MCMC Tuning in Practice

Initialisation, Step Size, Burn-in, Thinning, and the Curse of Dimensionality

Where we are in Chapter 5

- We can build MCMC algorithms (MH, Gibbs, MwG) that have the *posterior* as a stationary distribution.
- But two questions always show up in practice:
  1. **Where do we start the chain?** (initialisation)
  2. **How big should the proposal moves be?** (step size / proposal scale)
- There are no universal "magic" rules.
- What we *can* do: **diagnose** problems from *trace plots* and simple summaries.

Goal today: learn the core diagnostics (burn-in + thinning + step-size symptoms).

Definition: burn-in (warm-up) period

Definition (Burn-in / warm-up). The *burn-in period* is the number of early iterations discarded because the chain has not yet reached its stationary regime.

- In our setting, the stationary distribution is the posterior $\pi(\theta \mid y)$.
- Early states can be dominated by the arbitrary starting value $\theta^{(0)}$.
- We often discard the first B samples:

$$\{\theta^{(B+1)}, \theta^{(B+2)}, \dots, \theta^{(N)}\}.$$

Important: you usually decide B *after* looking at a trace plot.

Definition: thinning parameter

Definition (Thinning). The *thinning parameter* k is the interval between stored iterations:

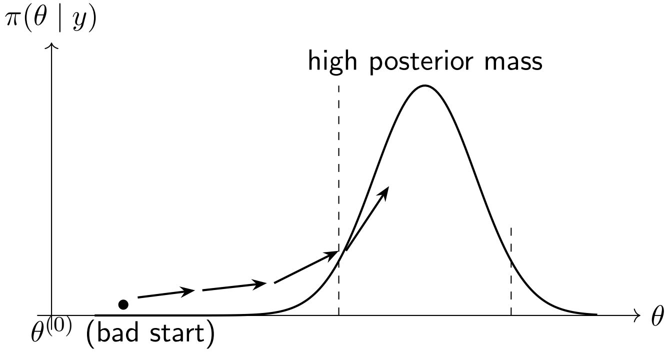
$$\text{store } \theta^{(k)}, \theta^{(2k)}, \theta^{(3k)}, \ldots$$

- Motivation 1: reduce storage (very long runs, many parameters).
- Motivation 2: reduce autocorrelation in the stored output.
- Reality check:
  - Thinning *does not create new information*; it discards samples.
  - If compute is cheap, it is often better to keep everything and compute effective sample size (ESS) later.

Rule of thumb for this module: thinning is mainly a *practical storage choice* and a *teaching tool* for correlation.

Intuition: starting in the wrong place

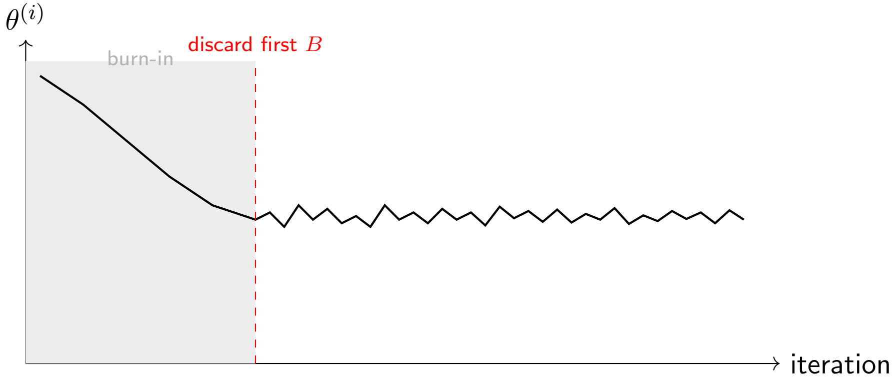
We want samples from a posterior like this:



- If $\theta^{(0)}$ is far from where $\pi(\theta \mid y)$ has mass, early steps are mostly “travel”.
- Those iterations are *not representative* of the posterior.

