Gibbs Sampling and Metropolis-within-Gibbs

Where we are in the course

- We spent several sessions on **Metropolis–Hastings (MH)**:
  - construct a Markov chain by *propose* $\rightarrow$ *accept/reject*
  - under the usual conditions (irreducible, aperiodic, correct invariant distribution), the chain targets the posterior.
- Today: **Gibbs sampling**
  - often *simpler* and *more efficient*
  - but works in **specific** cases: when full conditionals are easy to sample from.

Big picture: two extremes and the middle

Suppose we want samples from a posterior $\pi(\boldsymbol{\theta} \mid y)$, with $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$.

Extreme 1: Metropolis–Hastings (generic)

- Works for essentially any posterior *we can evaluate up to a constant*.
- But may be inefficient: many proposals can be rejected.

Extreme 2: Gibbs (special but powerful)

- Requires **closed-form full conditional distributions** that we can sample from.
- When available, updates are always accepted (no reject step).

In practice: Metropolis-within-Gibbs (the middle)

Some parameters have easy conditionals $\Rightarrow$ Gibbs; others do not $\Rightarrow$ MH updates.

Setup: posterior with many parameters

Assume a model with parameters

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n), \qquad \text{data } y, \qquad \text{posterior } \pi(\theta_1, \ldots, \theta_n \,|\, y).$$

Setup: posterior with many parameters

Assume a model with parameters

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n), \qquad \text{data } y, \qquad \text{posterior } \pi(\theta_1, \ldots, \theta_n \,|\, y).$$

The key extra ingredient for Gibbs sampling is the collection of **full conditional distributions**:

$$\pi(\theta_1 \,|\, \theta_2, \ldots, \theta_n, y),\ \pi(\theta_2 \,|\, \theta_1, \theta_3, \ldots, \theta_n, y),\ \ldots,\ \pi(\theta_n \,|\, \theta_1, \ldots, \theta_{n-1}, y).$$

Setup: posterior with many parameters

Assume a model with parameters

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n), \qquad \text{data } y, \qquad \text{posterior } \pi(\theta_1, \ldots, \theta_n \,|\, y).$$

The key extra ingredient for Gibbs sampling is the collection of **full conditional distributions**:

$$\pi(\theta_1 \,|\, \theta_2, \ldots, \theta_n, y),\ \pi(\theta_2 \,|\, \theta_1, \theta_3, \ldots, \theta_n, y),\ \ldots,\ \pi(\theta_n \,|\, \theta_1, \ldots, \theta_{n-1}, y).$$

Reminder: what is a full conditional?

Start from the joint posterior $\pi(\theta_1, \ldots, \theta_n \,|\, y)$ and **keep only terms that involve the parameter of interest**. Everything that does not depend on that parameter can be absorbed into the proportionality constant.

Why full conditionals are so convenient

If each full conditional has a standard form (Normal, Gamma, Beta, etc.), then:

- we can sample from it directly with a built-in RNG,
- we avoid tuning proposals / acceptance rates,
- every update is accepted (no MH rejection step).

Why full conditionals are so convenient

If each full conditional has a standard form (Normal, Gamma, Beta, etc.), then:

- we can sample from it directly with a built-in RNG,
- we avoid tuning proposals / acceptance rates,
- every update is accepted (no MH rejection step).

But: Gibbs is not always available

If one or more full conditionals do *not* have a tractable closed form (or are hard to sample from), then we cannot run a pure Gibbs sampler.

The Gibbs sampler: algorithm (general case)

Let $\boldsymbol{\theta}^{(i)} = (\theta_1^{(i)}, \ldots, \theta_n^{(i)})$ denote the Markov chain state at iteration i.

