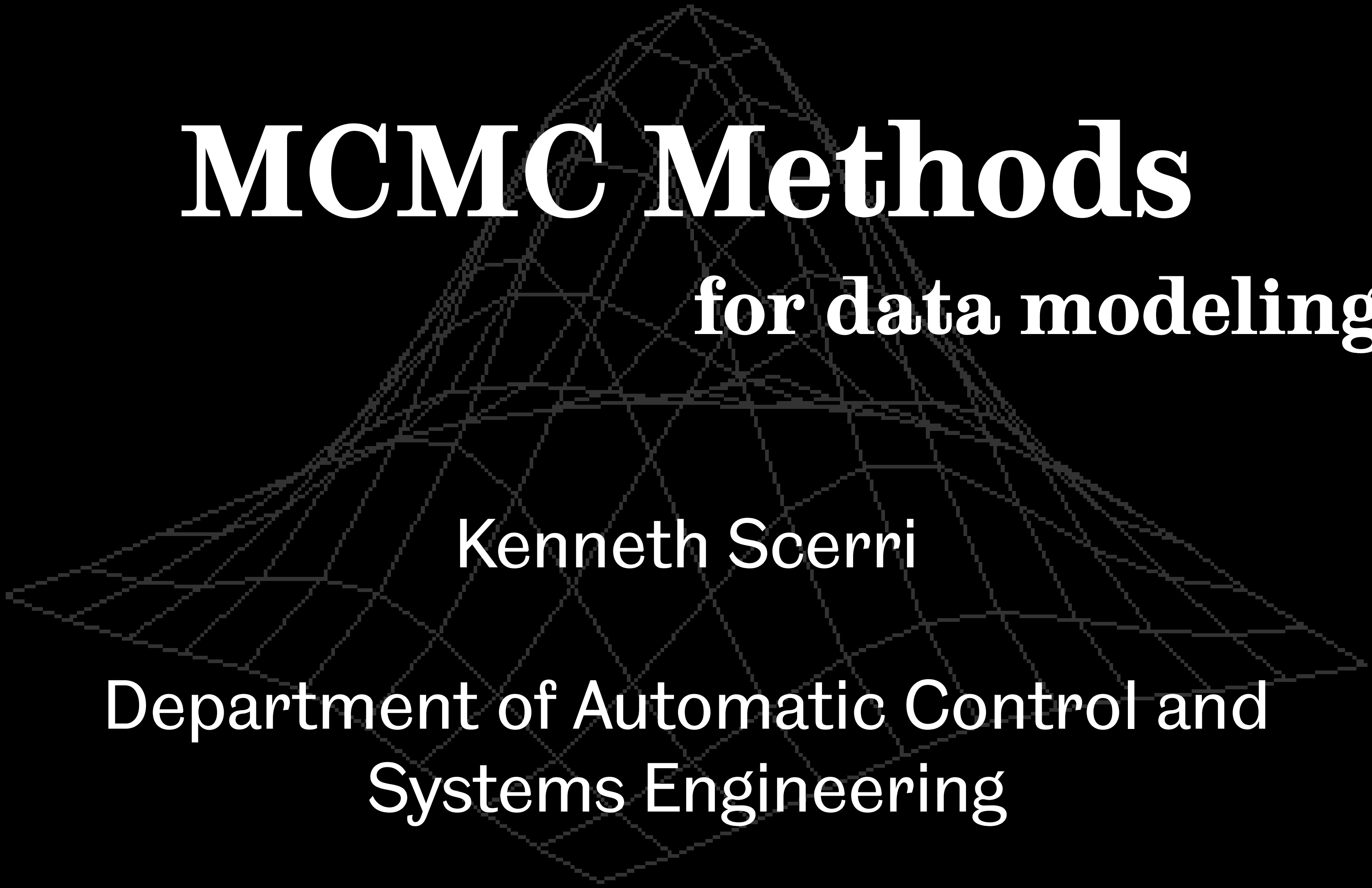
MCMC Methods

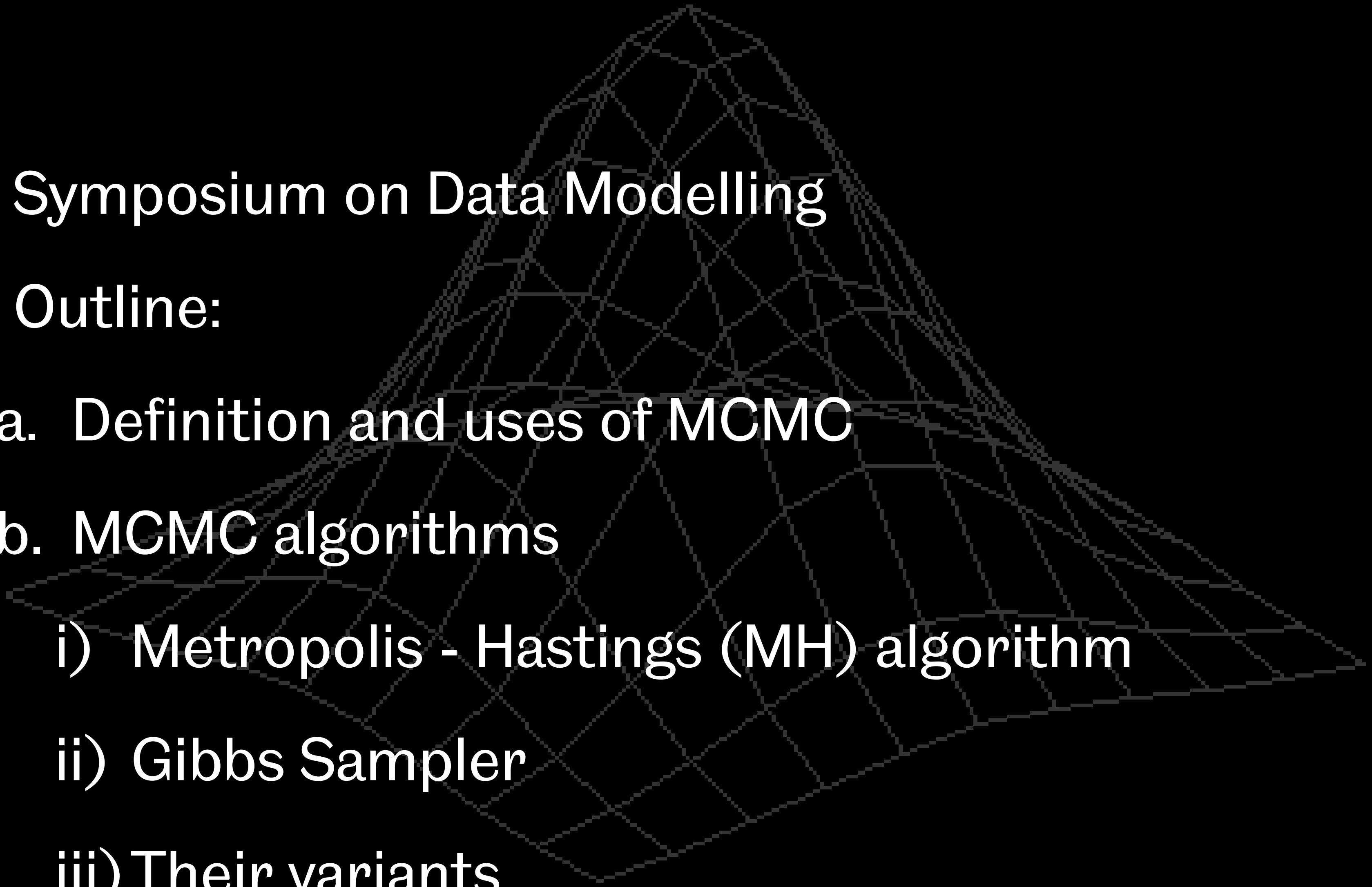
for data modeling



Kenneth Scerri

Department of Automatic Control and
Systems Engineering

Introduction



1. Symposium on Data Modelling
2. Outline:
    a. Definition and uses of MCMC
    b. MCMC algorithms
        i) Metropolis - Hastings (MH) algorithm
