : w_N$ is a chain started at time $N/2$ and so only depends on terms u_t for $t \in \{N/2, N/2+1, ..., N-1\}$, which does not overlap with the $u_t : \{1, ..., N/2-1\}$ used to generate this part of these chains.
Hence by independence of the u_t terms and the definition of ϵ in Theorem Condition 1, I is true with probability at least $1 - \epsilon$.

II. $v_N^* = x_N$.
This condition is equivalent to convergence of the chain v^* and x in N steps. v^* is just a continuation of v, which is a chain randomly sampled from π.
Hence by the definition of δ in Theorem Condition 2, II is true with probability at least $1 - \delta$.

III. $v_N^* = w_N$.
This condition is equivalent to convergence of the chain v^* with w in $N/2$ steps (because w is started at time $N/2$). Similarly to condition I, v^* is only generated with $u_1, ..., u_{N/2-1}$ and w is started at $N/2$, so there is no overlap between the u_t terms.
Hence by independence of the u_t terms and the definition of ϵ in Theorem Condition 1, III is true with probability at least $1 - \epsilon$.

Given these three conditions, observe that the probability of y_t^* not being practically coupled is the probability of at least one of these conditions not happening. Regardless of dependence between the conditions, we can bound this probability from above by the union bound:

$$P(\text{not I} \cup \text{not II} \cup \text{not III}) \leq P(\text{not I}) + P(\text{not II}) + P(\text{not III}) \leq 2\epsilon + \delta. \tag{4}$$

By the coupling inequality and the fact that y_t^* terms all have distribution π, if we let y_t be the practically coupled chain, we can conclude that

$$||y_t, \pi||_{TV} \leq 2\epsilon + \delta, \tag{5}$$

as desired. □

Readers of Neal's paper may be concerned that this bound seems messy and difficult to theoretically calculate – and these readers would absolutely be correct! While for some distributions and MCMC sampling schemes, we can intuit that this bound goes to 0 as N goes to infinity, it is difficult to calculate either ϵ or δ for fixed N. Coupled (pun intended) with the fact that it is already difficult to theoretically find a N that ensures convergence of the wrap around chain, we can see that circular coupling can be fairly inelegant, and requires quite a few regularity conditions. However, we can still try to simulate and empirically calculate some practical diagnostics.

1.2.2 Empirical Approximate Correctness

Because it is so difficult to find ϵ and δ as defined in the Theorem of Approximate Correctness, Neal assumes that Conditions 1 and 2 of that theorem are true and overlooks some of the dependencies between the wrap around chain and random numbers u_t, thus assuming that the wrap around chain will coalesce with high probability.

Because we cannot directly empirically verify Condition 1 (because this requires us to be able to sample a chain from the stationary distribution π, defeating the point of circular coupling in the first place), we can only attempt to check Condition 2. To do so, we generate auxiliary chains, which are started at various points in time later in the chain. For some r that divides N, Neal suggests starting the auxiliary chains at times $i * N/r$ for integers $i \in \{1, ..., r-1\}$. We can then find the time c_i that it takes for auxiliary chain i to coalesce with the wrap around chain. Because the time for the wrap around chain to coalesce should be much smaller than N, we can say it is bounded above by some $k < N/2$ and simulate each auxiliary chain only up until time k. If the coalescence times are all around as large as k, this is an issue because the coalescence times should be much smaller than N, so the diagnostics should be rerun with larger k.

If we indeed get that all auxiliary chains converge quickly, then we can also conclude that Condition 1 is likely true. This is because if we have two chains started from π, and say that they converge with the same chain started from p_0 by time $N/2$ with some probability that is at least $1 - \varepsilon$, then the two chains will also be coalescing with each other because they are converging to the same chain. Hence the probability that two chains started from π coalescing is at least $1 - 2\varepsilon$. By symmetry (because a chain started from π coalescing to p_0 is the same as the chain started from p_0 coalescing to the chain started from π), then if out of our auxiliary chains there is only a small proportion less than $\epsilon/2$ of chains started from p_0 not converging within time $N/2$, then two chains started from π will converge within time $N/2$ with probability $1 - \epsilon$.

To run these auxiliary chain diagnostics on multiple processors, Neal suggests a parallel simulation algorithm that roughly calculates the sequential process with auxiliary chains described before:

Algorithm 3. Parallel Simulation Algorithm
For each processor $i = 0, ..., r-1$,

1. Define $s := iN/r$. Randomly draw $u_t \overset{i.i.d}{\sim} U$ for $t = s, ..., s + N/r - 1$.
2. Draw $y_s \overset{i.i.d}{\sim} p_0$
3. For the t defined above generate $y_t = \phi(y_{t-1}, u_{t-1})$.
4. Set $z = \phi(y_{s+N/r-1}, u_{s+N/r-1})$ and send it to processor $i + 1$ as the value for $y_{s+N/r}$.
5. Repeat the steps 3 and 4 after receiving a new value for y_s. If all processors are waiting, end the program.

The computation time for this is bounded below by the computation time of running N/r Markov chain iterations, and if all of the chains coalesce within N/r iterations then the runtime is bounded above by the time used for $2N/r$ Markov chain iterations. Unfortunately, in the likely scenario that not all the chains coalesce, there is no clean upper bound on runtime. In fact, in certain edge cases described by Neal (such as when different starting points lead to different wrap around chains), this algorithm may not terminate, so fail-safes should be put in place.

