Circularly-Coupled Markov Chain Sampling

Radford M. Neal

Dept. of Statistics and Dept. of Computer Science
University of Toronto

http://www.cs.utoronto.ca/∼radford/
radford@stat.utoronto.ca

April 2000

Markov Chain Sampling

We wish to sample values x_t from some complex distribution, π.

We can't easily sample independently from π. Instead, we construct a Markov chain that leaves π invariant, and that will converge to π if run for long enough.

The transitions of the Markov chain are simulated on the computer as follows:

Standard Markov chain simulation:

1) Randomly draw x_0 from the initial state distribution, p_0.

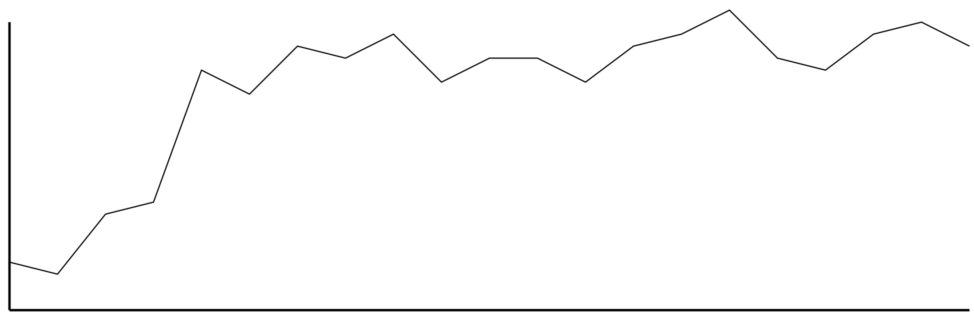
2) For $t = 1, \ldots, N$:

   Draw u_{t-1} from the distribution U, independently of previous draws.
   Let $x_t = \phi(x_{t-1}, u_{t-1})$.

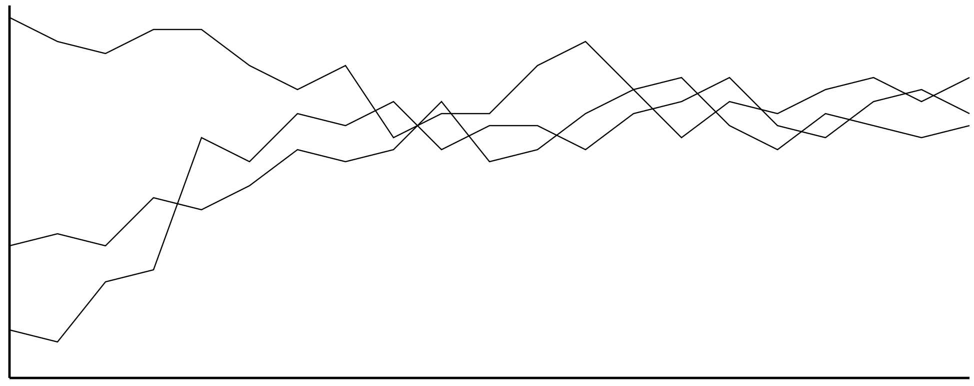
Problem: Diagnosing Convergence

One big problem with Markov chain sampling is how to determine if we've run the chain long enough to get close to the right distribution.

We could look at just one chain:



Or we could look at several:

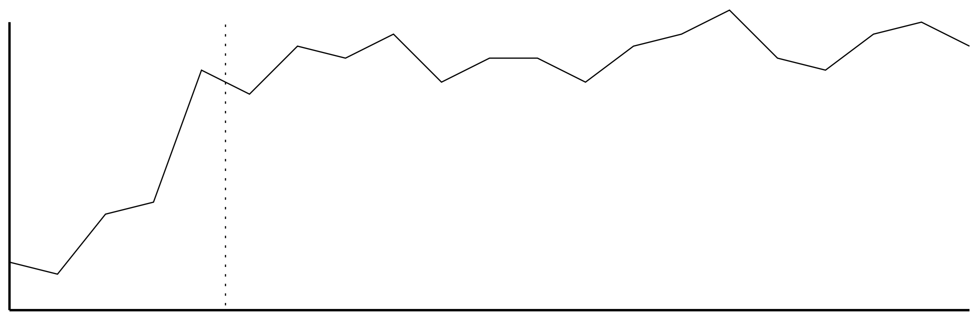


But we can't be sure what we see is the real asymptotic behaviour.

Problem: Discarding Burn-in Iterations

Assuming we think we *have* run long enough, how do we decide how much of the early part of the chain to discard as "burn-in" iterations?

We could judge by eye:



Or we could use some automatic procedure. But both techniques may introduce biases.

Problem: Simulating Chains in Parallel

Markov chain simulation appears to be an inherently sequential process — to generate x_{t+1}, we first must have generated x_t.

Parallelism can be exploited when simulating many independent realizations, and perhaps in the detailed computations needed for a single transition.

But can we also somehow bypass the apparent sequential nature of the simulation process?

Total Variation Distance and the Coupling Inequality

The total variation distance between two probability measures, μ and ν, is defined as the largest difference in the probabilities they assign to events:

$$d_{tv}(\mu,\nu) \ = \ \sup_A |\mu(A) - \nu(A)|$$

The coupling inequality can be used to bound this. If $X \sim \mu$ and $Y \sim \nu$, not necessarily independently, then

$$d_{tv}(\mu,\nu) \ \leq \ P(X \neq Y)$$

Proof:
$$\begin{aligned} P(X \neq Y) \ &\geq \ P(X \in A \text{ and } Y \notin A) \\ &\geq \ P(X \in A) \ - \ P(Y \in A) \\ &= \ \mu(A) \ - \ \nu(A) \end{aligned}$$

Similarly, $P(X \neq Y) \ \geq \ \nu(A) \ - \ \mu(A)$, hence $P(X \neq Y) \ \geq \ |\mu(A) \ - \ \nu(A)|$, for all A.