        ii) Gibbs Sampler
        iii) Their variants

How does this fit?

Three problems ‘solved’ by data modeling

1. Regression
2. Classification
3. Density Estimation

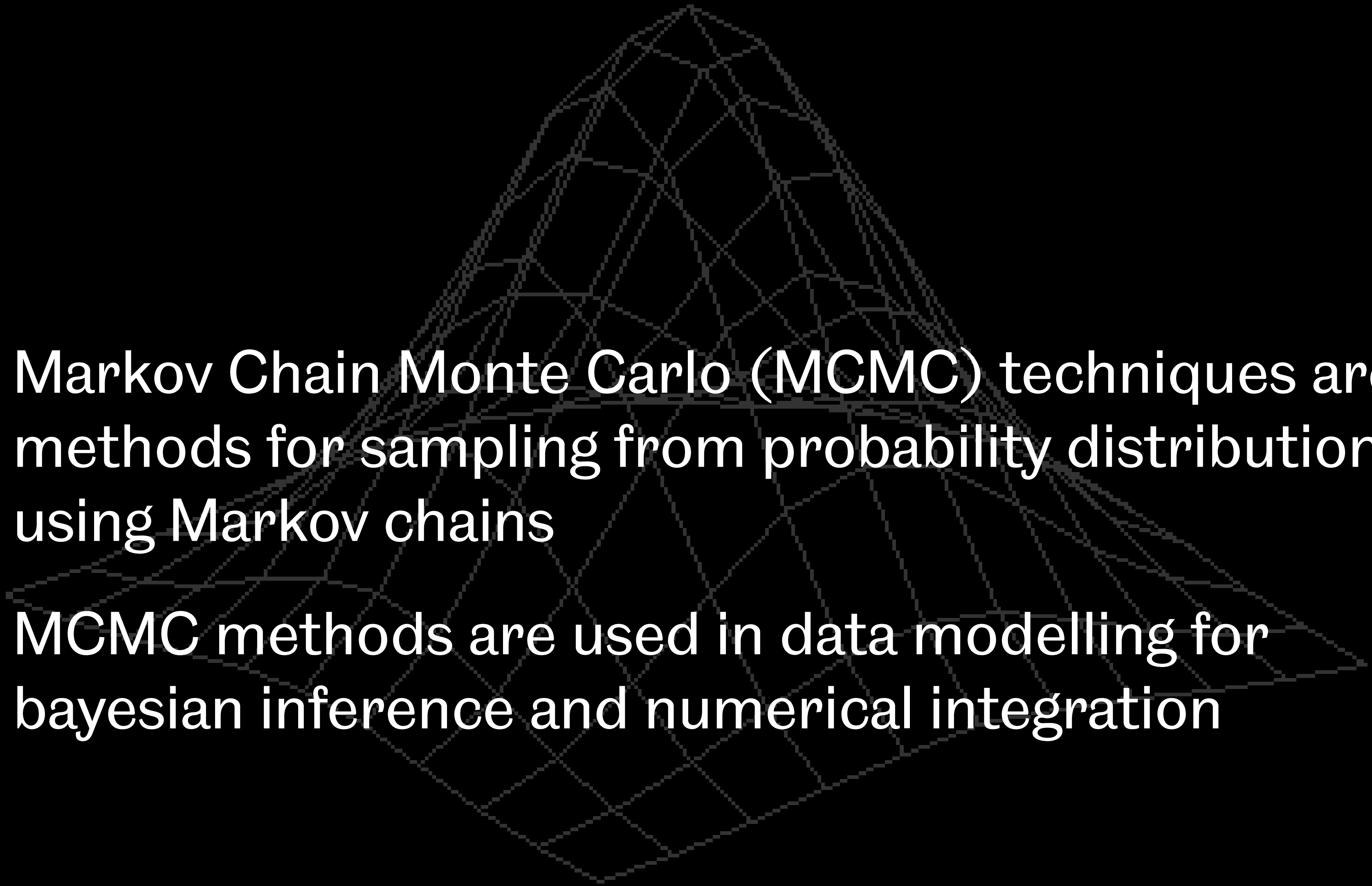
These are methods that estimate densities. These densities can then be utilized for both regression and classification