1.3 Random-Grid Metropolis

Of course, the idea of circular coupling is most valuable if such a coupling can actually be implemented. This requires Markov chain transitions that are easily computable, and quickly lead to coalescence. To this end, Neal chooses random-grid Metropolis updates.[1] Essentially, a random-walk Metropolis algorithm with proposals distributed uniformly within some distance w of the current state is straightforward, but due to the continuous nature of each proposal distribution, the probability of exact coalescence is 0. Instead, we can consider discretizing the proposal space into a "grid" of points, with the position of the grid chosen uniformly at random. In the one-dimensional case, this means the points all fall along a line with distance $2w$ between adjacent points. We then propose a move to whichever point is closest to the current state, accepting with probability

$$\frac{\pi(X^*)}{\pi(X_{t-1})}$$

(where π is the stationary distribution, X^* is the proposal, and X_{t-1} is the current state) and remaining at the same state otherwise.[2]

Note that the grid is re-chosen (at random) at each time period, which is what allows the equilibrium distribution to be continuous rather than discrete. Choosing the point on a randomly placed grid nearest our current state as the proposal is also very different from how we conceptualize random-walk Metropolis, where we simply choose a random proposal within an interval around our current state. Having all chains use the same grid in any given round, then, is what allows for a positive probability of coalescence (since many different possible starting states will all have the same nearest point on the grid): this is the idea behind the one-dimensional random-grid Metropolis algorithm!

1.3.1 Notation and Intuition

To solidify this algorithm and how it can be implemented (which we do in R in section 2.1 of this paper), let us introduce the following notation for the one-dimensional case:

$$u = (u_0, u_1)$$

$$f(x,u) = 2w\left[\left(u_1 - \frac{1}{2}\right) + Round\left(\frac{x}{2w} - \left(u_1 - \frac{1}{2}\right)\right)\right]$$

$$\phi(x,u) = \begin{cases} f(x,u), \text{ if } u_0 < \frac{\pi(f(x,u))}{\pi(x)} \\ x, \text{ otherwise} \end{cases}$$

Here, $f(x,u)$ is the next proposal given the current state x and a pair of i.i.d. $Uniform(0,1)$ random variables u, and $\phi(x,u)$ is the actual next state that either realizes the proposal after it is accepted or rejects the proposal and remains at the same state.

Note that u_0 implements the Metropolis "coin flip" that determines whether the proposal is accepted or rejected, while u_1 is essentially determining where the grid is located (thus determining the proposal). To see this, observe that for $\tilde{f}(x)$, which is the same as $f(x,u)$ but using 0 in place of $u_1 - \frac{1}{2}$, we have

$$\tilde{f}(x) = 2w\left[Round\left(\frac{x}{2w}\right)\right]$$

[1] For consistency with Neal's paper, we will refer to the Metropolis-Rosenbluth-Teller algorithm as the Metropolis algorithm though there is reason to believe that calling it the Rosenbluth-Teller algorithm would be more appropriate.

[2] Note that this is the Metropolis algorithm without Hastings' extension to non-symmetric proposal distributions.

This is clearly a function that finds the nearest point on the grid to x when one of the points falls exactly at 0. Returning to $f(x,u)$ by subtracting out $u_1 - \frac{1}{2} \sim Unif\left(-\frac{1}{2}, \frac{1}{2}\right)$ before rounding and adding it back afterwards, then, repositions the grid uniformly at random.

It was not immediately obvious to us that the random-grid Metropolis algorithm results in a stationary distribution of π, which is of course desirable for sampling from π using this method. Like with random-walk Metropolis, however, the stationary distribution actually does turn out to be π. To see this, notice that when we have a current state x, the nearest point on the grid will be distributed uniformly in an interval of width $2w$ around x; this will also be the proposal $f(x,u)$. This is the same as for our random-walk Metropolis algorithm: such a proposal would also be distributed uniformly in an interval of width $2w$ around x! As such, we retain helpful properties from random-walk Metropolis, such as the desired stationary distribution of π.

Returning to the original circular coupling paper, after introducing random-grid Metropolis, Radford Neal makes a few brief remarks on its effectiveness for circular coupling. In particular, he states that assuming w is chosen so that the acceptance rate is fairly high (substantially greater than $\frac{1}{2}$), then coalescence takes about as long as convergence to the equilibrium distribution, since the paths of two coupled chains must cross eventually when both sample from the equilibrium distribution, and when the two chains are within $2w$ it is likely to have both accept the same proposal state. Next, Neal utilizes random-grid Metropolis in an example of circular coupling and auxiliary diagnostic chains to check that the wrapped-around chain seems to come from the equilibrium distribution: we replicate much of this analysis in section 2.1 of this paper.

1.3.2 Multi-Dimensional Extension

Next, Neal discusses extending random-grid Metropolis to the multi-dimensional setting, and extensively analyzes an example using multivariate examples. It is possible to do so by updating the components of the state one at a time and using a one-dimensional random-grid strategy, but a multi-dimensional grid strategy can be employed instead. To do so, we must first redefine proposal states to be vectors of coordinates, and u to be a longer vector of uniform random variables, which we do as follows for the d-dimensional case: $u = (u_0, u_1, ..., u_d)$

$$[f(x,u)]_i = 2w\left[\left(u_i - \frac{1}{2}\right) + Round\left(\frac{x_i}{2w} - \left(u_i - \frac{1}{2}\right)\right)\right]$$

Now, each u_i is used to determine the ith component of the proposal state, while u_0 is still used to determine whether the overall proposal will be accepted or rejected. $\phi(x,u)$ does not need to change, since $\pi(f(x,u))$ and $\pi(x)$ will still just be probabilities. The intuition here is that we are laying down a d-dimensional grid, still with location determined uniformly at random, and again choosing the point on the grid nearest our current state as the proposal.

