LECTURE 18: MCMC, COSMOMC, COBAYA, AUTOMATIC DIFFERENTIATION, AND HMC

GONG-BO ZHAO

Learning goals

After this lecture, students should be able to:

- write down Bayes' theorem for parameter inference and explain the roles of likelihood, prior, posterior, and evidence;
- describe the Metropolis-Hastings algorithm and explain why ordinary MCMC can become inefficient in high-dimensional or highly correlated posteriors;
- understand the practical logic of cosmology samplers such as CosmoMC and Cobaya, including multiple chains, convergence tests, and fast-slow parameter structure;
- explain what automatic differentiation is, and why it is different from both symbolic differentiation and finite-difference derivatives;
- derive the basic HMC equations and understand why gradient-based proposals can move much farther than random-walk proposals;
- implement a simple pedagogical BAO-only fit for $(\Omega_{\rm m}, H_0 r_d)$ and understand why DESI BAO contours rotate with redshift.

1. Why parameter sampling is unavoidable in cosmology

Modern cosmology is a forward-model problem. One specifies a parameter vector $\boldsymbol{\theta}$, computes theoretical predictions for observables such as C_ℓ, $P(k)$, supernova distance moduli, BAO distance ratios, lensing correlation functions, or cluster counts, and then compares these predictions to data. The forward model may involve a Boltzmann code such as CAMB, a survey-window convolution, nuisance parameters, and several likelihood modules. Even before model comparison, the central object is the posterior distribution

$$p(\boldsymbol{\theta}|d) = \frac{\mathcal{L}(d|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{\mathcal{E}(d)}, \qquad \mathcal{E}(d) = \int \mathrm{d}\boldsymbol{\theta}\,\mathcal{L}(d|\boldsymbol{\theta})\pi(\boldsymbol{\theta}), \tag{1}$$

where $\mathcal{L}$ is the likelihood, π is the prior, and $\mathcal{E}$ is the Bayesian evidence.

In two parameters one can still draw a dense grid and evaluate Eq. (1) directly. But in realistic cosmological analyses one usually has a much larger space: six base cosmological parameters, calibration parameters, foreground and instrumental nuisance parameters, light-curve or photo-z systematics, and sometimes hyperparameters controlling theoretical errors. A uniform grid with n points per dimension requires n^D evaluations in D dimensions, so even $n = 20$ and $D = 10$ would already demand 10^{13} model evaluations. That

is impossible when a single likelihood evaluation may involve CAMB or CLASS plus a complicated survey likelihood.

Sampling methods solve this problem by concentrating evaluations where the posterior is appreciable. This is why MCMC became the default language of cosmological parameter inference in the WMAP and Planck eras, and why codes like CosmoMC and later Cobaya became central infrastructure for the field [4, 5, 6, 7].

2. Metropolis-Hastings MCMC

The basic idea. A Markov chain is a sequence of samples $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \ldots$ in which the next point depends only on the current point. We want to construct the transition rule so that the stationary distribution of the chain is the target posterior $p(\boldsymbol{\theta}|d)$.

The Metropolis-Hastings algorithm does this in two steps [1, 2]:

(1) From the current point $\boldsymbol{\theta}$, draw a proposal $\boldsymbol{\theta}'$ from a proposal density $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$.
(2) Accept the proposal with probability

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left[1, \frac{p(\boldsymbol{\theta}'|d)\, q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{p(\boldsymbol{\theta}|d)\, q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right]. \tag{2}$$

If the proposal is rejected, repeat the current state in the chain.

If the proposal is symmetric, $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}')$, Eq. (2) simplifies to the familiar Metropolis ratio

$$\alpha = \min\left[1, \frac{p(\boldsymbol{\theta}'|d)}{p(\boldsymbol{\theta}|d)}\right]. \tag{3}$$

Since the evidence $\mathcal{E}(d)$ cancels in the ratio, MCMC can explore the posterior without knowing the normalization.

Why does it work? The key condition is detailed balance,

$$p(\boldsymbol{\theta}|d)\, T(\boldsymbol{\theta} \to \boldsymbol{\theta}') = p(\boldsymbol{\theta}'|d)\, T(\boldsymbol{\theta}' \to \boldsymbol{\theta}), \tag{4}$$

where T is the transition kernel. Metropolis-Hastings is constructed precisely so that detailed balance holds. If, in addition, the chain is irreducible and aperiodic, the posterior becomes the unique stationary distribution.