Coupling Markov Chains

To exploit the coupling inequality, we need to introduce appropriate dependencies. We can do this for two realizations of a Markov chain by using the same random numbers for both.

For instance, $\{X_t\}$ and $\{Y_t\}$ may be defined by

$$x_t = \phi(x_{t-1}, u_{t-1}), \quad y_t = \phi(y_{t-1}, u_{t-1})$$

with the random numbers, u_t, in common.

The initial states, x_0 and y_0, might come independently from the same or different distributions.

The same Markov chain can be produced using many different combinations of distributions for the random u_t and transition functions ϕ. Different schemes can produce different dependencies between the coupled chains.

We hope for rapid *coalescence*, where $X_t = Y_t$.

Uses of Coupling

1. Prove theoretical bounds on convergence rates (eg, Rosenthal 1995).

   Suppose $X_0 \sim p_0$ but $Y_0 \sim \pi$, so $Y_t \sim \pi$ for all t. If we can show $P(X_t \neq Y_t) \leq \epsilon$ for some t, then we know X_t has converged to within ϵ of π.

2. Investigate convergence rates empirically.

   Johnson (1996) simulates several coupled chains from dispersed initial states. The time for them all to coalesce is indicative of the time needed for convergence.

3. Exact sampling using "coupling from the past" (Propp & Wilson 1996).

   Requires not only a coupling scheme, but also a method for keeping track of large sets of states.

Basic Circular Coupling Algorithm

1) Randomly draw x_0 from the initial state distribution, p_0.

2) For $t = 1, \ldots, N$:

   Draw u_{t-1} from the distribution U, independently of previous draws.

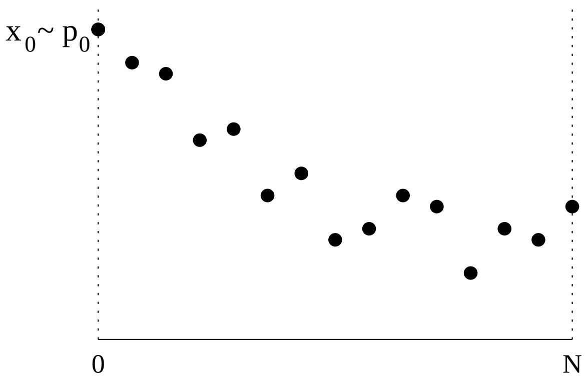
   Let $x_t = \phi(x_{t-1}, u_{t-1})$.

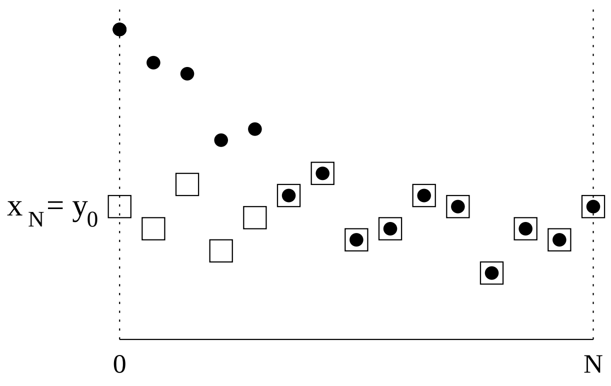
3) Let $y_0 = x_N$.

4) For $t = 1, \ldots, N$ while $y_{t-1} \neq x_{t-1}$:

   Let $y_t = \phi(y_{t-1}, u_{t-1})$.

5) Let the remaining y_t be the same as the corresponding x_t.