Though he does not explain the details, Neal notes that both the single-component and multi-dimensional random-grid Metropolis strategies tend to bring distant coupled chains close together; however, this is not necessarily true of chains that are already relatively close, which can mean that exact coalescence is slow. Using sampling from a multivariate Normal distribution as his example, he demonstrates (under the assumption that it is rare for one chain to accept its proposal and the other to reject, so we don't see this occur in any two consecutive time periods) that two chains that are far apart tend to grow systematically closer until they get down to a distance proportional to our w. Finally, Neal simulates a coupling with a nine-dimensional multivariate Normal distribution to empirically demonstrate the distance reduction properties he showed theoretically; he notes that

the single-component updates worked better than the multi-dimensional random-grid Metropolis strategy here, but that this result was specific to the choice of 9 dimensions rather than something larger. Overall, these distance reduction properties for random-grid Metropolis in multiple dimensions (whether using the single-component or multi-dimensional update strategies) rely on a variety of assumptions and are highly limited. This is why Neal actually suggests using random-grid Metropolis for exact coalescence, but combining it with other coupled updates; we discuss such strategies in the next section of this paper.

1.4 Combining Random-grid and Other Coupled Updates

Random-grid Metropolis updates are capable of causing exact coalescence, and the probability of exact meeting increases as chains get closer together. However, random-grid Metropolis is not a very efficient way of sampling from the target distribution, and it is not the most effective way of bringing the chains closer together.

Therefore, Neal introduces the following practical strategy for circular coupling detailed on page 27 of the paper:

1. An update, or series of updates, that is designed to efficiently sample from the target distribution, and that is coupled so as to cause chains to approach closer.
2. A random-grid Metropolis update, which can lead to exact meeting of chains that are already close together.

Step 2 of the strategy utilizes the random-grid Metropolis method described in Section 1.3 that allows for the possibility of exact meetings. Neal points out that it is probably best to perform only a single random-grid update because random-grid updates tend to move chains further apart when unsuccessful in causing them to meet exactly. More rigorously, one can show that if a Metropolis proposal in a random-grid update is accepted by both chains, but fails to induce meeting, then a second identical random-grid update must also fail. On the other hand, if one or both chains rejects the Metropolis proposal, a second update may induce meeting, but will, with high probability, move chains farther apart if it fails.

Neal also shows via a "rough analysis" that for high-dimensional problems, a single multi-dimensional update should be preferred over the individual component updating approach described in Section 1.3.2; however, Neal concedes that exceptions to this rule may occur and that a preference between the two depends on dependence between components and the choice of coupling in step 1.

With regards to step 1, Neal acknowledges the vast literature on Markov chain updates that cause chains to get closer together. We have studied several methods in Stat 248 in our discussion of common random numbers, drift and minorization conditions, and contraction and iterated random functions.

To use Neal's circular coupling strategy, however, we limit ourselves to methods that utilize common random numbers, specifically methods that use the same number of random numbers per update irrespective of the state of the chain. This is because circular coupling requires us to run the (Y_n) chain with the same random numbers as the (X_n) chain. If the number of random variates used is not deterministic, then it is impossible to guarantee that the (X_n) and the (Y_n) chain will use the same random numbers and therefore no chance that the two chains will meet exactly. Neal discusses three coupling strategies with these desired properties: coupled Metropolis, coupled Metropolis-adjusted Langevin, and coupled Gibbs sampling.

1.4.1 Random-Walk Metropolis

As a possibility for Step 1, Neal discusses coupled Metropolis updating that uses a symmetric proposal distribution centered at the current state. For most common symmetric proposal distributions like the uniform or the normal distribution, there are well-known ways to sample from the proposal distribution using a deterministic number of auxiliary random variables (ex. Box-Muller for the Normal distribution). Then, the proposal is accepted based on a draw from a uniform random variable. From this construction, it is clear that coupled Metropolis updates satisfy the conditions for use with circular couplings. Though Neal does not mention this, it would also be relatively simple to use Metropolis-Hastings updates without a symmetric proposal distribution provided that the proposal also uses a deterministic number of auxiliary random variables.

Unfortunately, while coupled Metropolis does lead to meeting eventually when used for circular coupling, it does so at a very slow rate, especially in high dimensions. Though empirical simulation, Neal concludes that using Metropolis updates with a normal proposal distribution might be a feasible approach if the user is willing to vary the variance of the proposal distribution and the choice of w in the random-grid with time. Even then, however, the Metropolis updating seems poorly suited for circular coupling as the time that it takes for chains to get close enough to induce meeting is on the order of a couple hundred thousand iterations in a test case with a 9-dimensional target distribution. More efficient updating methods, such as Metropolis-adjusted Langevin sampling, should be used if necessary.