Strengths and weaknesses. The great virtue of random-walk Metropolis is robustness. It needs only log posterior values, not gradients, Hessians, or special geometric information. The weakness is equally important: it explores parameter space diffusively. If the posterior is narrow, curved, or highly correlated, then a random-walk proposal either takes tiny steps and mixes slowly, or takes large steps and gets rejected. In high dimensions this becomes increasingly painful.

This tension is familiar in cosmology. Parameters such as ω_b, ω_c, θ_*, τ, A_s, and n_s are strongly correlated in different ways across different data combinations; nuisance parameters can be extremely fast to evaluate compared to the cosmological parameters that

trigger a new Boltzmann solve; and survey likelihoods often involve large covariance matrices. These practical issues motivated much of the design of CosmoMC and later Cobaya [5, 6].

3. What makes a chain trustworthy?

A chain is useful only if it has actually explored the posterior. In classroom discussions students often focus too much on the acceptance rate and not enough on geometry and autocorrelation. Four ideas are more important.

First, one usually runs multiple chains from over-dispersed starting points. If all chains converge to the same region and show similar one-dimensional and two-dimensional marginals, that is much more reassuring than a single long run.

Second, one monitors autocorrelation. The integrated autocorrelation time is

$$\tau_{\rm int} = 1 + 2\sum_{t=1}^{\infty} \rho_t, \tag{5}$$

where ρ_t is the lag-t autocorrelation of the chain. The effective sample size is then approximately

$$N_{\rm eff} \simeq \frac{N}{\tau_{\rm int}}. \tag{6}$$

A chain with 10^5 points but $\tau_{\rm int} = 200$ only contains a few hundred effectively independent samples.

Third, one monitors between-chain consistency. The classic Gelman-Rubin statistic compares the variance between chains to the variance within each chain [3]. In cosmological practice one often quotes $\hat{R} - 1$ or a closely related variant; the DESI DR2 cosmology analysis, for example, uses a Gelman-Rubin stopping criterion together with an effective-sample-size requirement [21].

Fourth, one should look at trace plots and simple physical derived quantities, not just final contour plots. Many pathologies are obvious in traces long before they become obvious in a polished triangle plot.

4. CosmoMC

CosmoMC was introduced by Lewis and Bridle as a practical Monte Carlo engine for cosmological parameter inference [4]. In its current public form it is described as a parallelized Fortran 2008 MCMC sampler for general and cosmological applications, with GetDist included for chain analysis and plotting [12, 13]. Historically, CosmoMC was tightly coupled to CAMB and to the likelihood infrastructure that underpinned WMAP and Planck analyses.

For students, the most important conceptual points are these.

CosmoMC is not just "Metropolis in a loop". The original Lewis-Bridle paper already emphasized practical choices such as physically meaningful parameterizations, parallel chains, and importance sampling [4]. Later development made the sampler much more efficient when some parameters are computationally expensive ("slow") and others are cheap ("fast"). Antony Lewis' fast-slow paper explains how decorrelation and dragging moves can greatly improve efficiency when nuisance parameters are cheap but cosmological parameters require expensive theory calculations [5].

This matters because in a CMB or large-scale-structure likelihood, changing a nuisance parameter may only affect a foreground model or calibration term, while changing a cosmological parameter may require recomputing transfer functions or background distances. A sampler that treats all directions equally leaves a great deal of speed on the table.

The student-level workflow. At a practical level, the CosmoMC workflow is conceptually straightforward:

(1) define the cosmological model and likelihood combination;
(2) specify priors and proposal settings in configuration files;
(3) run multiple MPI chains in parallel;
(4) monitor convergence and combine chains;
(5) analyze the results with GetDist.

The details are software-specific, but the logic is the same as in any serious MCMC analysis.

Why students should still learn it. Even though new users are now often directed toward Cobaya rather than CosmoMC [12, 13], many ideas that students encounter in modern cosmology papers were shaped by CosmoMC practice: proposal covariance learning, multiple chains, Gelman-Rubin stopping, fast-slow block structure, and standard post-processing with GetDist.