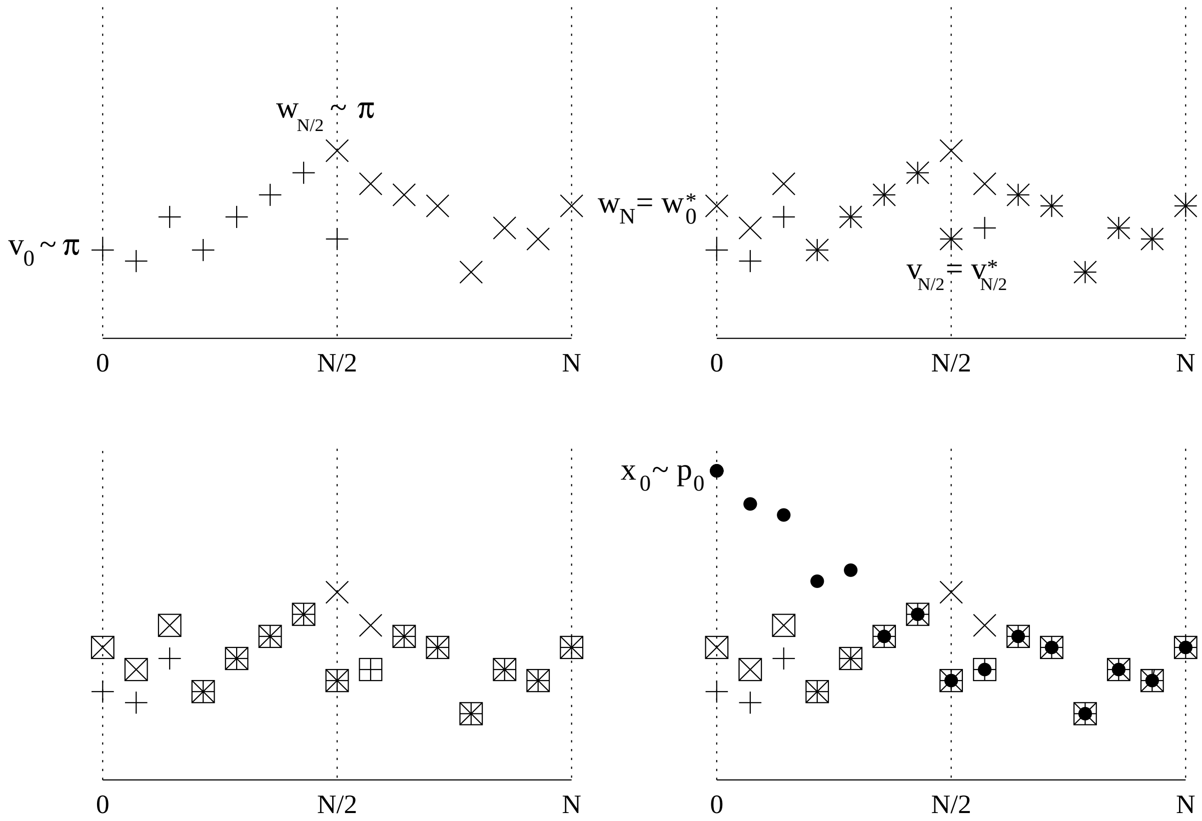
Theorem on Approximate Correctness

Each of the points $y_0, \ldots, y_N$ generated by a circularly-coupled Markov chain simulation has a distribution that is within $2\epsilon + \delta$ of the equilibrium distribution, π, in total variation distance, provided ϵ and δ are such that the following conditions on coupled chains hold:

1) If two chains are started from states drawn from π, independently of each other, and of the u_t, they will coalesce within $N/2$ iterations with probability at least $1 - \epsilon$.

2) If one chain is started from a state drawn from π, independently of the u_t, and another is started from a state drawn from p_0, independently of the initial state of the other chain and of the u_t, then the two chains will coalesce within N iterations with probability at least $1 - \delta$.

Proof of the Theorem



The top left shows generation of $v_0, \ldots, v_{N/2}$ and $w_{N/2}, \ldots, w_N$ starting from states drawn from π. The top right shows the continuation of these sequences, $v_{N/2} = v^*_{N/2}, \ldots, v^*_N$ and $w_N = w^*_0, \ldots, w^*_{N/2}$. In the bottom left, the sequence $y_0, \ldots, y_N$ used for estimation is marked. The bottom right shows a sequence $x_0, \ldots, x_N$ started from p_0 coalescing with the sequence $y_0, \ldots, y_N$, permitting this sequence to be found without the need to sample from π.

Using Auxiliary Chains to Check the Assumptions Needed for Correctness

1-5) Perform steps (1) to (5) of the basic circular coupling procedure.

6) Let c_0 be the number of steps needed for the wrapped-around chain to coalesce with the original chain.

7) For $i = 1, \ldots, r-1$:

    Let $s = iN/r$.

    Randomly draw $z_{i,s}$ from p_0.

    Set $c_i = 0$.

    For $t = s+1, \ldots, s+k$ (modulo N) while $z_{i,t-1} \neq y_{t-1}$:

        Let $z_{i,t} = \phi(z_{i,t-1}, u_{t-1})$.

        Set $c_i = c_i + 1$.

The coalescence times, $c_0, \ldots, c_{r-1}$, are indicative of whether condition (2) holds. If almost all c_i are less than $N/2$, we also have reason to think that condition (1) holds.

A Parallel Simulation Method

In parallel, processors numbered $i = 0, \ldots, r-1$ do the following:

1) Let $s = iN/r$.

2) For $t = s, \ldots, s + N/r - 1$:

   Randomly draw u_t from the distribution U, independently of other draws.

3) Randomly draw y_s from the distribution p_0, independently of other draws.

4) For $t = s + 1, \ldots, s + N/r - 1$:

   Set $y_t = \phi(y_{t-1}, u_{t-1})$.

5) Set $z = \phi(y_{s+N/r-1}, u_{s+N/r-1})$.

6) Send z to processor $i+1$ (modulo r) as the new value for $y_{s+N/r}$.

7) Repeat the following:

   Wait for a new value for y_s to be received from processor $i-1$ (modulo r).

   For $t = s+1, \ldots, s+N/r-1$ while $y_t \neq \phi(y_{t-1}, u_{t-1})$:

      Set $y_t = \phi(y_{t-1}, u_{t-1})$.

   If $z \neq \phi(y_{s+N/r-1}, u_{s+N/r-1})$:

      Set $z = \phi(y_{s+N/r-1}, u_{s+N/r-1})$.

      Send z to processor $i + 1$ (modulo r) as the new value for $y_{s+N/r}$.

Terminate when all processors are waiting.

A Strategy for Coupling Chains

For these algorithms to be useful in practice, we need a way of coupling good Markov chain samplers so as to produce rapid coalescence.

For continuous state spaces, we can use a hybrid strategy, in which two kinds of updates alternate:

1. Updates that are good at moving about the space, and will tend to bring chains closer and closer together.

2. Specialized updates that can produce *exact* coalescence once the states in two chains have come close together.

I'll look at Langevin updates for (1) and a specialized Metropolis method for (2).

Hypercube Metropolis Updates

The Metropolis algorithm updates state x_t by proposing a candidate state, x^*, which is accepted with probability $\min(1, \pi(x^*)/\pi(x_t))$. If accepted, $x_{t+1} = x^*$; otherwise $x_{t+1} = x_t$.

A proposal distribution, $g(x^*|x_t)$, is needed, which must be symmetrical: $g(x'|x) = g(x|x')$.

One possibility is for x^* to be uniformly distributed over a hypercube with sides of width w, centred on the current state.

Coupling for Hypercube Metropolis

The most obvious implementation of these transitions (in one dimension) is to generate

$$x^* \;=\; x_t + w(u_{1t} - 1/2)$$

and then accept if $u_{2t} < \min(1, \pi(x^*)/\pi(x_t))$, where $u_{kt} \sim \text{Uniform}(0,1)$.

But this cannot produce exact coalescence with non-zero probability.