$$y = \sum_{i=1}^{N} w_i x_i + e$$

We will be using

$$y = \sum_{i=1}^{N} \theta_i x_i + e$$

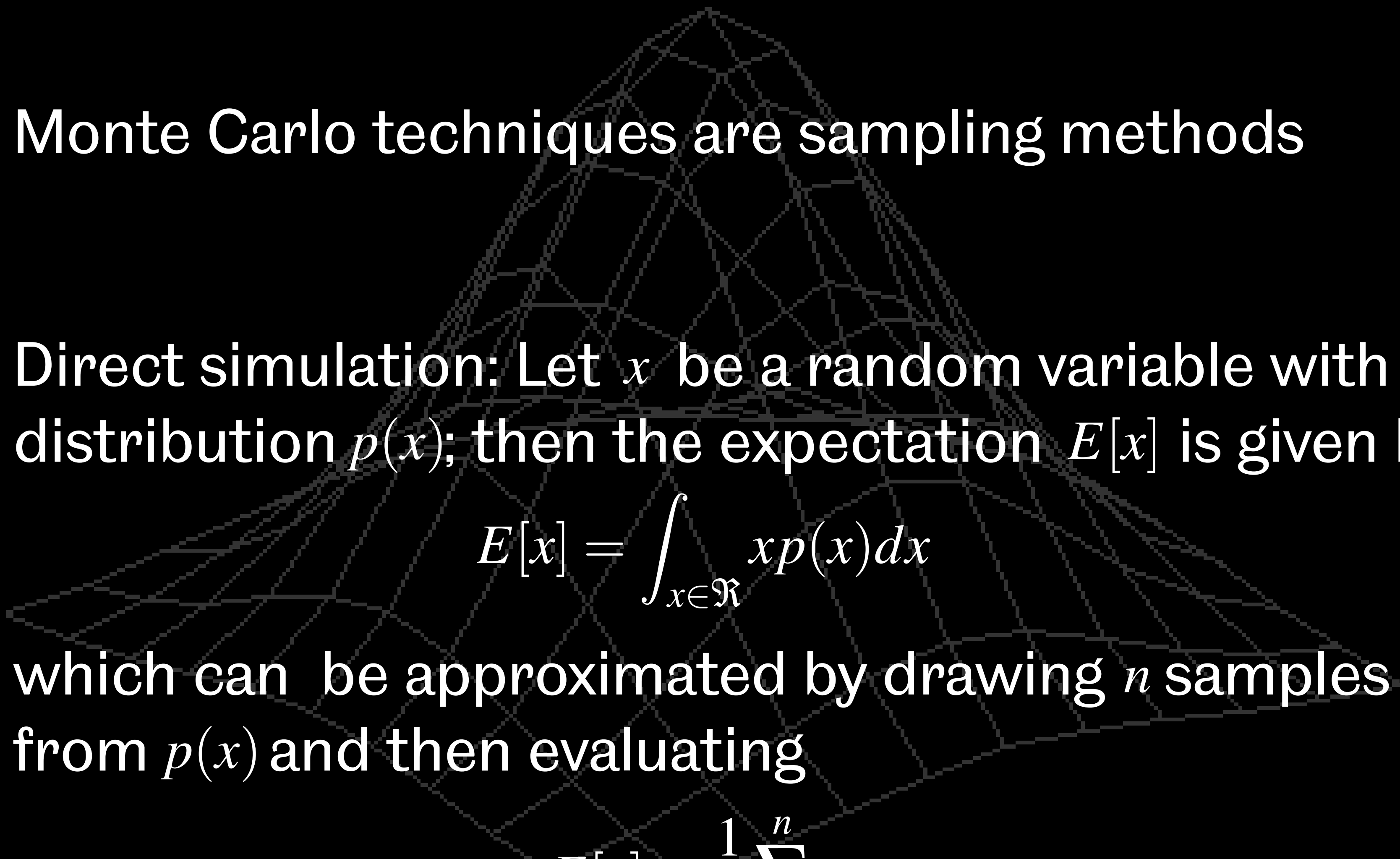
Definition



Markov Chain Monte Carlo (MCMC) techniques are methods for sampling from probability distributions using Markov chains