Gibbs sampling (one sweep)

1. **Initialise:** choose $\boldsymbol{\theta}^{(0)}$ (any reasonable starting values).
2. For $i = 1, 2, \ldots$:
   1. Sample
   $$\theta_1^{(i)} \sim \pi\Big(\theta_1 \,\Big|\, \theta_2^{(i-1)}, \ldots, \theta_n^{(i-1)}, y\Big).$$
   2. Sample
   $$\theta_2^{(i)} \sim \pi\Big(\theta_2 \,\Big|\, \theta_1^{(i)}, \theta_3^{(i-1)}, \ldots, \theta_n^{(i-1)}, y\Big).$$
   3. Continue similarly:
   $$\theta_j^{(i)} \sim \pi\Big(\theta_j \,\Big|\, \theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)}, y\Big).$$
   4. Finally sample
   $$\theta_n^{(i)} \sim \pi\Big(\theta_n \,\Big|\, \theta_1^{(i)}, \ldots, \theta_{n-1}^{(i)}, y\Big).$$

The Gibbs sampler: algorithm (general case)

Let $\boldsymbol{\theta}^{(i)} = (\theta_1^{(i)}, \ldots, \theta_n^{(i)})$ denote the Markov chain state at iteration i.

Gibbs sampling (one sweep)

1. **Initialise:** choose $\boldsymbol{\theta}^{(0)}$ (any reasonable starting values).
2. For $i = 1, 2, \ldots$:
   1. Sample
   $$\theta_1^{(i)} \sim \pi\Big(\theta_1 \,\Big|\, \theta_2^{(i-1)}, \ldots, \theta_n^{(i-1)}, y\Big).$$
   2. Sample
   $$\theta_2^{(i)} \sim \pi\Big(\theta_2 \,\Big|\, \theta_1^{(i)}, \theta_3^{(i-1)}, \ldots, \theta_n^{(i-1)}, y\Big).$$
   3. Continue similarly:
   $$\theta_j^{(i)} \sim \pi\Big(\theta_j \,\Big|\, \theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}, \theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)}, y\Big).$$
   4. Finally sample
   $$\theta_n^{(i)} \sim \pi\Big(\theta_n \,\Big|\, \theta_1^{(i)}, \ldots, \theta_{n-1}^{(i)}, y\Big).$$

Key idea: "most recent values"

When sampling $\theta_j^{(i)}$:

- parameters already updated in this sweep use iteration i:

$$\theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}$$

- parameters not yet updated still use iteration $i-1$:

$$\theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)}.$$

Key idea: "most recent values"

When sampling $\theta_j^{(i)}$:

- parameters already updated in this sweep use iteration i:

$$\theta_1^{(i)}, \ldots, \theta_{j-1}^{(i)}$$

- parameters not yet updated still use iteration $i-1$:

$$\theta_{j+1}^{(i-1)}, \ldots, \theta_n^{(i-1)}.$$

Why do we do this?

Because it makes each update condition on the *current* state of the chain. This produces a valid Markov chain with the desired invariant distribution under standard conditions.

Order of updates: does it matter?

- You can update in order $1 \to 2 \to \cdots \to n$ (most common in teaching).
- You can also update in a different fixed order, or even a random-scan order.

Order of updates: does it matter?

- You can update in order $1 \to 2 \to \cdots \to n$ (most common in teaching).
- You can also update in a different fixed order, or even a random-scan order.

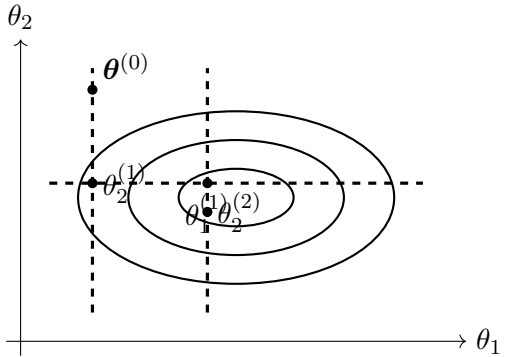
Main message

The target posterior does not change (it is determined by the model, prior, and data). The **efficiency** (mixing speed, autocorrelation, etc.) can change depending on the order.