Instead, we can generate

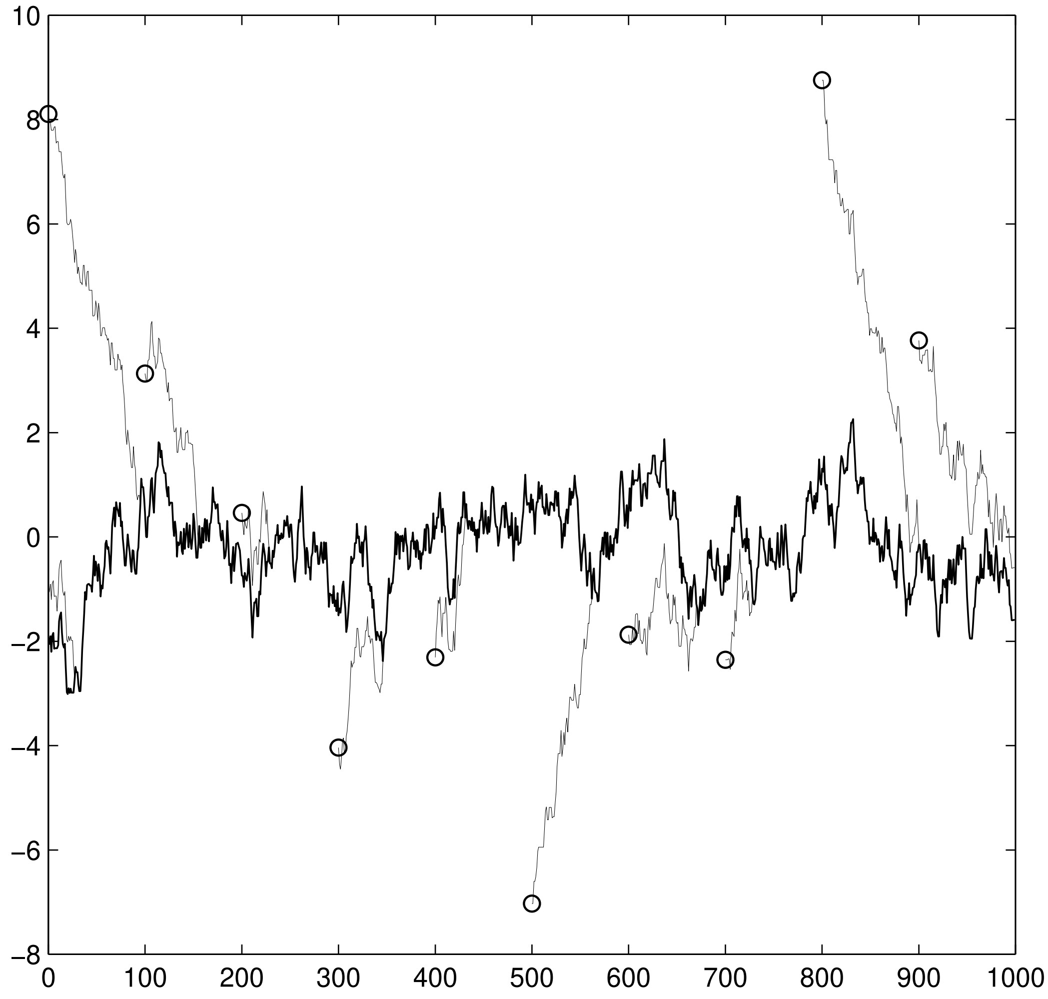
$$x^* \;=\; x_t + w[(u_{1t} - 1/2) + \text{Round}(x_t/w - (u_{1t} - 1/2))]$$

In any number of dimensions, we can visualize this as randomly positioning a grid of hypercubes, then letting x^* be the centre of the hypercube containing the current state, x_t.

Exact coalescence occurs if the states of two chains are in the same hypercube.

Example: Univariate Normal

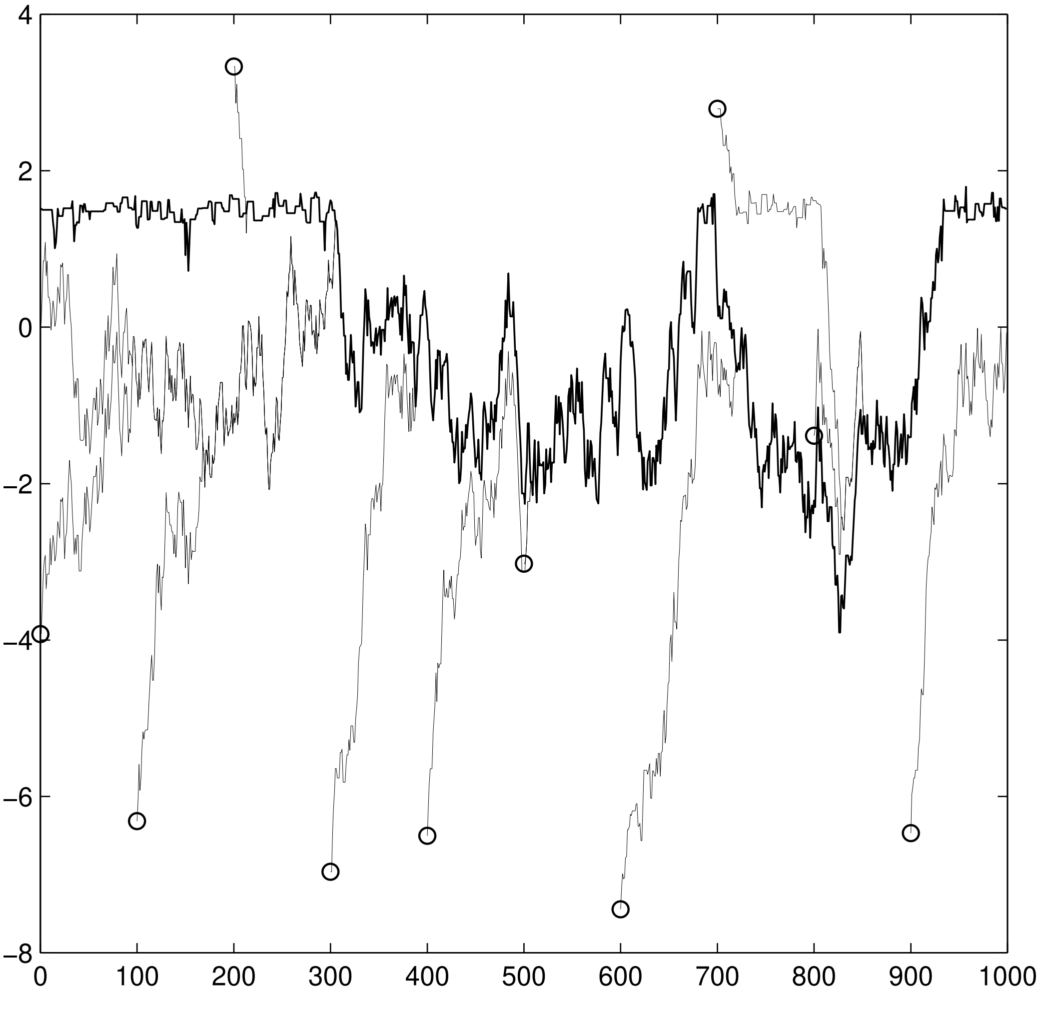
Parallel circularly-coupled sampling from $N(0,1)$, with 10 initial states from $N(0,5^2)$, using 100×10 iterations of a hypercube Metropolis update with $w = 1$.