1.4.2 Metropolis-adjusted Langevin

Langevin methods are useful for continuous distributions where the gradient of the probability density can be computed. The Metropolis-adjusted Langevin algorithm (MALA) as described on page 37 of the paper provides exact convergence to the correct stationary distribution. Given a current state x_t, MALA introduces a momentum vector p and uses that momentum vector to improve efficiency as the proposal follows the gradient of the target density. Defining $E(x) = -\log \pi(x)$ and $H(x,p) = E(x) + |p|^2/2$ and stepsize hyperparameter ϵ, the algorithm follows these five steps:

Algorithm 4. Metropolis-adjusted Langevin Algorithm (MALA)

1. Replace $p \sim \mathcal{N}(0, I)$ with an independent draw from the multivariate normal distribution with mean 0 and covariance I.
2. Set $p' = p - (\epsilon/2)\nabla E(x)$.
3. Set $x^* = x_t + \epsilon p'$.
4. Set $p^* = p' - (\epsilon/2)\nabla E(x^*)$.
5. Accept (x^*, p^*) with probability $\min[1, \exp(H(x_t, p) - H(x^*, p^*))]$
   Else, keep x unchanged but negate p.

Neal shows via empirical simulation that this scheme when combined with random-grid Metropolis provides meeting times that are much faster than random-walk Metropolis. Using the same 9-dimensional test distribution as before, Neal found convergence in times on the order of 1,000 steps instead of several 100,000 steps used in random-walk Metropolis

Alternatively, Horowitz 1991 and Neal 1996a replace step 1 with the following: Change p to $\alpha p + (1-\alpha^2)^{1/2} n$, where $n \sim \mathcal{N}(0, I)$. This usually improves efficiency in bringing chains closer together at the cost of introducing α as an additional persistence hyperparameter. Using this improvement,

Neal discovered convergence at around 200 steps using the same test distribution.

1.4.3 Gibbs sampling

Gibbs Sampling with circular coupling can be conducted quite similarly to how it would be without circular coupling. The only difference is that instead of using other means to directly sample from the conditional distribution, we circularly reuse the random seeds and "wrap around" the conditional information from x_N. We can do so through inverse sampling: applying the inverse of the conditional CDF of the distribution to $u_t \overset{i.i.d}{\sim} \text{Unif}(0,1)$. Since this uses a fixed number of uniform random variables, it is also a viable candidate for circular coupling. For the most part, Neal uses Gibbs sampling when we have full conditional distributions of a set of many parameters. We will give a more thorough discussion of how circularly coupled Gibbs operates in section 2.

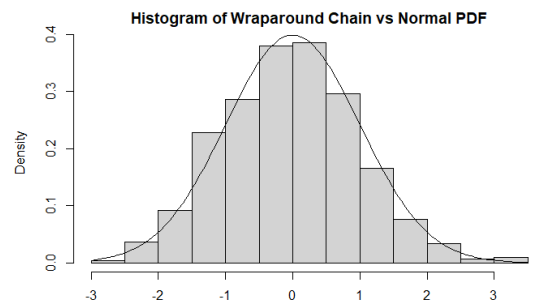
2 Sampling Method Implementations

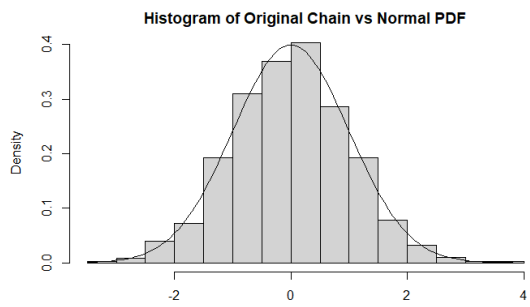
To further explore circularly coupled Markov Chains, we implemented some of the techniques illustrated in the paper in R. Our code for these implementations, as well as our logistic regression implementation in section 3 of this paper, can be found in our GitHub repository: `https://github.com/sethbilliau/circular-coupling/`.

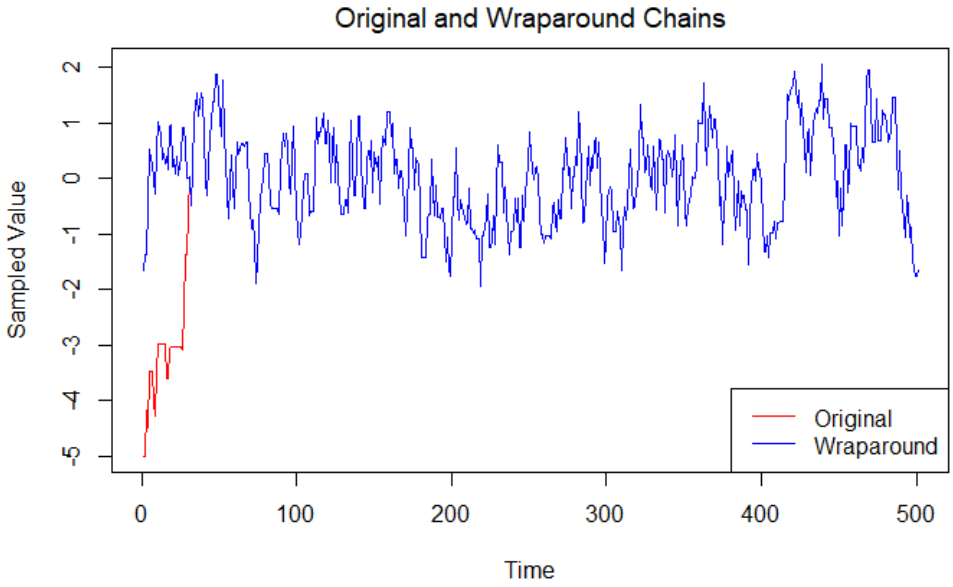
2.1 Random-Grid Metropolis

First, we implement the one-dimensional random-grid Metropolis algorithm, since it is useful both for Radford Neal's logistic regression implementation that we replicate below, as well as for an example of circular coupling and auxiliary diagnostic chains. We also implement the multi-dimensional random-grid Metropolis algorithm (not to be confused with the one-component algorithm for multivariate distributions), since it is used in an update step for the logistic regression example, but we do not show any results in this section for the sake of brevity.