Geometric intuition (2D case)

For $n = 2$, the posterior is a joint density $\pi(\theta_1, \theta_2 \mid y)$ on the plane. Gibbs sampling alternates between:

- sampling θ_2 given the current θ_1 (vertical conditional slice),
- sampling θ_1 given the current θ_2 (horizontal conditional slice).



What you should take from the picture

- Each move is easy: sample from a 1D conditional distribution.
- Over many iterations, the chain explores the 2D joint density.
- Gibbs can mix *slowly* if variables are highly correlated (movement is axis-aligned, which can "zig-zag").

What you should take from the picture

- Each move is easy: sample from a 1D conditional distribution.
- Over many iterations, the chain explores the 2D joint density.
- Gibbs can mix *slowly* if variables are highly correlated (movement is axis-aligned, which can "zig-zag").

So why do we still love it?

When full conditionals are available and reasonably well-behaved, Gibbs is simple to implement and can be very efficient per iteration.

Example: hierarchical exponential model (recall)

We observe claim times $Y_1, \ldots, Y_n$:

$$Y_i \,\big|\, \lambda \sim \text{Exponential}(\lambda), \quad i = 1, \ldots, n.$$

Example: hierarchical exponential model (recall)

We observe claim times $Y_1, \ldots, Y_n$:

$$Y_i \,\big|\, \lambda \sim \text{Exponential}(\lambda), \quad i = 1, \ldots, n.$$

We put a prior on λ:

$$\lambda \,\big|\, \gamma \sim \text{Exponential}(\gamma).$$

Example: hierarchical exponential model (recall)

We observe claim times $Y_1, \ldots, Y_n$:

$$Y_i \,\big|\, \lambda \sim \text{Exponential}(\lambda), \quad i = 1, \ldots, n.$$

We put a prior on λ:

$$\lambda \,\big|\, \gamma \sim \text{Exponential}(\gamma).$$

And a hyperprior on γ:

$$\gamma \,\big|\, \mu \sim \text{Exponential}(\mu), \qquad \text{with } \mu \text{ fixed (chosen by us).}$$

Example: hierarchical exponential model (recall)

We observe claim times $Y_1, \ldots, Y_n$:

$$Y_i \,\big|\, \lambda \sim \text{Exponential}(\lambda), \quad i = 1, \ldots, n.$$

We put a prior on λ:

$$\lambda \,\big|\, \gamma \sim \text{Exponential}(\gamma).$$

And a hyperprior on γ:

$$\gamma \,\big|\, \mu \sim \text{Exponential}(\mu), \qquad \text{with } \mu \text{ fixed (chosen by us).}$$

Goal

Sample from the joint posterior $\pi(\lambda, \gamma \,|\, y)$.

Step 1: write the posterior (up to proportionality)

Bayes:

$$\pi(\lambda, \gamma \,|\, y) \;\propto\; \pi(y \,|\, \lambda)\, \pi(\lambda \,|\, \gamma)\, \pi(\gamma).$$

Step 1: write the posterior (up to proportionality)

Bayes:

$$\pi(\lambda, \gamma \,|\, y) \;\propto\; \pi(y \,|\, \lambda)\, \pi(\lambda \,|\, \gamma)\, \pi(\gamma).$$

Likelihood (Exponential):

$$\pi(y \,|\, \lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda y_i} = \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} y_i \right).$$

Step 1: write the posterior (up to proportionality)

Bayes:

$$\pi(\lambda, \gamma \,|\, y) \;\propto\; \pi(y \,|\, \lambda)\, \pi(\lambda \,|\, \gamma)\, \pi(\gamma).$$

Likelihood (Exponential):

$$\pi(y \,|\, \lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda y_i} = \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} y_i \right).$$

Prior on λ:

$$\pi(\lambda \,|\, \gamma) = \gamma e^{-\gamma \lambda}.$$

Step 1: write the posterior (up to proportionality)