Trace plot story: the chain “travels” then mixes

A typical trace plot with burn-in looks like this:



- After burn-in, we want a “hairy caterpillar” / “white noise around a level”.
- Burn-in is chosen by inspecting when the chain stabilises into its typical region.

Example reference: starting at the wrong mean

Typical situation (e.g. a Normal mean model).

- You initialise at $\mu^{(0)} = 1$.
- The posterior for μ is centred nearer $\mu \approx 0.5$.
- The first few hundred iterations show a drift from 1 down to 0.5.

Practical action:

- Choose a burn-in B that removes the drift region.
- Then compute posterior summaries using $\{\mu^{(B+1)}, \dots, \mu^{(N)}\}$.

Key message: burn-in is about removing dependence on the arbitrary starting point.

The other tuning knob: step size

In random-walk Metropolis(-within-Gibbs), the proposal scale controls how far you jump.

Random-walk normal proposal (common):

$$\theta' = \theta^{(i-1)} + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

- σ^2 is the **step size** (proposal variance).
- If σ is too large: most proposals land in low-density regions $\Rightarrow$ many rejections.
- If σ is too small: almost everything is accepted, but the chain moves slowly and explores poorly.

Step size interacts with burn-in: if the chain moves too slowly, it takes much longer to even reach the typical region.

Acceptance probability reminder (why scale matters)

For Metropolis–Hastings:

$$\alpha(\theta^{(i-1)}, \theta') = \min\left\{1,\ \frac{\pi(\theta' \mid y)\, q(\theta^{(i-1)} \mid \theta')}{\pi(\theta^{(i-1)} \mid y)\, q(\theta' \mid \theta^{(i-1)})}\right\}.$$

- For *symmetric* random-walk proposals $q(\theta' \mid \theta) = q(\theta \mid \theta')$:

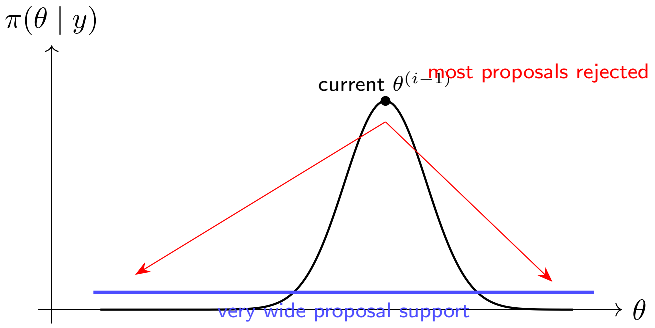
$$\alpha = \min\left\{1, \frac{\pi(\theta' \mid y)}{\pi(\theta^{(i-1)} \mid y)}\right\}.$$

- If θ' is far away in a low-density region, $\pi(\theta' \mid y)$ is tiny $\Rightarrow$ reject.

So: proposal scale decides how often you propose “reasonable” moves.

Case 1: step size too large

Suppose the posterior mass is concentrated, but your proposal is extremely wide.

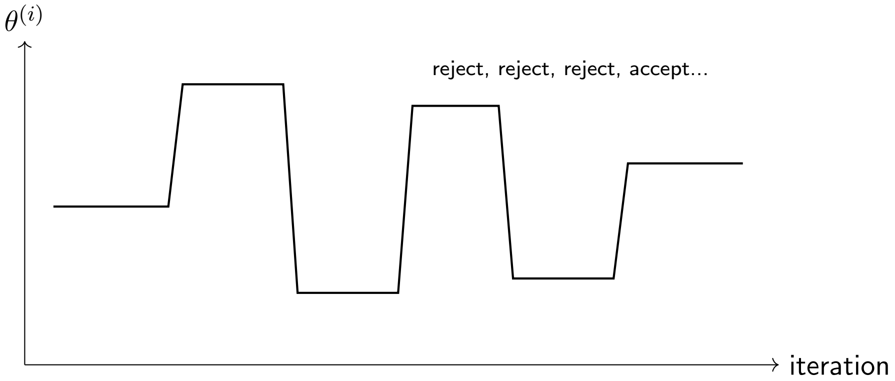


- Most proposed values land where $\pi(\theta' \mid y)$ is tiny.
- Acceptance rate becomes very small.
- The chain repeats the same value many times $\Rightarrow$ long flat segments in trace plot.