5. Cobaya

Cobaya, by Torrado and Lewis, is a more general and more modular Bayesian analysis framework [6]. The current documentation describes it as a framework for sampling and statistical modelling that can explore arbitrary priors or posteriors using several samplers, including the advanced MCMC sampler inherited from CosmoMC, and that supports MPI parallelization, modular likelihoods, and external theory codes such as CAMB and CLASS [7, 8, 9].

Why Cobaya feels modern. The most important change from a student perspective is that Cobaya makes the pipeline explicit. A Cobaya input is usually a YAML dictionary with top-level blocks such as

`likelihood`, `params`, `prior`, `sampler`, `theory`, `output`.

This makes it easier to read the logic of an analysis. One can inspect the likelihood block, see which theory code is being called, understand which parameters are sampled, and identify where a custom likelihood or prior should be added [8].

Cobaya also exposes the modularity that students need for research: one can swap CAMB for CLASS, change the likelihood set, add derived parameters, or point to a different external package path without rewriting the whole sampler. The official documentation also provides automatic installation of common cosmology requisites and currently lists internal DESI DR2 BAO likelihoods such as `bao.desi_dr2` [10, 11].

A minimal YAML skeleton. The following is a schematic BAO run structure that shows the logic students should recognize:

```
likelihood:
  bao.desi_dr2: null

theory:
  camb: null

params:
  ombh2:
    prior: {min: 0.020, max: 0.025}
  omch2:
    prior: {min: 0.08, max: 0.14}
  H0:
    prior: {min: 55, max: 85}
  As:
    prior: {min: 1.5e-9, max: 3.5e-9}
  ns:
    prior: {min: 0.90, max: 1.05}
  tau:
    prior: {min: 0.01, max: 0.12}

sampler:
  mcmc:
    Rminus1_stop: 0.01

output: chains/desi_dr2_demo
```

This is intentionally only a skeleton. A realistic production run needs model-specific choices, good reference points, and often a more informative prior structure. But the example makes the architecture clear: the theory code computes observables, the likelihood compares them to data, and the sampler explores the posterior.

CosmoMC versus Cobaya. The relationship between the two codes is best understood historically and practically. CosmoMC is the older cosmology-specific workhorse; Cobaya is the newer and more general framework. The official CosmoMC README explicitly notes that Cobaya incorporates a version of CosmoMC's sampler and most of its features, while providing more general speed optimization and support for multiple inter-dependent theory and likelihood codes [12]. For students, the cleanest summary is that learning Cobaya helps with modern workflow, while learning CosmoMC helps with the historical language of the literature and the logic of mature cosmology pipelines.

6. Automatic differentiation

What AD is, and what it is not. Automatic differentiation (AD) is not symbolic manipulation, and it is not finite differencing. Finite differencing approximates derivatives by repeated function evaluations, for example

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \tag{7}$$

which is easy but suffers from truncation and subtraction errors. Symbolic differentiation manipulates algebraic expressions. Automatic differentiation instead applies the chain rule exactly to the sequence of elementary operations executed by a program [14].

The key insight is that a numerical program is a composition of simple operations such as addition, multiplication, `exp`, `log`, `sin`, matrix multiplication, and so on. If we know how each elementary operation propagates derivatives, then the whole program can propagate derivatives.

Forward mode and dual numbers. Forward-mode AD can be introduced with dual numbers. Write

$$\hat{x} = x + \epsilon \dot{x}, \qquad \epsilon^2 = 0. \tag{8}$$

Then elementary functions propagate derivatives automatically:

$$\sin(x + \epsilon \dot{x}) = \sin x + \epsilon \cos x \, \dot{x}, \tag{9}$$

$$(x + \epsilon \dot{x})(y + \epsilon \dot{y}) = xy + \epsilon(x\dot{y} + y\dot{x}). \tag{10}$$

At the end of the computation, the coefficient of ϵ is the directional derivative. Julia's `ForwardDiff.jl` implements forward-mode AD using exactly this logic, with dual numbers and efficient vector-forward strategies [15, 16].