MCMC methods are used in data modelling for bayesian inference and numerical integration

Monte Carlo Methods



Monte Carlo techniques are sampling methods

Direct simulation: Let x be a random variable with distribution $p(x)$; then the expectation $E[x]$ is given by:

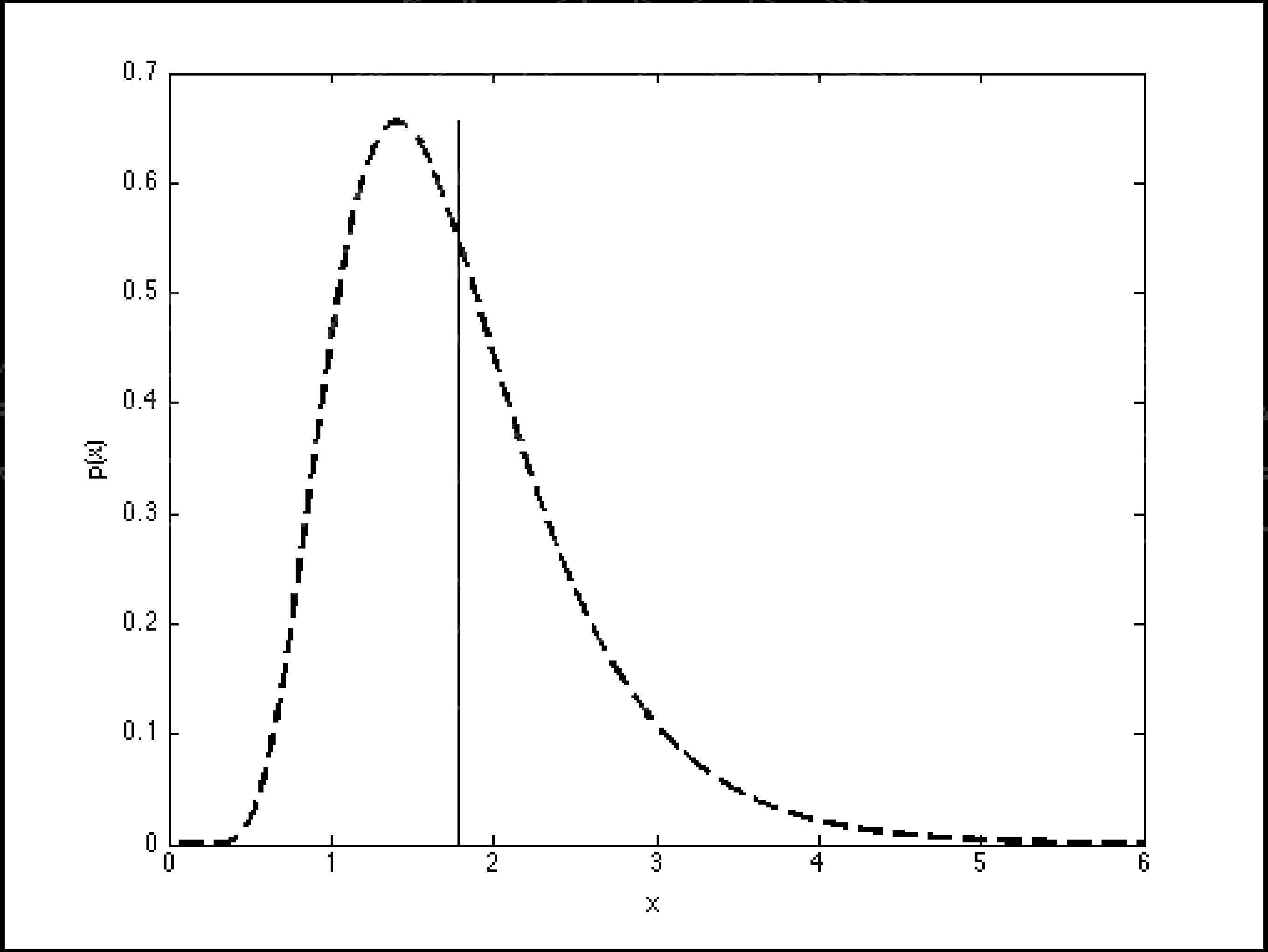
$$E[x] = \int_{x\in\Re} x p(x) dx$$

which can be approximated by drawing n samples from $p(x)$ and then evaluating

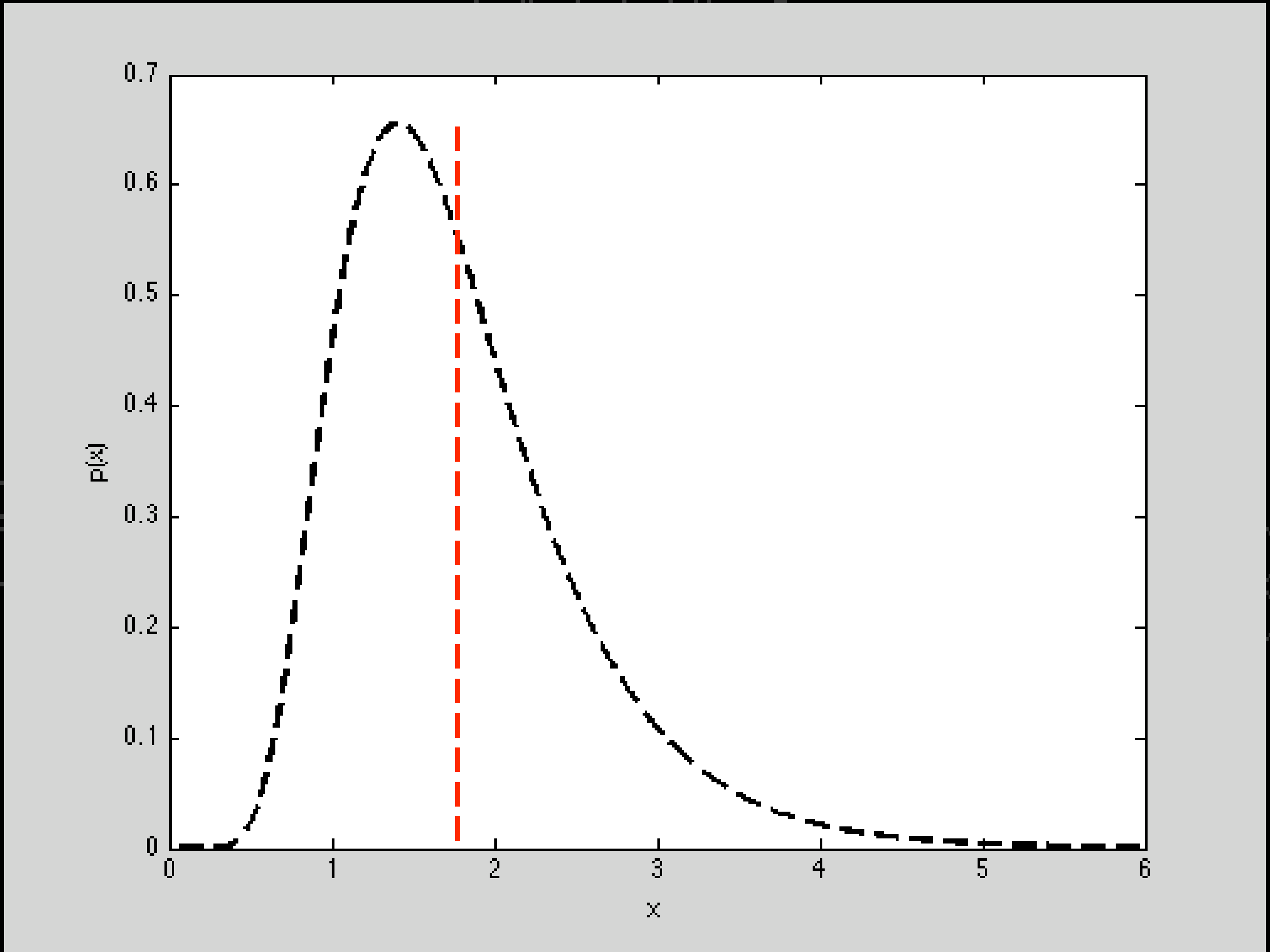
$$E[x] \approx \frac{1}{n}\sum_{i=1}^{n} x_i$$