Bayes:

$$\pi(\lambda, \gamma \,|\, y) \;\propto\; \pi(y \,|\, \lambda)\, \pi(\lambda \,|\, \gamma)\, \pi(\gamma).$$

Likelihood (Exponential):

$$\pi(y \,|\, \lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda y_i} = \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} y_i \right).$$

Prior on λ:

$$\pi(\lambda \,|\, \gamma) = \gamma e^{-\gamma \lambda}.$$

Hyperprior on γ:

$$\pi(\gamma) = \mu e^{-\mu \gamma}.$$

Step 1: write the posterior (up to proportionality)

Bayes:

$$\pi(\lambda, \gamma \,|\, y) \;\propto\; \pi(y \,|\, \lambda)\, \pi(\lambda \,|\, \gamma)\, \pi(\gamma).$$

Likelihood (Exponential):

$$\pi(y \,|\, \lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda y_i} = \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} y_i \right).$$

Prior on λ:

$$\pi(\lambda \,|\, \gamma) = \gamma e^{-\gamma \lambda}.$$

Hyperprior on γ:

$$\pi(\gamma) = \mu e^{-\mu \gamma}.$$

Multiply:

$$\pi(\lambda, \gamma \,|\, y) \propto \lambda^n \exp\Big(-\lambda \sum y_i \Big)\; \gamma e^{-\gamma \lambda}\; \mu e^{-\mu \gamma}.$$

Step 2: derive full conditional for λ

We keep only terms involving λ in the joint posterior:

$$\pi(\lambda \,|\, \gamma, y) \propto \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} y_i\right) \exp(-\gamma\lambda).$$

Step 2: derive full conditional for λ

We keep only terms involving λ in the joint posterior:

$$\pi(\lambda \,|\, \gamma, y) \propto \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} y_i\right) \exp(-\gamma\lambda).$$

Combine exponent terms:

$$\pi(\lambda \,|\, \gamma, y) \propto \lambda^n \exp\left(-\lambda\Big(\sum_{i=1}^{n} y_i + \gamma\Big)\right).$$

Step 2: derive full conditional for λ

We keep only terms involving λ in the joint posterior:

$$\pi(\lambda \,|\, \gamma, y) \propto \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} y_i\right) \exp(-\gamma\lambda).$$

Combine exponent terms:

$$\pi(\lambda \,|\, \gamma, y) \propto \lambda^n \exp\left(-\lambda\Big(\sum_{i=1}^{n} y_i + \gamma\Big)\right).$$

Distribution matching

This is a Gamma distribution (shape–rate form):

$$\lambda \,|\, \gamma, y \sim \text{Gamma}\Big(\alpha = n+1,\ \beta = \sum_{i=1}^{n} y_i + \gamma\Big),$$

where density is proportional to $\lambda^{\alpha-1} e^{-\beta\lambda}$

Step 3: derive full conditional for γ

Keep only terms involving γ:

$$\pi(\gamma \mid \lambda, y) \propto \gamma\, e^{-\gamma\lambda}\, e^{-\mu\gamma}.$$

Step 3: derive full conditional for γ

Keep only terms involving γ:

$$\pi(\gamma \mid \lambda, y) \propto \gamma\, e^{-\gamma\lambda}\, e^{-\mu\gamma}.$$

Combine exponent:

$$\pi(\gamma \mid \lambda, y) \propto \gamma \exp(-\gamma(\lambda + \mu)).$$

Step 3: derive full conditional for γ

Keep only terms involving γ:

$$\pi(\gamma \mid \lambda, y) \propto \gamma\, e^{-\gamma\lambda}\, e^{-\mu\gamma}.$$

Combine exponent:

$$\pi(\gamma \mid \lambda, y) \propto \gamma \exp(-\gamma(\lambda + \mu)).$$

Distribution matching

This is also Gamma (shape–rate):

$$\gamma \mid \lambda, y \sim \text{Gamma}\big(\alpha = 2,\ \beta = \lambda + \mu\big).$$

(If you prefer, it is a special case of Gamma rather than a plain exponential, because of the extra leading factor γ.)