Reverse mode. If the target function is a scalar, reverse mode is often preferable in high dimensions because one backward sweep yields the whole gradient at a cost that is typically only a modest multiple of the original function evaluation. In machine-learning language, reverse-mode AD is what backpropagation exploits. In HMC this matters because the target is usually the scalar log posterior $\log p(\boldsymbol{\theta}|d)$, whose gradient with respect to many parameters is required at every trajectory step [14].

Why AD matters for cosmology. Hand-coding derivatives of a realistic cosmology likelihood is error-prone. Even a "simple" background-only analysis may involve numerical integrals, covariance matrices, priors on transformed parameters, and multiple derived distances. AD makes HMC practical because the sampler can ask for gradients of the log posterior without the user differentiating everything by hand.

7. Hamiltonian Monte Carlo

From diffusion to guided motion. Random-walk Metropolis proposes a local step and then asks whether to accept it. HMC instead augments the parameter vector $\boldsymbol{q}$ with an auxiliary momentum $\boldsymbol{p}$ and defines the Hamiltonian

$$H(\boldsymbol{q},\boldsymbol{p}) = U(\boldsymbol{q}) + K(\boldsymbol{p}), \qquad U(\boldsymbol{q}) = -\log p(\boldsymbol{q}|d), \qquad K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^T\mathbf{M}^{-1}\boldsymbol{p}, \tag{11}$$

where $\mathbf{M}$ is a positive-definite mass matrix [17, 18, 19]. Hamilton's equations are

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \frac{\partial H}{\partial \boldsymbol{p}} = \mathbf{M}^{-1}\boldsymbol{p}, \qquad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\frac{\partial H}{\partial \boldsymbol{q}} = -\nabla U(\boldsymbol{q}). \tag{12}$$

The gradient ∇U is exactly the place where AD enters.

The leapfrog integrator. Exact Hamiltonian flow preserves phase-space volume and conserves H. In practice we simulate the dynamics numerically. The standard choice is the leapfrog integrator:

$$\boldsymbol{p}\left(t+\tfrac{\epsilon}{2}\right) = \boldsymbol{p}(t) - \frac{\epsilon}{2}\nabla U\big(\boldsymbol{q}(t)\big), \tag{13}$$

$$\boldsymbol{q}(t+\epsilon) = \boldsymbol{q}(t) + \epsilon\,\mathbf{M}^{-1}\boldsymbol{p}\left(t+\tfrac{\epsilon}{2}\right), \tag{14}$$

$$\boldsymbol{p}(t+\epsilon) = \boldsymbol{p}\left(t+\tfrac{\epsilon}{2}\right) - \frac{\epsilon}{2}\nabla U\big(\boldsymbol{q}(t+\epsilon)\big). \tag{15}$$

This map is reversible and symplectic, so the accumulated Hamiltonian error stays controlled. After L leapfrog steps we apply a Metropolis correction,

$$\alpha = \min\left[1, \exp\big(-H(\boldsymbol{q}',\boldsymbol{p}') + H(\boldsymbol{q},\boldsymbol{p})\big)\right]. \tag{16}$$

Because the proposal follows the local geometry of the posterior, accepted moves can be long and efficient rather than diffusive.

Tuning and caveats. HMC introduces three practical choices: the step size ϵ, the trajectory length $L\epsilon$, and the mass matrix $\mathbf{M}$. If ϵ is too large, integration error causes rejection; if too small, the sampler wastes time. If L is too short, HMC degenerates toward a local method; if too long, it can waste computation or retrace its path. The No-U-Turn Sampler (NUTS) adaptively sets trajectory length and is often the easiest gradient-based algorithm to use in practice [20].

The most important conceptual warning for students is that HMC is not automatically superior in every cosmology setting. Hard parameter boundaries, discontinuous likelihood pieces, numerical noise, or expensive non-differentiable external codes can limit its usefulness. That is one reason why CosmoMC-style Metropolis sampling is still the default in much of production cosmology, even though HMC is mathematically elegant.