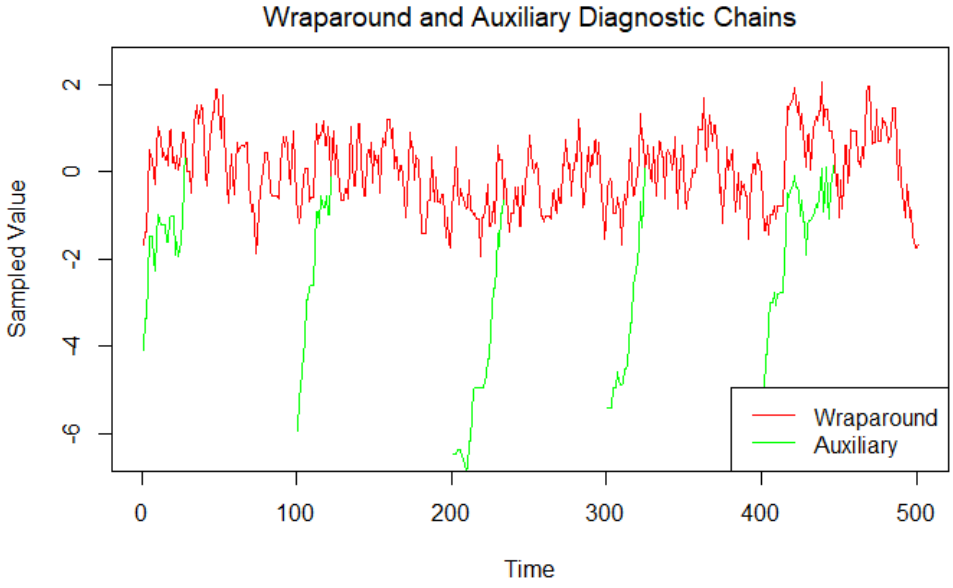
As shown in the third plot below, for our example using coupled random-grid Metropolis, we see coalescence between the original and wrap around chains quite quickly, using a standard normal distribution as the target and a starting state of -5. Running several more simulations, we can see that the histogram of draws from the wrap around chain (using the last iteration as the draw) follows a standard normal PDF well, though it is not noticeably better than the original chain by much.







Finally, using the auxiliary diagnostic chain strategy of Neal's paper with 5 chains, we can see that these diagnostic chains all coalesce with the wrap around chain fairly quickly as well; this does not necessarily imply the approximate correctness of the circular coupling procedure, but it does hint that approximate correctness may indeed hold, as Neal discusses at length.



2.2 Gibbs

Similarly to Neal, we simulate a circularly coupled Gibbs Sampler on a multivariate normal prior with a multivariate normal posterior. Say that our target distribution $\pi = \mathcal{N}(\mu, \mathbf{V})$, so for multivariate normal vectors $\vec{Y_1}, \vec{Y_2}$, with vector of means $\vec{\mu}$ and covariance matrix V, we have $(\vec{Y_1}, \vec{Y_2}) \sim \mathcal{N}(\vec{\mu}, V)$.

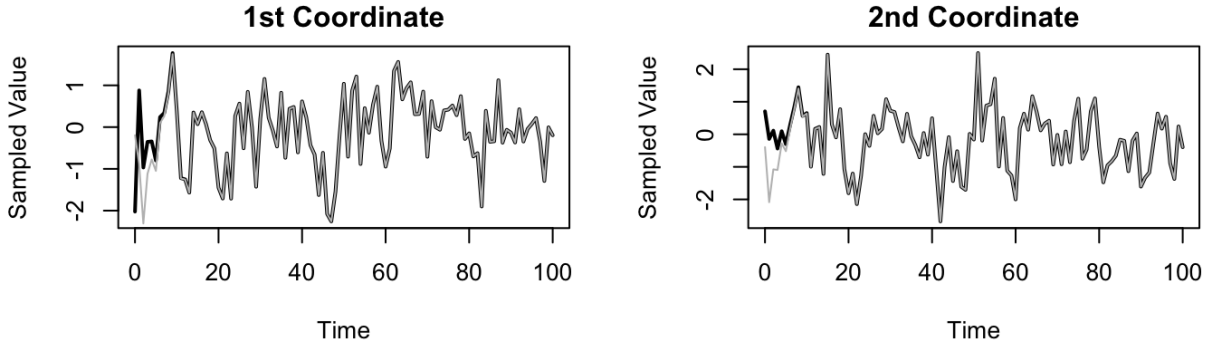
For this simulation, we will let the MVN vector have 4 elements, so let $\vec{Y_1}$ be a trivariate normal distribution, and let $\vec{Y_2}$ be the normal distribution that we are currently trying to update. Then, if we write the covariance matrix as $V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$, we can write the conditional distribution (to use for updating) as