Summary: full conditionals in this example

Let $S = \sum_{i=1}^{n} y_i$.

Full conditionals

$$\lambda \,|\, \gamma, y \sim \text{Gamma}\big(n+1,\ S+\gamma\big), \qquad \gamma \,|\, \lambda, y \sim \text{Gamma}\big(2,\ \lambda+\mu\big),$$

(shape–rate parameterisation).

Summary: full conditionals in this example

Let $S = \sum_{i=1}^{n} y_i$.

Full conditionals

$$\lambda \,|\, \gamma, y \sim \text{Gamma}\big(n+1,\ S+\gamma\big), \qquad \gamma \,|\, \lambda, y \sim \text{Gamma}\big(2,\ \lambda+\mu\big),$$

(shape–rate parameterisation).

- Both are easy to sample from $\Rightarrow$ **pure Gibbs is possible**.
- No accept/reject required.

Gibbs sampler for the hierarchical exponential model

Algorithm

1. Initialise $\lambda^{(0)}$ and $\gamma^{(0)}$.
2. For $i = 1, 2, \ldots$:
   1. Sample $\lambda^{(i)} \sim \text{Gamma}\big(n+1,\ S+\gamma^{(i-1)}\big)$.
   2. Sample $\gamma^{(i)} \sim \text{Gamma}\big(2,\ \lambda^{(i)}+\mu\big)$.

Gibbs sampler for the hierarchical exponential model

Algorithm

1. Initialise $\lambda^{(0)}$ and $\gamma^{(0)}$.
2. For $i = 1, 2, \ldots$:
   1. Sample $\lambda^{(i)} \sim \text{Gamma}\big(n+1,\ S+\gamma^{(i-1)}\big)$.
   2. Sample $\gamma^{(i)} \sim \text{Gamma}\big(2,\ \lambda^{(i)}+\mu\big)$.

- Notice the "most recent values" rule: γ uses $\lambda^{(i)}$, not $\lambda^{(i-1)}$.

Implementation mindset: store and update

In code, you almost always:

- choose number of iterations T,
- create vectors to store draws, e.g. `lambda[1:T]`, `gamma[1:T]`,
- run a loop and overwrite/update at each iteration.

Implementation mindset: store and update

In code, you almost always:

- choose number of iterations T,
- create vectors to store draws, e.g. `lambda[1:T]`, `gamma[1:T]`,
- run a loop and overwrite/update at each iteration.

Why "`i-1`" matters

When updating $\lambda^{(i)}$ you need $\gamma^{(i-1)}$ because $\gamma^{(i)}$ has not been sampled yet.

R pseudocode (shape–rate parameterisation)

```r
T <- 10000
lambda <- numeric(T)
gamma <- numeric(T)

# data summaries
n <- length(y)
S <- sum(y)

# hyperparameter (chosen by us)
mu <- 0.01

# initialise
lambda[1] <- 0.5
gamma[1] <- 0.5

for (i in 2:T) {
  # 1) update lambda | gamma, y
  lambda[i] <- rgamma(1, shape = n + 1, rate = S + gamma[i-1])

  # 2) update gamma | lambda, y
  gamma[i] <- rgamma(1, shape = 2, rate = lambda[i] + mu)
}
```

Diagnostics: trace plots and posterior summaries

After running the chain:

- **Trace plot** (hairy caterpillar): does it look stationary and well-mixed?
- **Burn-in**: discard initial iterations if needed.
- **Posterior density**: histogram / KDE of stored samples.
- **Posterior mean, credible intervals**, etc.

Diagnostics: trace plots and posterior summaries

After running the chain:

- **Trace plot** (hairy caterpillar): does it look stationary and well-mixed?
- **Burn-in**: discard initial iterations if needed.
- **Posterior density**: histogram / KDE of stored samples.
- **Posterior mean, credible intervals**, etc.