Example: Univariate Normal Mixture

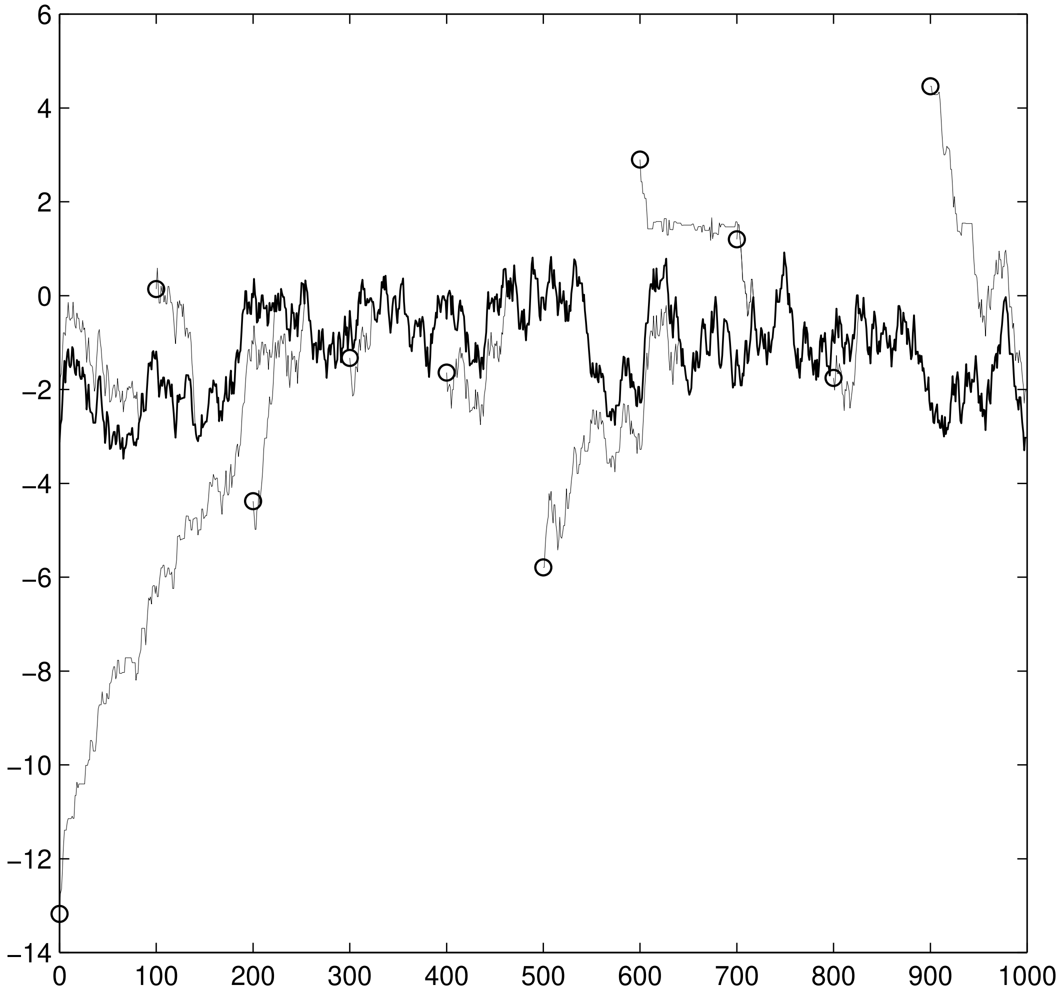
Same, but sampling from the mixture

$$(3/4)N(-1,1) \ + \ (1/4)N(1.5, 0.1^2)$$



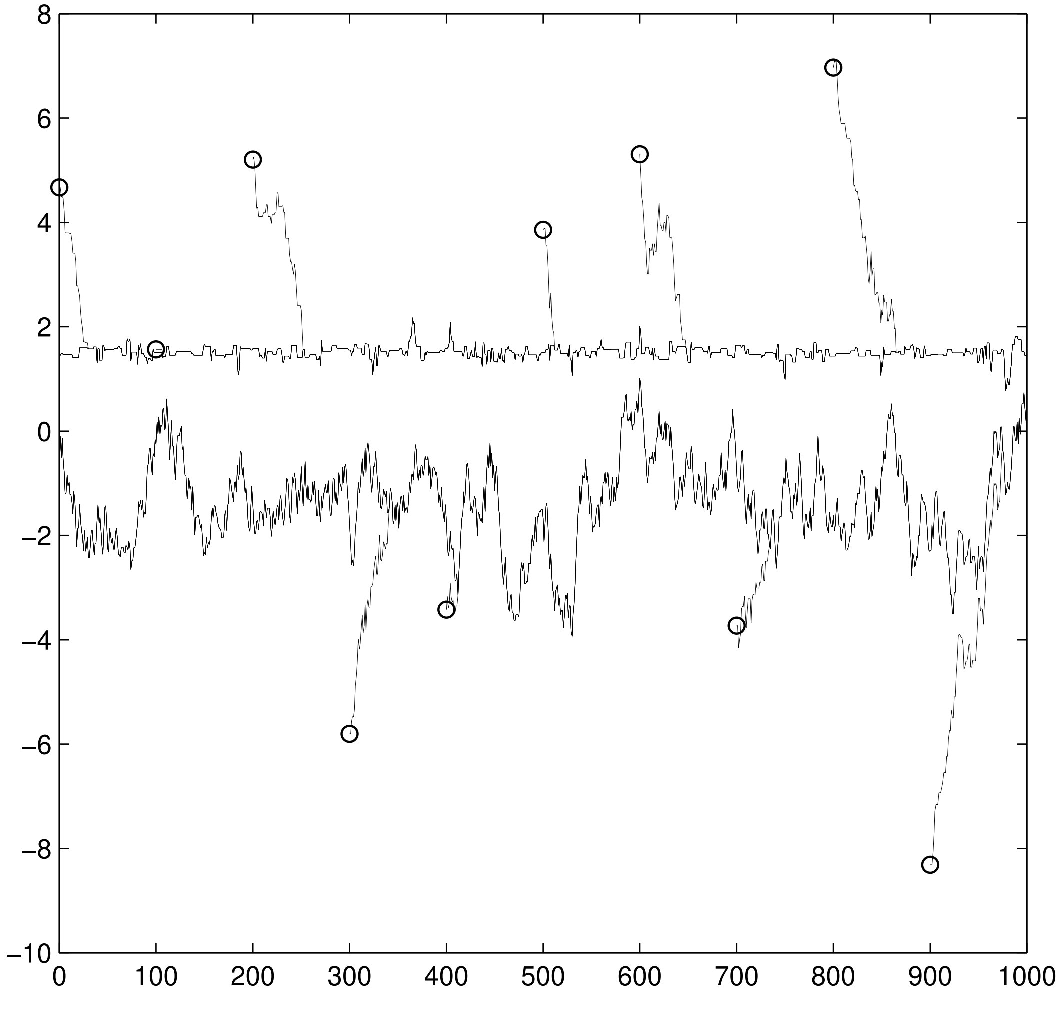
A Failure to Sample Both Modes

Same, but with a different random seed, that leads to only the lower mode being sampled, without any clear indication of a problem.



A Failure to Coalesce

Same, but with a different random seed, that results in the chains not coalescing.



Coupled Langevin Updates

For more difficult problems, hypercube Metropolis is probably not very good on its own. We can try combining it with another method.

In a Langevin update, the proposed state is

$$x^* \;=\; x_t + (\epsilon^2/2)\nabla \log \pi(x_t) + \epsilon u_t$$

where $u_t \sim N(0, I)$. The stepsize, ϵ, is chosen as large as possible while keeping acceptance high.

If we couple Langevin updates in the obvious way, chains tend to get closer and closer, but not coalesce exactly. We can fix that with an occasional hypercube Metropolis update.

Example: Logistic Regression

Model:

$$P(y_i{=}1 \mid x_i, \beta) = \left[\,1 + \exp(-\,(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_5 x_{i5}))\,\right]^{-1}$$