Trace plot symptom: step size too large

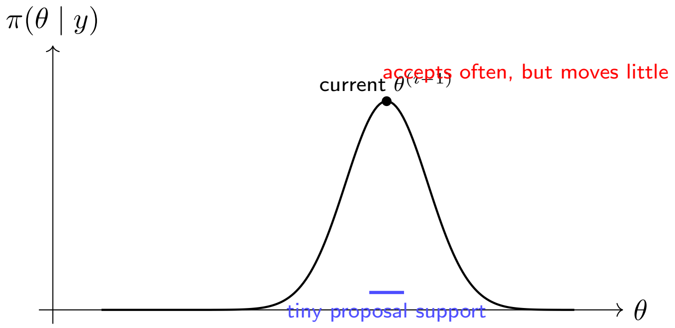
What you often see:

- Long **flat stretches** (many rejections in a row).
- Occasional **big jumps** when something finally gets accepted.
- Sometimes **wild oscillation** between extremes if accepted jumps overshoot.



Case 2: step size too small

Now the proposal is extremely narrow around the current value.

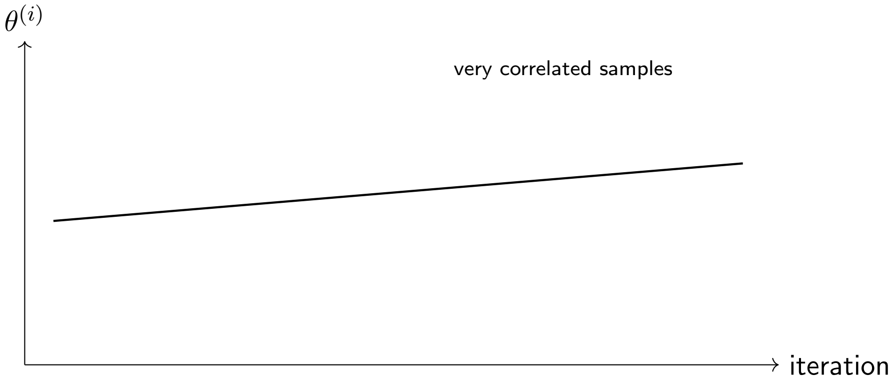


- You propose very nearby values, typically still in moderate/high density.
- Acceptance rate is high.
- But the chain **random-walks slowly** $\Rightarrow$ strong autocorrelation.

Trace plot symptom: step size too small

What you often see:

- The chain **creeps**.
- It can stay in one region for ages before moving to another region.
- It does not look like a nice “hairy caterpillar”.



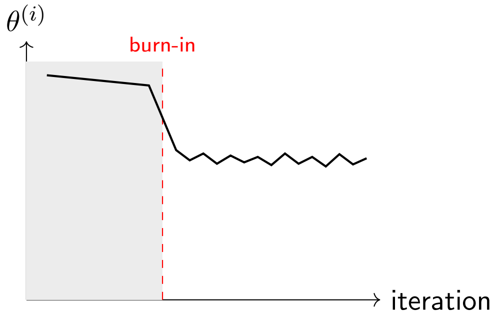
Burn-in and step size interact

If step size is too small:

- chain takes tiny steps
- reaching the high-density region can take a long time
- burn-in can become huge

If step size is too large:

- chain rejects almost everything
- it may take ages to accept a move into the high-density region
- burn-in can also become huge



Bottom line: poor tuning wastes computation twice: slow convergence + inefficient exploration.

Thinning: when does it make sense?

Recall: thinning stores only every k-th iteration.

- **Storage constraint:** if you run 10^8 iterations and saving everything is impossible.
- **Heavy autocorrelation:** if the chain barely moves, consecutive samples look almost the same.

But be careful:

- If you can afford it, keep the full chain and compute diagnostics later.
- Thinning can hide problems: a poorly mixing chain may *look* better after thinning.