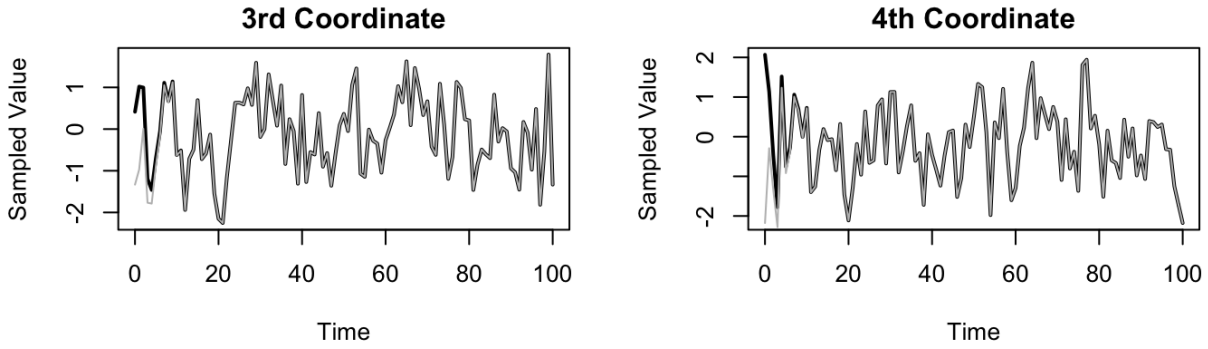
$$\mathbf{Y}_2|\mathbf{Y}_1 \sim \mathcal{N}\left(\mu_2 + V_{21}V_{11}^{-1}(Y_1 - \mu_1), V_{22} - V_{21}V_{11}^{-1}V_{12}\right).$$

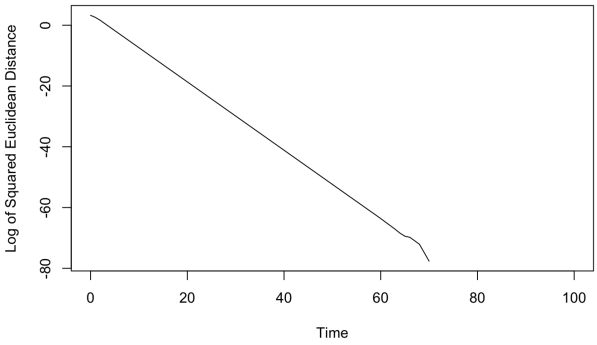
For ease of computation in this Gibbs sampler, we will choose a symmetric covariance matrix V such that the covariance between any pair of normals is the same, and the variance of any normal is also the same. Hence $V_{22}, V_{21}, V_{12}, V_{11}$ stay the same from term to term that we are updating.

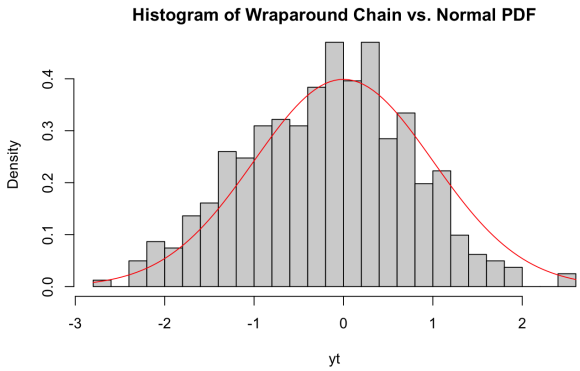
Let our starting distribution p_0 be a series of i.i.d. Normals.

We got that convergence happened quite quickly for $N = 100$, within roughly 10 terms for each element of our multivariate normal. However, we can see from our histogram that our y_t chain has a density that is skewed compared to the correct distribution of π.