8. A short comparison of sampling strategies

Method	**Needs gradients?**	**Main strength**	**Main weakness**
Random-walk Metropolis	No	Very general and robust	Diffusive; inefficient for correlated high-dimensional posteriors
CosmoMC-style fast-slow MCMC	No	Exploits cosmology-specific speed hierarchy; excellent with nuisance blocks	Still fundamentally a random-walk method
HMC / NUTS	Yes	Long-distance proposals with high acceptance; scales well in smooth posteriors	Requires reliable gradients and careful geometry/tuning

9. Example: BAO-only fit to $(\Omega_{\rm m}, H_0 r_d)$ using DESI

Why this example is pedagogically clean. A BAO-only flat-ΛCDM example is excellent for teaching because the background equations are simple and the data constrain a very transparent parameter combination. As emphasized in the DESI DR2 cosmology paper, when fitting BAO data only one can sample over $\Omega_{\rm m}$ and the combination $H_0 r_d$ rather than H_0 and r_d separately [21]. Up to a factor of the speed of light, this is the same degeneracy often written as $H_0 r_d/c$.

A small notation warning is worth making explicit for students. In this BAO-only example the relevant matter-density parameter is $\Omega_{\rm m}$. The symbol w_0 belongs to dynamical dark-energy models and is a different parameter; DESI's BAO-only contour in this plane is not a $(w_0, H_0 r_d)$ fit.

For flat ΛCDM,

$$E(z) \equiv \frac{H(z)}{H_0} = \sqrt{\Omega_{\rm m}(1+z)^3 + 1 - \Omega_{\rm m}}. \tag{17}$$

The comoving angular-diameter distance and radial BAO distance are then

$$\frac{D_M(z)}{r_d} = \frac{c}{H_0 r_d} \int_0^z \frac{{\rm d}z'}{E(z')}, \tag{18}$$

$$\frac{D_H(z)}{r_d} = \frac{c}{H_0 r_d} \frac{1}{E(z)}. \tag{19}$$

For the low-redshift isotropic BGS point one instead uses

$$\frac{D_V(z)}{r_d} = \frac{c}{H_0 r_d} \left[\frac{z}{E(z)} \left(\int_0^z \frac{{\rm d}z'}{E(z')} \right)^2 \right]^{1/3}. \tag{20}$$

A classroom likelihood can therefore be written as

$$-2 \ln \mathcal{L} = [\mathbf{D} - \boldsymbol{t}(\Omega_{\rm m}, H_0 r_d)]^T \mathbf{C}^{-1} [\mathbf{D} - \boldsymbol{t}(\Omega_{\rm m}, H_0 r_d)]. \tag{21}$$

A useful computational trick. At fixed $\Omega_{\rm m}$, all three BAO observables in Eqs. (18)–(20) scale linearly with $c/(H_0 r_d)$. Therefore one can write

$$\boldsymbol{t}(\Omega_{\rm m}, H_0 r_d) = A\,\boldsymbol{g}(\Omega_{\rm m}), \qquad A \equiv \frac{c}{H_0 r_d}, \tag{22}$$

which implies

$$\chi^2 = A^2\,\boldsymbol{g}^T\mathbf{C}^{-1}\boldsymbol{g} - 2A\,\mathbf{D}^T\mathbf{C}^{-1}\boldsymbol{g} + \mathbf{D}^T\mathbf{C}^{-1}\mathbf{D}. \tag{23}$$

This is pedagogically important. It makes the BAO degeneracy immediately visible and also speeds up contour calculations, because one only needs to evaluate the redshift integrals as functions of $\Omega_{\rm m}$.

Which data are used here? For a fully reproducible classroom example, I use the public DESI DR2 BAO mean vectors and covariance matrices distributed in the Cobaya BAO data repository, specifically the `desi_bao_dr2` files that correspond to the DESI DR2 galaxy/QSO and Lyα BAO measurements [11, 23, 21, 22]. The combined mean vector contains:

- one isotropic BGS point in D_V/r_d at $z = 0.295$;
- anisotropic D_M/r_d and D_H/r_d points for LRG1, LRG2, LRG3+ELG1, ELG2, and QSO;
- a final Lyα block at $z = 2.33$.

Because the covariance matrix is block diagonal between tracers, one can also plot each tracer's contour separately and see the redshift evolution of the degeneracy direction.