: thinning is a pragmatic tool, not a cure.

What you can actually do in practice (trial and error)

There is no universal “Goldilocks” value of σ that works for all problems.

Practical workflow:

1. Pick a reasonable starting value (data-based guess, MLE, prior mean/mode).
2. Run a short pilot chain.
3. Inspect trace plots and acceptance rate.
4. Adjust σ:
   - stuck + flat segments $\Rightarrow$ decrease σ
   - slow creeping $\Rightarrow$ increase σ
5. Re-run longer once it looks like a “hairy caterpillar” after burn-in.

This is normal: tuning is an experimental loop.

MCMC is often inefficient

- MCMC is powerful because it lets us sample from complicated posteriors.
- But in high dimensions it can become **painfully slow**.
- Intuition: in high dimension, geometry becomes weird:
  - “most volume is near the boundary”
  - “typical points are far from the mode”

Volume of an n-dimensional ball (hypersphere) and where the mass lives.

Volume of an n-dimensional ball

Let $B_n(R)$ be an n-dimensional ball of radius R.

$$\mathrm{Vol}(B_n(R)) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)} R^n.$$

Now consider a smaller ball inside it with radius $R_1 < R$:

$$\mathrm{Vol}(B_n(R_1)) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)} R_1^n.$$

Ratio (inner volume / outer volume):

$$\frac{\mathrm{Vol}(B_n(R_1))}{\mathrm{Vol}(B_n(R))} = \left(\frac{R_1}{R}\right)^n.$$

Most of the volume is in the outer shell

The **proportion of volume outside** the inner ball is

$$P_{\text{shell}} = 1 - \left(\frac{R_1}{R}\right)^n .$$

Example. Take $n = 10$, $R = 1$, $R_1 = 1/2$:

$$P_{\text{shell}} = 1 - \left(\frac{1}{2}\right)^{10} = 1 - \frac{1}{1024} \approx 0.9990.$$

- About **99.9%** of the volume is *outside* radius $1/2$.
- The “central region” is tiny in volume as dimension grows.

Geometric punchline: in high dimension, “typical” points live in a thin outer shell.

Why this hurts Metropolis–Hastings

Think about a posterior in high dimension:

- There may be a mode (peak) and a region near the mode where density is high.
- But a huge amount of probability mass may live in regions far from the mode (the geometry/typical-set effect).

Random-walk MH difficulty:

- If you propose moves that are too bold: you jump into very low density $\Rightarrow$ reject.
- If you propose moves that are too timid: you stay correlated and explore slowly.
- In high dimension, balancing these becomes harder: **acceptance tends to drop unless you shrink step sizes,** which slows exploration.

So: tuning becomes more delicate, and mixing can collapse as n grows.

Beyond basic MCMC (just pointers)

Two important ideas that go beyond "vanilla" MH/Gibbs:

1. **Sequential Monte Carlo (SMC):**
   - maintains a set of particles approximating the posterior
   - can update the approximation as new data arrives
   - avoids restarting the whole inference from scratch each time
2. **Hamiltonian Monte Carlo (HMC):**
   - uses **gradient information** of $\log \pi(\theta \mid y)$
   - proposals follow approximate Hamiltonian dynamics
   - often far more efficient in moderate/high dimensions

Message: MCMC is not "one algorithm" — there is a big toolbox when basic random-walk MH struggles.

Wrap-up

- **Initialisation:** try data-based guesses, but often you are “flying blind”.
- **Burn-in:** discard early iterations while the chain travels to typical regions.
- **Step size:** diagnose from trace plots:
  - too large $\Rightarrow$ many rejections, flat segments, occasional jumps
  - too small $\Rightarrow$ creeping, high autocorrelation, slow exploration
- **Thinning:** mostly for storage; not a cure for poor mixing.
- **Curse of dimensionality:** high dimension makes random-walk proposals increasingly inefficient.

If your chain doesn’t look like a hairy caterpillar after burn-in, something is wrong.