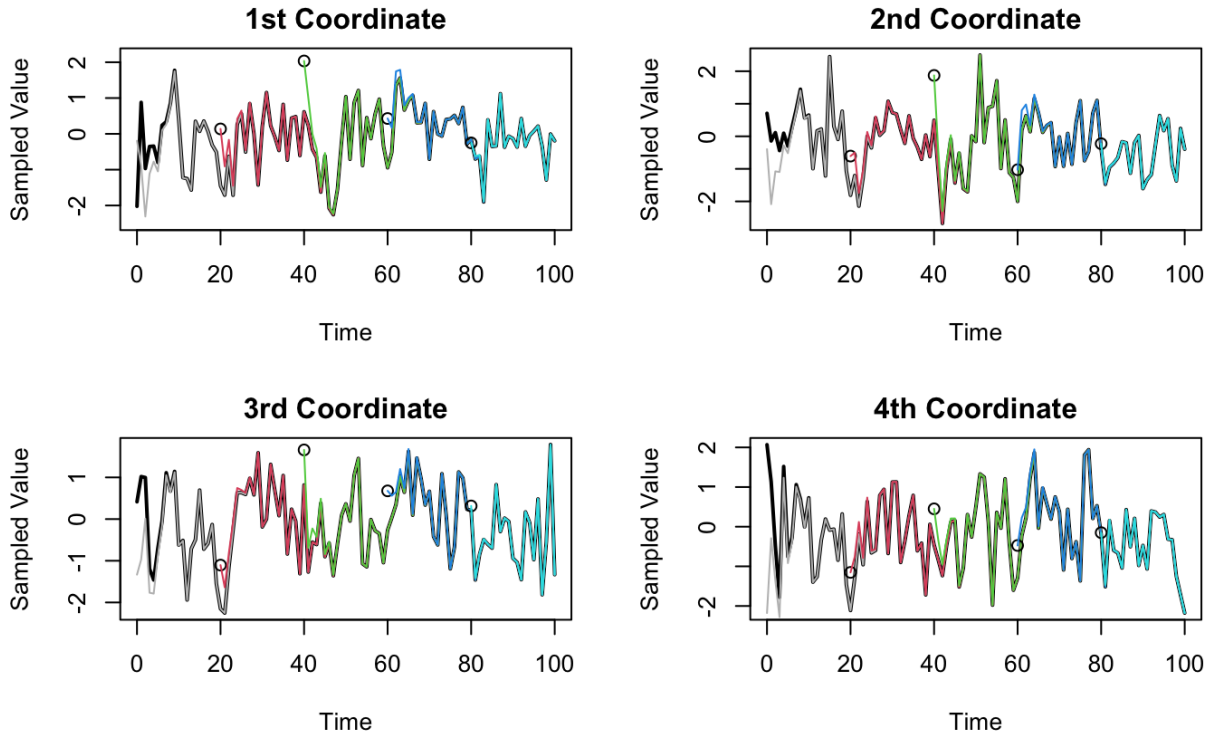








When testing auxiliary chains, we also got that convergence was quite fast, and we got convergence of auxiliary chains for both $r = 4$ and $r = 5$ (shown).



3 Logistic Regression Implementation

To illustrate how circular coupling works in practice, Neal proposes the following Bayesian hierarchical logistic regression problem. Suppose that you have a 150×4 predictor matrix X whose columns are the predictors x_1, x_2, x_3, x_4. You also have a response vector $C \in \{1, 2, 3\}$. Let $k \in 1, 2, 3$ and $i \in 1, \ldots, 150$. Then, $C \mid X$ is modeled as follows:

$$P(c_i = k \mid X) = \frac{\exp(z_i k)}{\sum_{k'=1}^{3} \exp(z_{ik'})} \text{ where } z_{ik} = b_{0k} + \sum_{j=1}^{4} b_{jk} x_{ij}$$

The prior specifications are as follows:

$$\begin{aligned} b_{0k} &\sim \mathcal{N}(0, 1), \ k = 1, 2, 3 \\ b_{jk} \mid \tau_j &\sim N(0, 1/\tau_j), \text{ for } j = 1, 2, 3, 4, \text{ and } k = 1, 2, 3 \\ \tau_j \mid \tau^* &\sim \text{Expo}(\tau^*), \text{ for } j = 1, 2, 3, 4 \\ \tau^* &\sim \text{Expo}(1) \end{aligned}$$

Predictor variables X were simulated from a multivariate normal with mean 0, variance 2 and correlation 1/2 and class variables were simulated accordingly with specific values of b_{kj} given in equation 34 of the paper.

Then, Neal sampled from the posterior distribution using the following sampling strategy. One iteration of the Markov Chain went according to the following algorithm:

Algorithm 5. Bayesian Logistic Regression Algorithm

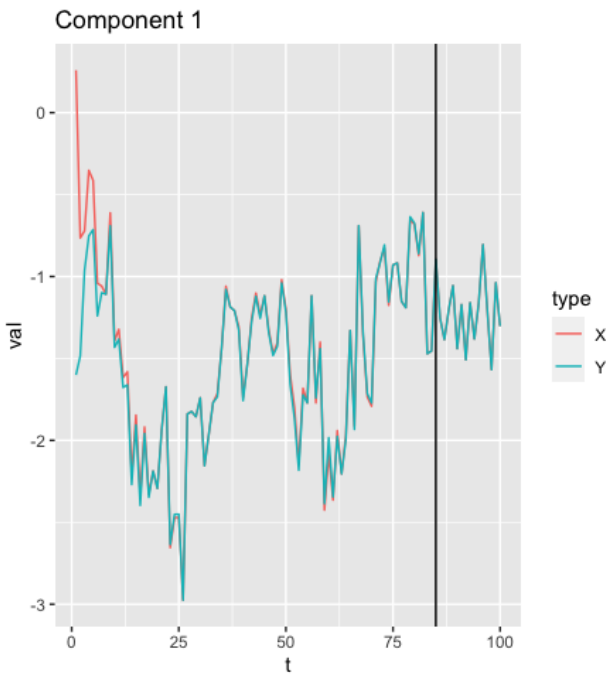
1. 10 repetitions of the following:
   (a) 10 Langevin updates with stepsize $\epsilon = 0.05$ and persistence $\alpha = 0.97$
   (b) 25 Random-grid Metropolis updates for $\log(\tau^*)$ using a proposal with $w = 0.1$
   (c) 1 Gibbs sampling update for each of the τ_j
2. 1 Random-grid Metropolis update for $\log(\tau^*)$, using a proposal with $w = 0.01$.
3. 1 Gibbs sampling update for each of the τ_j
4. A replacement of the momentum variables by values drawn from $\mathcal{N}(0, I)$

Neal did not provide replication code for this paper, so we reimplemented this sampling scheme. Our implementation of this sampling scheme was relatively inefficient as we used numerical computation of the gradient of $E(x)$ instead of calculating the gradient analytically. Nonetheless, we are happy to have a working implementation given the complexity of this sampling scheme.