Figure 1 is exactly the kind of plot students should learn to interpret. The low-z BGS measurement gives a broad band because it is isotropic and constrains only D_V/r_d. The anisotropic tracers constrain both D_M/r_d and D_H/r_d, so their degeneracy directions rotate. When the tracer blocks are combined, the intersection of these rotated bands produces a tight contour. In my reproduction, the combined posterior is centered near $\Omega_{\rm m} \simeq 0.298$ and $H_0 r_d \simeq 1.015 \times 10^4\,{\rm km\,s^{-1}}$, in good qualitative agreement with the DESI DR2 BAO-only result.

10. A compact Julia example

The point of the Julia example is not to replace Cobaya or CosmoMC. It is to make autodiff and HMC transparent. To keep the HMC dynamics smooth, the example samples an unconstrained variable pair $\boldsymbol{u} = (u_1, u_2) \in \mathbb{R}^2$ and maps it to the physical parameters via logistic transforms,

$$\Omega_{\rm m} = \Omega_m^{\rm min} + (\Omega_m^{\rm max} - \Omega_m^{\rm min})\,\sigma(u_1), \qquad H_0 r_d = X_{\rm min} + (X_{\rm max} - X_{\rm min})\,\sigma(u_2), \tag{24}$$

with $\sigma(u) = 1/(1+e^{-u})$. If one wants a uniform prior in the physical parameters, one adds the Jacobian term of this transformation to the log posterior.

The full companion script is provided separately as `lecture18_desi_bao_hmc_example.jl`. The central logic is short enough to display here.