Monte Carlo Integration

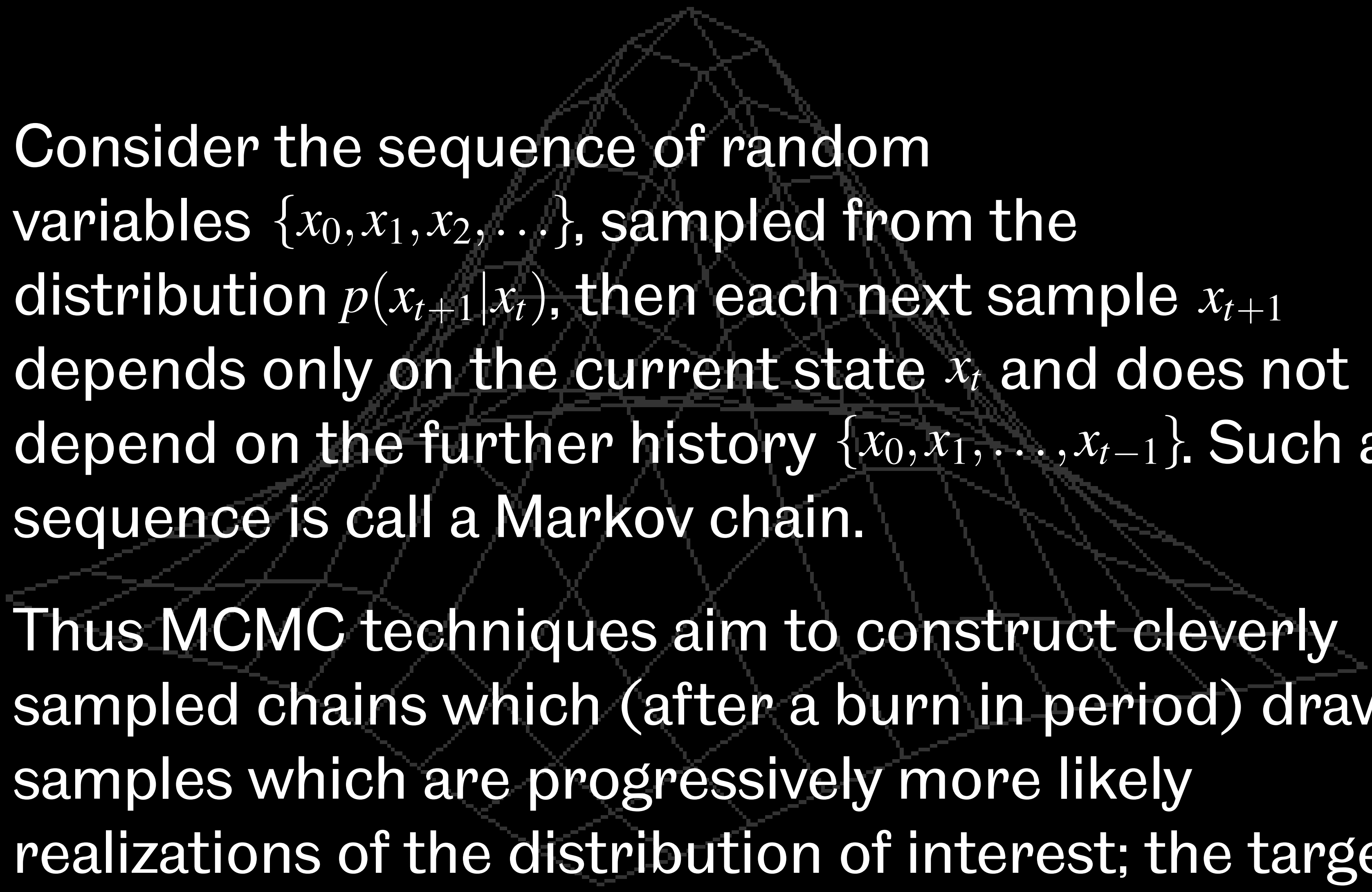
Example 1. Mean of a lognormal distribution by direct simulation