Typical workflow:

$$\{\lambda^{(i)}\}_{i=1}^{T} \Rightarrow \text{estimate } \mathbb{E}[\lambda \,|\, y],\ \mathrm{Var}(\lambda \,|\, y),\ \mathrm{CI}_{95\%}, \ldots$$

and similarly for γ.

Looking at the joint posterior: scatter/contours

One of the nicest things in Bayesian inference:

- you do not only get point estimates,
- you get **the full joint distribution** (or an approximation via samples).

Looking at the joint posterior: scatter/contours

One of the nicest things in Bayesian inference:

- you do not only get point estimates,
- you get **the full joint distribution** (or an approximation via samples).

With Gibbs samples $(\lambda^{(i)}, \gamma^{(i)})$ you can:

- make a scatter plot,
- add a contour plot / 2D density estimate,
- study dependence: *if* λ *is around* 0.25*, what range of* γ *is plausible?*

Gibbs vs MH: same target, different efficiency

Important concept

The posterior $\pi(\boldsymbol{\theta} \,|\, y)$ is determined by the model and the prior. **It does not depend on the sampling algorithm.**

Gibbs vs MH: same target, different efficiency

Important concept

The posterior $\pi(\boldsymbol{\theta} \mid y)$ is determined by the model and the prior. **It does not depend on the sampling algorithm.**

- Gibbs and MH (when correctly constructed) target the *same* posterior.
- But:
  - MH can reject many proposals $\Rightarrow$ more wasted iterations.
  - Gibbs always moves (it draws from the conditional) $\Rightarrow$ often cheaper per effective sample.

Gibbs vs MH: same target, different efficiency

Important concept

The posterior $\pi(\boldsymbol{\theta} \mid y)$ is determined by the model and the prior. **It does not depend on the sampling algorithm.**

- Gibbs and MH (when correctly constructed) target the *same* posterior.
- But:
  - MH can reject many proposals $\Rightarrow$ more wasted iterations.
  - Gibbs always moves (it draws from the conditional) $\Rightarrow$ often cheaper per effective sample.

Caveat

Gibbs can still mix slowly in high correlation settings (zig-zag behaviour). Efficiency is about *effective sample size*, not just iteration count.

A brief preview: using geometry/gradients (advanced MCMC)

Basic MH proposals are often random-walk style:

$$\theta' = \theta^{(i-1)} + \varepsilon.$$

They ignore where the high posterior mass is.

A brief preview: using geometry/gradients (advanced MCMC)

Basic MH proposals are often random-walk style:

$$\theta' = \theta^{(i-1)} + \varepsilon.$$

They ignore where the high posterior mass is.
In many problems, you can do better by using local information:

- move proposals toward regions of higher posterior density,
- incorporate gradients (and sometimes curvature) of $\log \pi(\theta \mid y)$.

A brief preview: using geometry/gradients (advanced MCMC)

Basic MH proposals are often random-walk style:

$$\theta' = \theta^{(i-1)} + \varepsilon.$$

They ignore where the high posterior mass is.
In many problems, you can do better by using local information:

- move proposals toward regions of higher posterior density,
- incorporate gradients (and sometimes curvature) of $\log \pi(\theta \mid y)$.

Takeaway

There are many MCMC variants designed to reduce burn-in and improve mixing. In this course, we focus on the two core workhorses: MH and Gibbs.

Metropolis-within-Gibbs: motivation

Pure Gibbs requires all full conditionals to be easy to sample from.

Metropolis-within-Gibbs: motivation

Pure Gibbs requires all full conditionals to be easy to sample from.
But in real models:

- some parameters have conjugate full conditionals (easy),
- others have messy conditionals (no closed form).