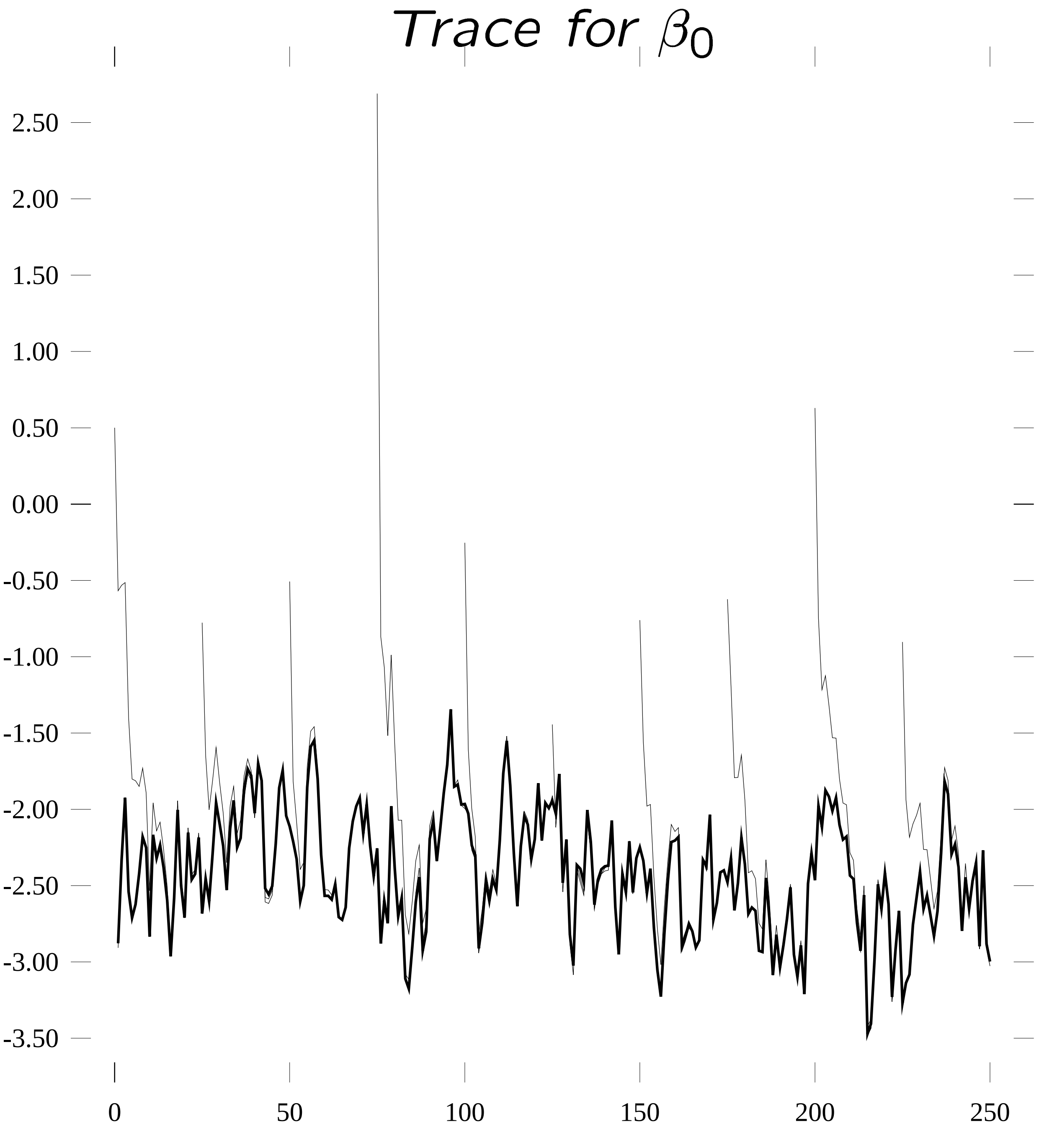
Prior: $\beta \ \sim \ N(0, I)$

Data: 500 cases generated with $\beta_0 = -2.5$, $\beta_1 = 0.5$, $\beta_2 = 0$, $\beta_3 = 1.0$, $\beta_4 = 0$, $\beta_5 = -0.2$, and with x_i generated independently from $N(\mu, \Sigma)$, with $\mu = [1, 2, 2, 2, 0]$ and

$$\Sigma \ = \ \begin{bmatrix} 1.04 & 1 & 1 & 1 & 0 \\ 1 & 1.01 & 1.01 & 1.01 & 0 \\ 1 & 1.01 & 1.02 & 1.01 & 0 \\ 1 & 1.01 & 1.01 & 1.02 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

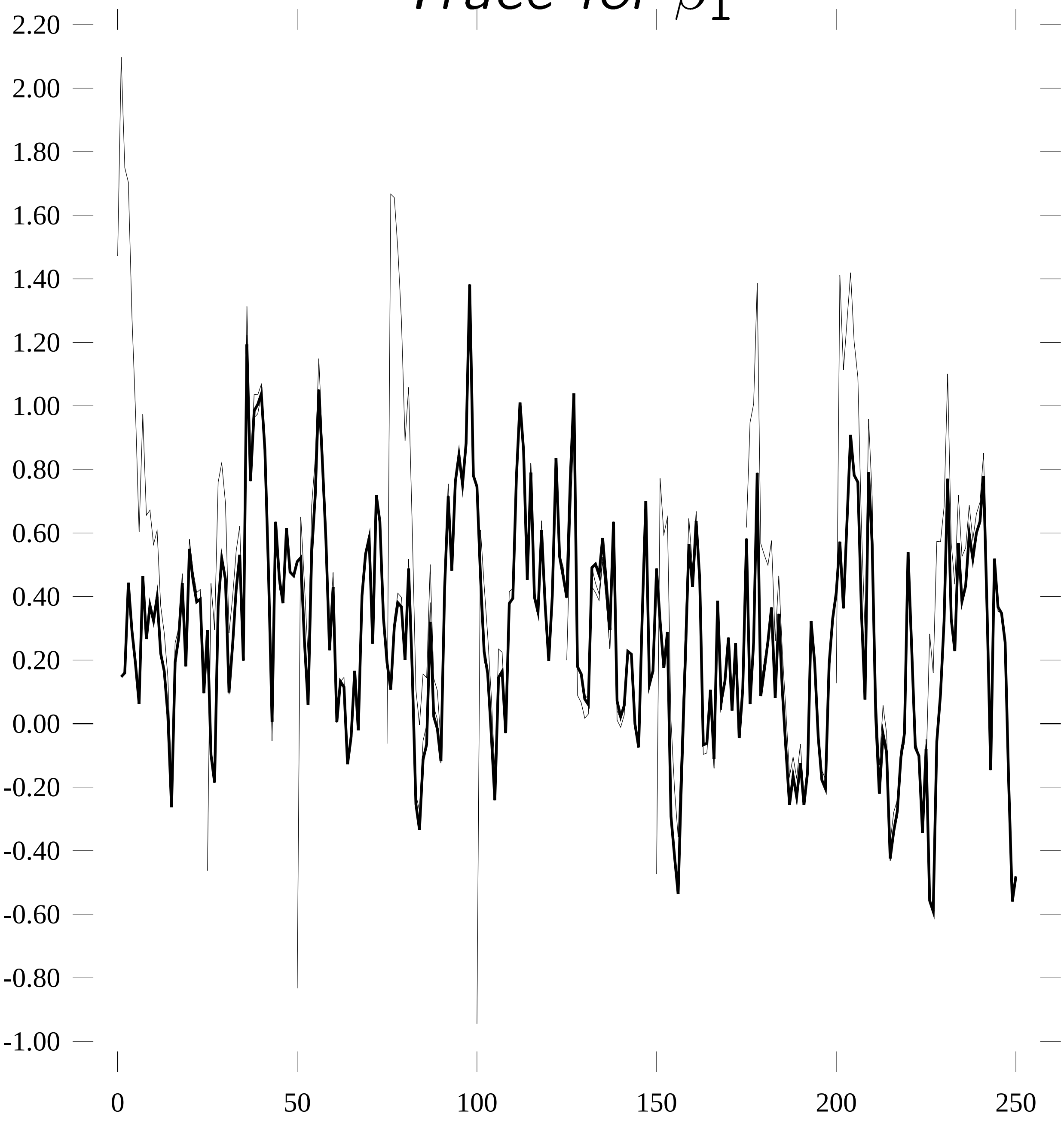
Sampling method: Parallel circularly-coupled simulation with 10 initial states from the prior. 25×10 iterations, each consisting of 200 Langevin updates ($\epsilon = 0.03$) followed by one hypercube Metropolis update ($w = 0.03$).