Monte Carlo Integration



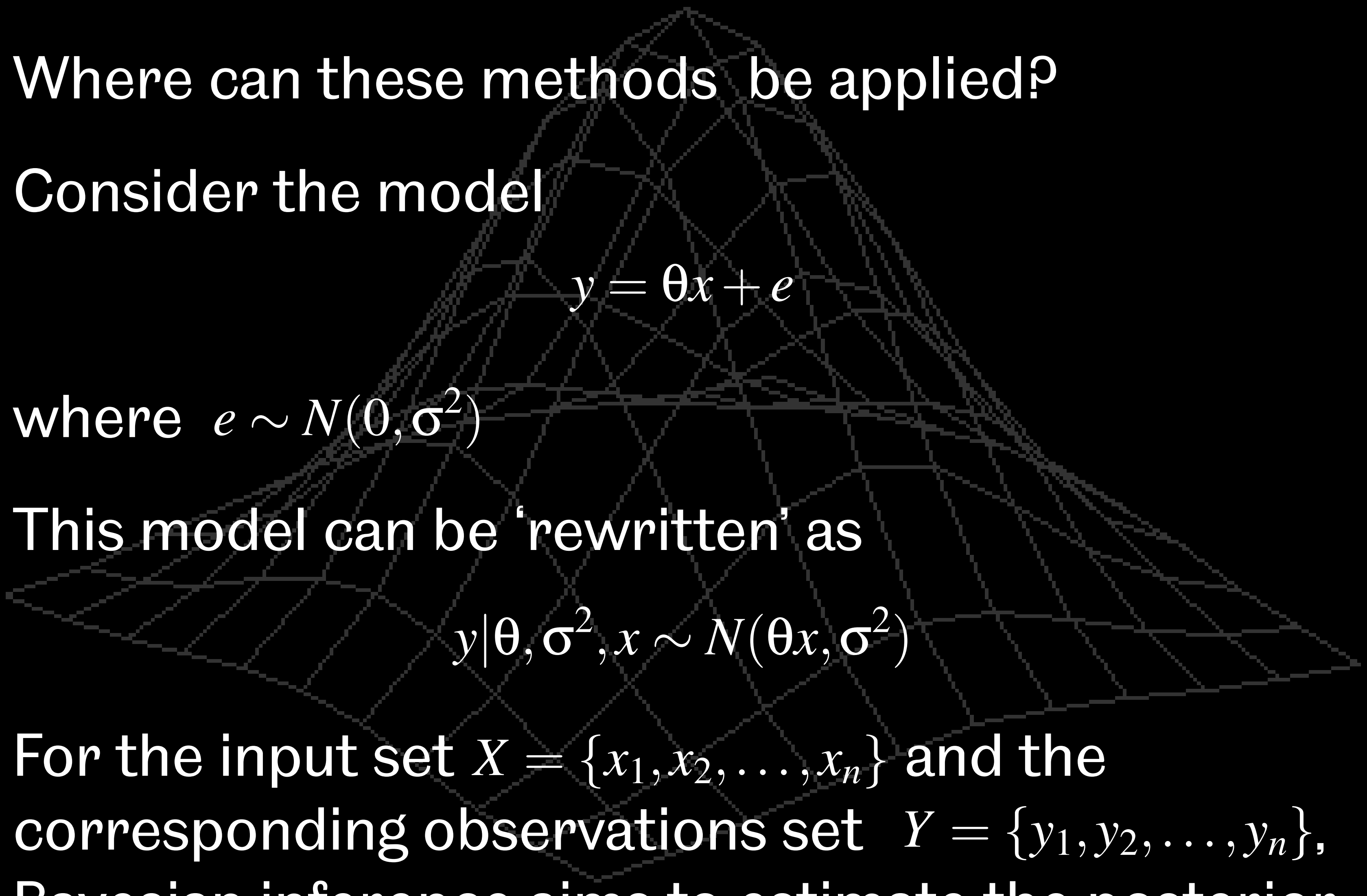
Markov Chains



Consider the sequence of random variables $\{x_0, x_1, x_2, \ldots\}$, sampled from the distribution $p(x_{t+1}|x_t)$, then each next sample x_{t+1} depends only on the current state x_t and does not depend on the further history $\{x_0, x_1, \ldots, x_{t-1}\}$. Such a sequence is call a Markov chain.

Thus MCMC techniques aim to construct cleverly sampled chains which (after a burn in period) draw samples which are progressively more likely realizations of the distribution of interest; the target distribution.

Bayesian Inference



Where can these methods be applied?