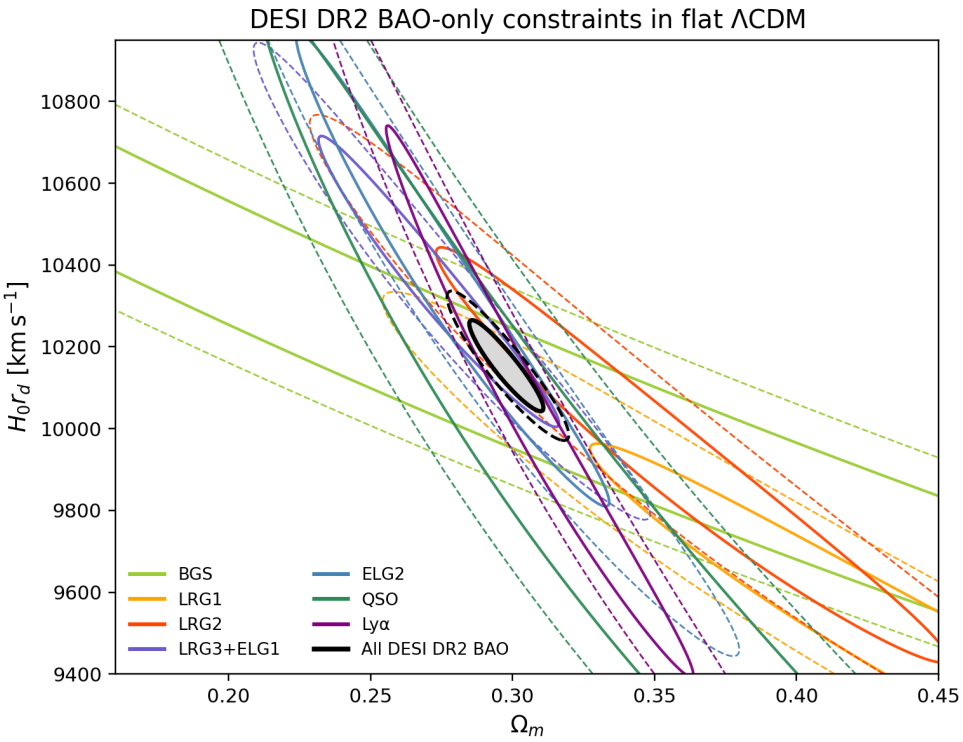

FIGURE 1. A pedagogical reproduction of the DESI DR2 BAO-only contour plot in the $(\Omega_{\rm m}, H_0 r_d)$ plane, using the public DESI DR2 BAO mean vector and covariance files distributed with Cobaya's BAO likelihoods [23, 11]. Solid and dashed lines show approximate 68% and 95% contours for each tracer and for the combined DESI DR2 BAO fit in flat ΛCDM. The counterclockwise rotation of the degeneracy direction with increasing redshift is the same qualitative behavior emphasized by DESI in their figure of individual-tracer consistency [21].

```
using LinearAlgebra, Random, Statistics, ForwardDiff

const c_light = 299792.458

sigmoid(x) = inv(one(x) + exp(-x))

function unpack(u)
    s1, s2 = sigmoid(u[1]), sigmoid(u[2])
    Omega_m = 0.05 + 0.55*s1
    H0rd = 9000.0 + 2500.0*s2
    logJ = log(0.55*s1*(1-s1)) + log(2500.0*s2*(1-s2))
```

```
    return Omega_m, H0rd, logJ
end

E(z, Omega_m) = sqrt(Omega_m*(1 + z)^3 + 1 - Omega_m)

function chiint(z, Omega_m; n = 800)
    h = z / n
    s = inv(E(0.0, Omega_m)) + inv(E(z, Omega_m))
    for i in 1:(n - 1)
        zi = i*h
        s += (isodd(i) ? 4.0 : 2.0) / E(zi, Omega_m)
    end
    return h*s/3.0
end

function logposterior(u)
    Omega_m, H0rd, logJ = unpack(u)
    model = bao_model_vector(Omega_m, H0rd)
    Delta = data_vector .- model
    return -0.5 * dot(Delta, Cinv * Delta) + logJ
end

U(u) = -logposterior(u)
gradU(u) = ForwardDiff.gradient(U, u)

function leapfrog(q, p, eps, L)
    p .-= 0.5 * eps * gradU(q)
    for i in 1:L
        q .+= eps * p
        if i != L
            p .-= eps * gradU(q)
        end
    end
    p .-= 0.5 * eps * gradU(q)
    return q, -p
end
```

Students should pause at three places.

First, `ForwardDiff.gradient(U, u)` is the autodiff step. No analytic derivative of the BAO integrals is written down by hand.

Second, the potential $U = -\log p$ is exactly the quantity required by HMC. Once U and ∇U are available, the leapfrog integrator follows automatically.

Third, the transformation from unconstrained u_i to physical parameters is not a technical nuisance. It is one of the most important practical ideas in real sampling: a good parameterization often matters as much as the choice of sampler.

11. How this classroom example connects back to CosmoMC and Cobaya

At this point students often ask a sensible question: if HMC is so elegant, why do cosmology papers still use CosmoMC-style Metropolis chains so often? The answer is practical.

CosmoMC and Cobaya are complete inference ecosystems. They manage complicated likelihood combinations, external theory codes, nuisance hierarchies, chain restarts, MPI, post-processing, and standardized data products. Their default MCMC engines are robust and well-tested on exactly the kinds of likelihoods cosmologists use every day [5, 6, 9].

The Julia example in this lecture serves a different purpose. It teaches the geometry of sampling and the role of autodiff on a problem where students can understand every moving part. Once students understand that, it becomes much easier to appreciate what CosmoMC and Cobaya are doing for them in full analyses.

12. Summary

The central lessons of this lecture are straightforward.

Bayesian inference in cosmology is a forward-model problem, and the posterior is rarely accessible on a brute-force grid. Metropolis-Hastings MCMC solves the normalization problem and remains the basic conceptual foundation of many cosmological pipelines. CosmoMC turned this logic into a highly optimized cosmology sampler; Cobaya generalized it into a modular inference framework with modern interfaces and current DESI likelihood support.

Automatic differentiation gives us gradients of the log posterior without hand differentiation. HMC then uses those gradients to build long-distance, high-acceptance proposals through approximate Hamiltonian dynamics. This is mathematically elegant and often much more efficient than a random walk.

Finally, the DESI BAO-only $(\Omega_{\rm m}, H_0 r_d)$ example is valuable because it ties the statistics back to cosmology. The equations are simple, the degeneracy is physically interpretable, and the contour rotation with redshift shows directly how complementary tracers combine into a tight cosmological measurement.