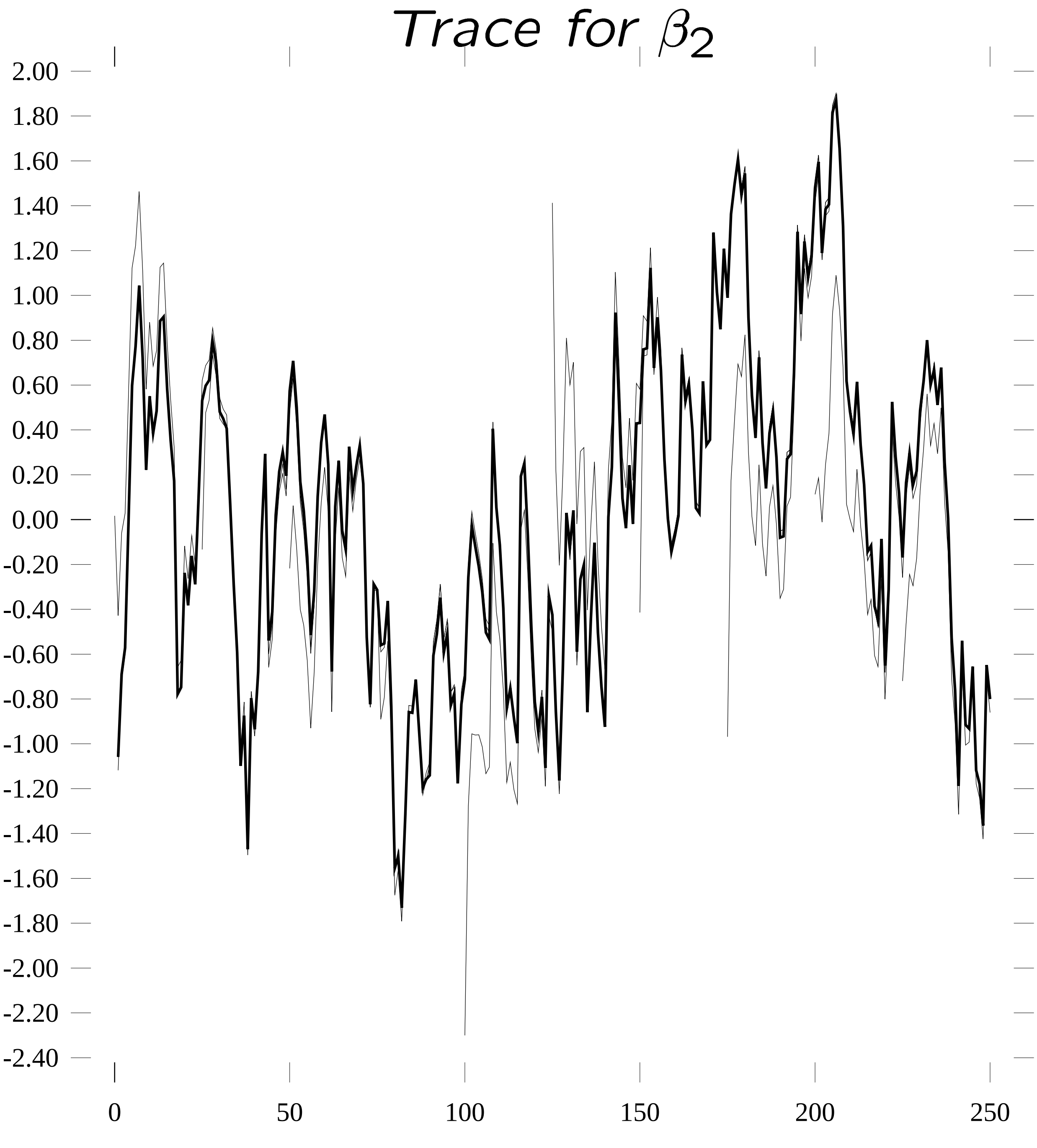
Coalescence occurred after 6 stages (36 runs).



Trace for β_0
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
-2.00
-2.50
-3.00
-3.50
0
50
100
150
200
250

Trace for β_1




Trace for β_2
2.00
1.80
1.60
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
-1.20
-1.40
-1.60
-1.80
-2.00
-2.20
-2.40
0
50
100
150
200
250

Discussion

- Circular coupling *may* be able to replace current convergence diagnostics and burn-in elimination methods.

- The theoretical savings are only a factor of two at most — the main benefit is from ease of use in practice.

- Achieving these benefits requires good coupling schemes. Need more research on schemes for hybrid Monte Carlo, Gibbs sampling, Metropolis, slice sampling, etc.

- Parallel simulation is another benefit, but will most users have multiple processors?