Consider the model

$$y = \theta x + e$$

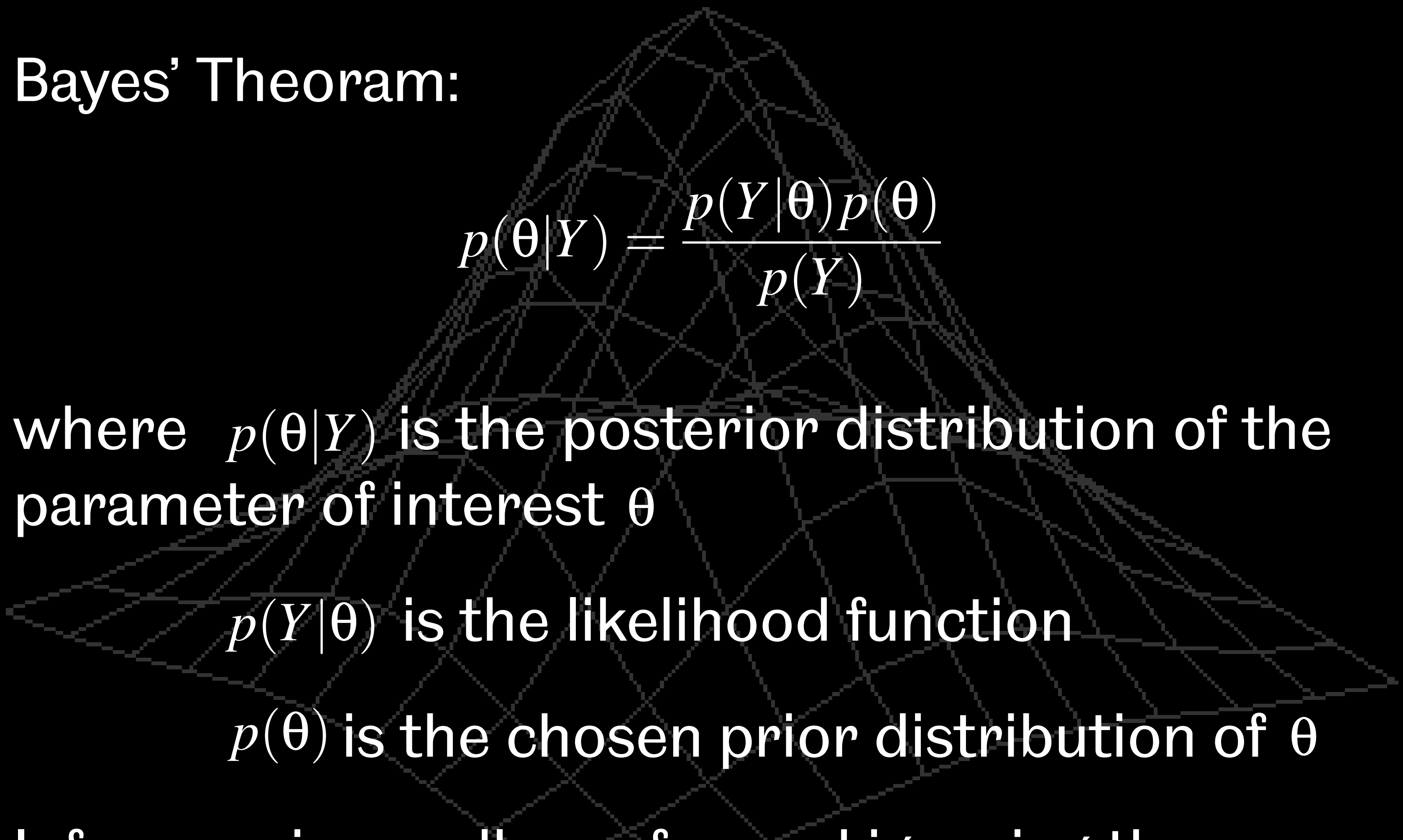
where $e \sim N(0, \sigma^2)$

This model can be ‘rewritten’ as

$$y|\theta, \sigma^2, x \sim N(\theta x, \sigma^2)$$

For the input set $X = \{x_1, x_2, \ldots, x_n\}$ and the corresponding observations set $Y = \{y_1, y_2, \ldots, y_n\}$, Bayesian inference aims to estimate the posterior distribution of the parameter θ via Bayes Theorem.