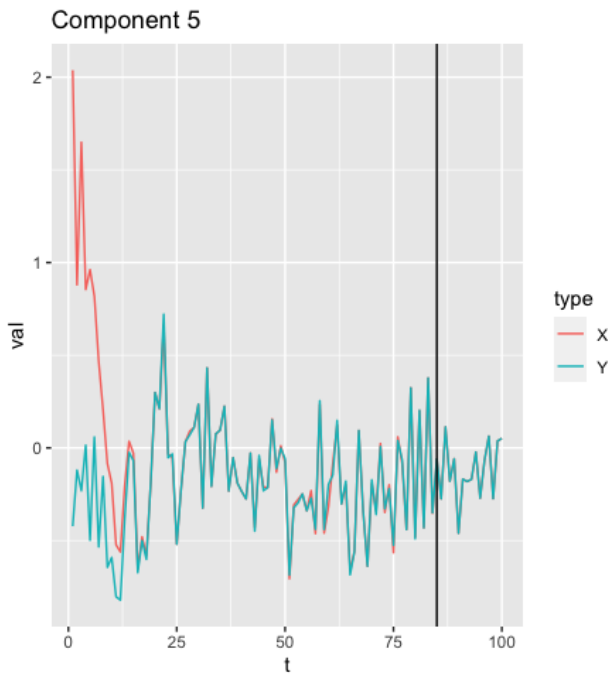
Overall, we found that implementing the circular coupling portion of this sampling scheme to be relatively straight-forward. However, it is not obvious how to set the value of N without some trial and error nor is it obvious how best to tune the hyper parameters w for the random-grid Metropolis updates.

Figures of the circular coupling are given below to the traceplots of a select subset of components. We see that in general, the (X_n) chain begins at a more over-dispersed initial distribution than the (Y_n) chain which begins where the (X_n) chain ends (note that the initial state of the (Y_n) chain is not plotted at time 0 to avoid redundancy). We found coalescence (and faithful coupling) at 85 steps even though the chains became very close visually after around 20 steps. It is interesting that it took another 60 or so iterations for exact meeting even after chains became closely coupled. This illustrates that even for visually close chains, the chance of meeting for a single update of random-grid Metropolis in 15 dimensions is still not a guarantee.

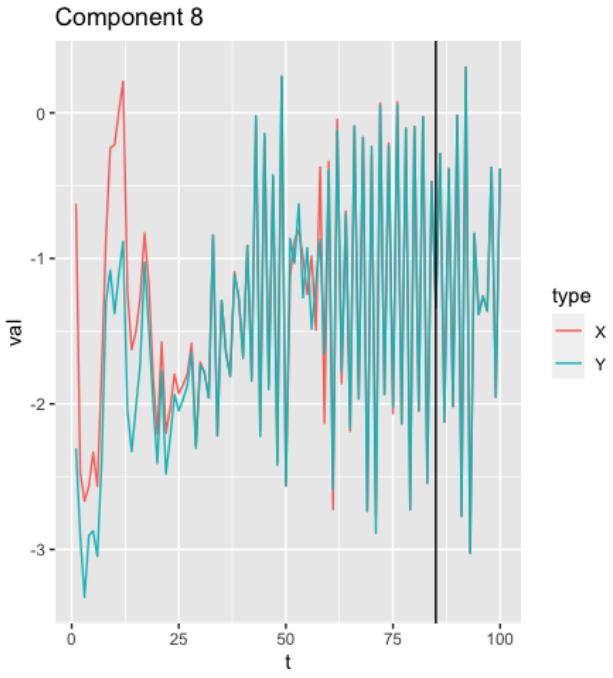
Figures of a subset of individual component traces are given below:



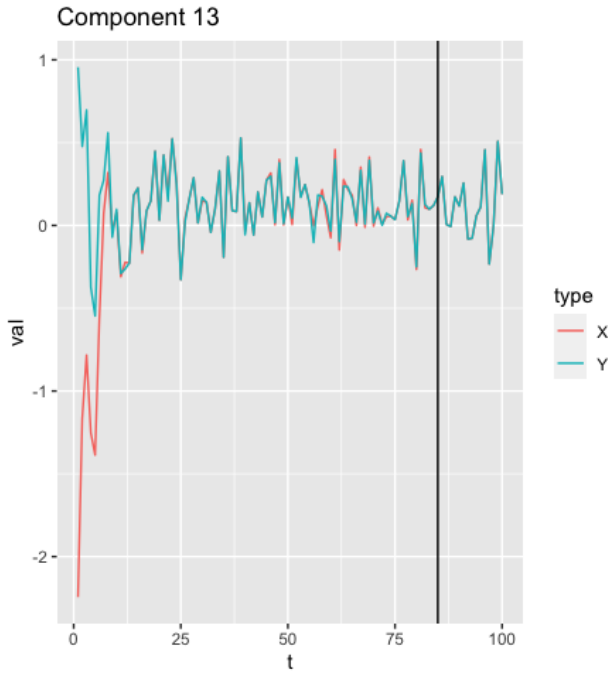
(c) Component 1



(d) Component 5



(e) Component 8



(f) Component 13

Circular Coupling Plots