Metropolis-within-Gibbs: motivation

Pure Gibbs requires all full conditionals to be easy to sample from.
But in real models:

- some parameters have conjugate full conditionals (easy),
- others have messy conditionals (no closed form).

Idea

Update what you can with Gibbs, and use MH for the rest.

Metropolis-within-Gibbs: generic algorithm

Suppose $\theta_1, \ldots, \theta_k$ have tractable full conditionals, but $\theta_{k+1}, \ldots, \theta_n$ do not.

One sweep at iteration i

1. **Gibbs updates:** for $j = 1, \ldots, k$,

$$\theta_j^{(i)} \sim \pi(\theta_j \,|\, \text{rest}, y).$$

2. **MH updates:** for $j = k+1, \ldots, n$,
   - propose $\theta_j' \sim q_j(\cdot \,|\, \theta_j^{(i-1)})$,
   - accept with the appropriate MH probability targeting the *full conditional* $\pi(\theta_j \,|\, \text{rest}, y)$.
3. Repeat for $i = 1, 2, \ldots$.

Metropolis-within-Gibbs: acceptance probability (one component)

When updating component θ_j (holding all others fixed at their current values), we want a Markov step that leaves the full conditional invariant:

$$\pi(\theta_j \,|\, \theta_{-j}, y), \qquad \theta_{-j} = (\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_n).$$

Metropolis-within-Gibbs: acceptance probability (one component)

When updating component θ_j (holding all others fixed at their current values), we want a Markov step that leaves the full conditional invariant:

$$\pi(\theta_j \,|\, \theta_{-j}, y), \qquad \theta_{-j} = (\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_n).$$

If we propose $\theta_j' \sim q_j(\cdot \,|\, \theta_j)$, then accept with

$$\alpha = \min\left\{1,\ \frac{\pi(\theta_j' \,|\, \theta_{-j}, y)\ q_j(\theta_j \,|\, \theta_j')}{\pi(\theta_j \,|\, \theta_{-j}, y)\ q_j(\theta_j' \,|\, \theta_j)}\right\}.$$

Metropolis-within-Gibbs: acceptance probability (one component)

When updating component θ_j (holding all others fixed at their current values), we want a Markov step that leaves the full conditional invariant:

$$\pi(\theta_j \,|\, \theta_{-j}, y), \qquad \theta_{-j} = (\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_n).$$

If we propose $\theta_j' \sim q_j(\cdot \,|\, \theta_j)$, then accept with

$$\alpha = \min\left\{1,\ \frac{\pi(\theta_j' \,|\, \theta_{-j}, y)\ q_j(\theta_j \,|\, \theta_j')}{\pi(\theta_j \,|\, \theta_{-j}, y)\ q_j(\theta_j' \,|\, \theta_j)}\right\}.$$

- This is just MH, but the target is the **conditional** distribution.
- Everything else is treated as fixed during this sub-step.

Practical message: "mix and match" freely

You do *not* have to update all Gibbs-first then MH-last. You can interleave:

- θ_1 via MH,
- θ_2 via Gibbs,
- θ_3 via MH,
- θ_4 via Gibbs,
- etc.

Practical message: "mix and match" freely

You do *not* have to update all Gibbs-first then MH-last. You can interleave:

- θ_1 via MH,
- θ_2 via Gibbs,
- θ_3 via MH,
- θ_4 via Gibbs,
- etc.

Core rule

Each sub-update must leave the desired conditional invariant, given the current values of the other parameters.

Wrap-up

- **Gibbs sampling**:
  - requires full conditionals with closed forms that are easy to sample from,
  - alternates sampling each parameter from its conditional given the latest values of others,
  - often efficient because there is no rejection step.
- **Hierarchical exponential example**:
  - compute posterior up to proportionality,
  - derive full conditionals by "keep only relevant terms",
  - recognise Gamma forms $\Rightarrow$ implement Gibbs in a few lines.
- **Metropolis-within-Gibbs**:
  - use Gibbs where possible, MH where necessary.