Suggested reading for students

For the cosmology-sampler viewpoint, start with Lewis and Bridle [4] and then Lewis' fast-slow sampler paper [5]. For Cobaya, read Torrado and Lewis [6] together with the current documentation [7, 11]. For HMC, Neal [18] and Betancourt [19] are the most useful references for building intuition. For automatic differentiation, Baydin *et al.* and the ForwardDiff references are the best compact introductions [14, 15, 16]. For the DESI example itself, read the DESI DR2 BAO papers and inspect the public BAO data files distributed with Cobaya [21, 22, 23].

References

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.* **21**, 1087-1092 (1953).

[2] W. K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika* **57**, 97-109 (1970).

[3] A. Gelman and D. B. Rubin, "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science* **7**, 457-472 (1992).

[4] A. Lewis and S. Bridle, "Cosmological Parameters from CMB and Other Data: a Monte Carlo Approach," *Phys. Rev. D* **66**, 103511 (2002), arXiv:astro-ph/0205436.

[5] A. Lewis, "Efficient Sampling of Fast and Slow Cosmological Parameters," *Phys. Rev. D* **87**, 103529 (2013), arXiv:1304.4473 [astro-ph.CO].

[6] J. Torrado and A. Lewis, "Cobaya: Code for Bayesian Analysis of Hierarchical Physical Models," *JCAP* **05**, 057 (2021), arXiv:2005.05290 [astro-ph.IM].

[7] J. Torrado and A. Lewis, "Cobaya documentation," `https://cobaya.readthedocs.io/`, accessed June 2026.

[8] J. Torrado and A. Lewis, "Cobaya input and invocation documentation," `https://cobaya.readthedocs.io/en/latest/input.html`, accessed June 2026.

[9] J. Torrado and A. Lewis, "Cobaya MCMC sampler documentation," `https://cobaya.readthedocs.io/en/latest/sampler_mcmc.html`, accessed June 2026.

[10] J. Torrado and A. Lewis, "Cobaya cosmology installation documentation," `https://cobaya.readthedocs.io/en/latest/installation_cosmo.html`, accessed June 2026.

[11] A. Lewis, P. Lemos, and J. Torrado, "Cobaya BAO likelihood documentation," `https://cobaya.readthedocs.io/en/latest/likelihood_bao.html`, accessed June 2026.

[12] A. Lewis, "CosmoMC GitHub repository README," `https://github.com/cmbant/CosmoMC`, accessed June 2026.

[13] A. Lewis, "CosmoMC website," `https://cosmologist.info/cosmomc/`, accessed June 2026.

[14] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic Differentiation in Machine Learning: a Survey," *J. Mach. Learn. Res.* **18**, 1-43 (2018), arXiv:1502.05767.

[15] J. Revels, M. Lubin, and T. Papamarkou, "Forward-Mode Automatic Differentiation in Julia," arXiv:1607.07892 [cs.MS].

[16] J. Revels, M. Lubin, and T. Papamarkou, "ForwardDiff.jl documentation," `https://juliadiff.org/ForwardDiff.jl/`, accessed June 2026.

[17] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Phys. Lett. B* **195**, 216-222 (1987).

[18] R. M. Neal, "MCMC Using Hamiltonian Dynamics," in *Handbook of Markov Chain Monte Carlo*, edited by S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, Chapman & Hall/CRC (2011), arXiv:1206.1901.

[19] M. Betancourt, "A Conceptual Introduction to Hamiltonian Monte Carlo," arXiv:1701.02434 [stat.ME].

[20] M. D. Hoffman and A. Gelman, "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.* **15**, 1593-1623 (2014), arXiv:1111.4246.

[21] M. Abdul Karim *et al.* [DESI Collaboration], "DESI DR2 Results II: Measurements of Baryon Acoustic Oscillations and Cosmological Constraints," arXiv:2503.14738 [astro-ph.CO].

[22] M. Abdul Karim *et al.* [DESI Collaboration], "DESI DR2 Results I: Baryon Acoustic Oscillations from the Lyman Alpha Forest," arXiv:2503.14739 [astro-ph.CO].

[23] A. Lewis and collaborators, "Cobaya BAO data repository," `https://github.com/CobayaSampler/bao_data`, accessed June 2026.