Bayesian Inference



Bayes' Theoram:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}$$

where $p(\theta|Y)$ is the posterior distribution of the parameter of interest θ

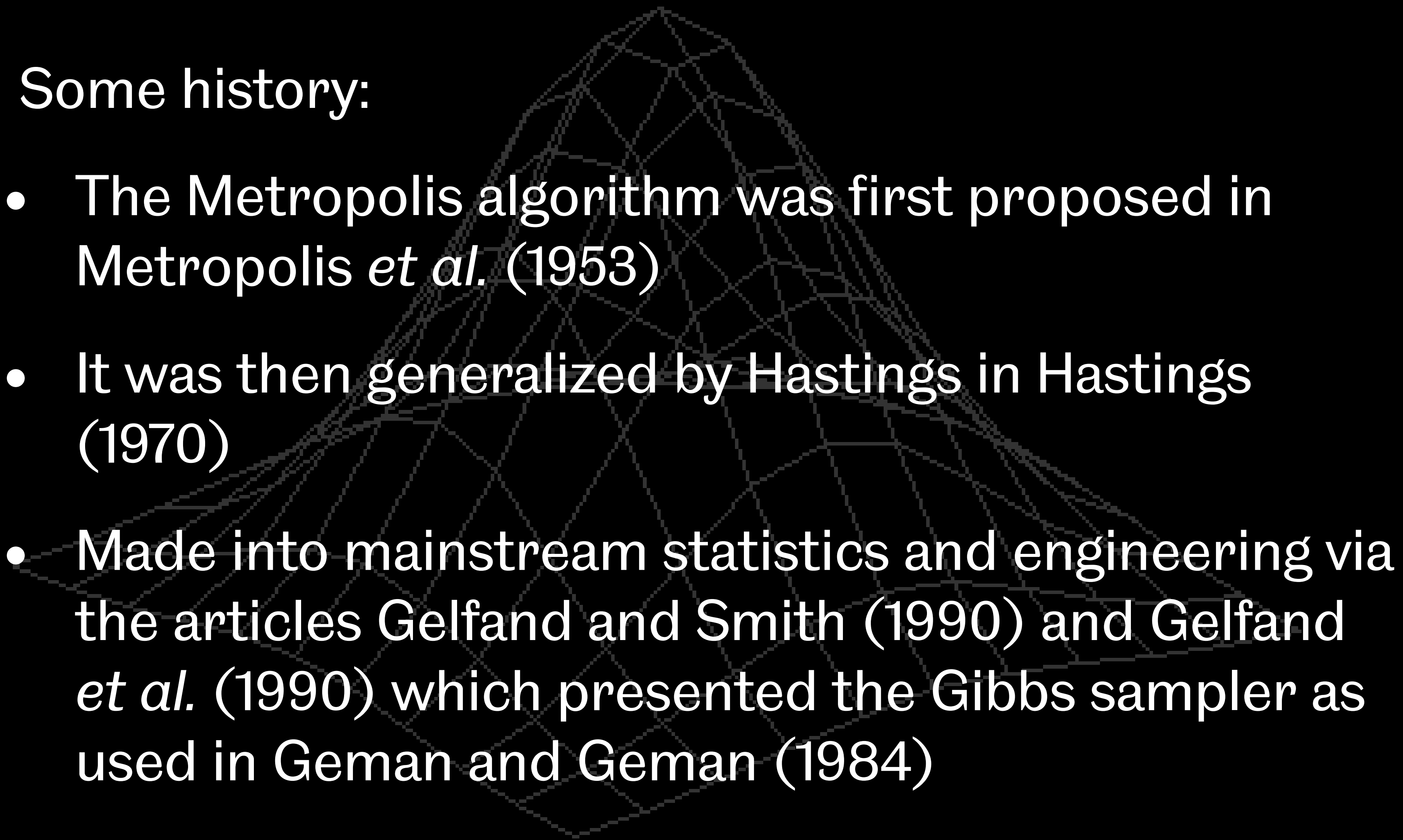
$p(Y|\theta)$ is the likelihood function

$p(\theta)$ is the chosen prior distribution of θ

Inference is usually performed ignoring the normalizing constant $p(Y)$, thus utilizing

$$p(\theta|Y) \propto p(Y|\theta)p(\theta)$$

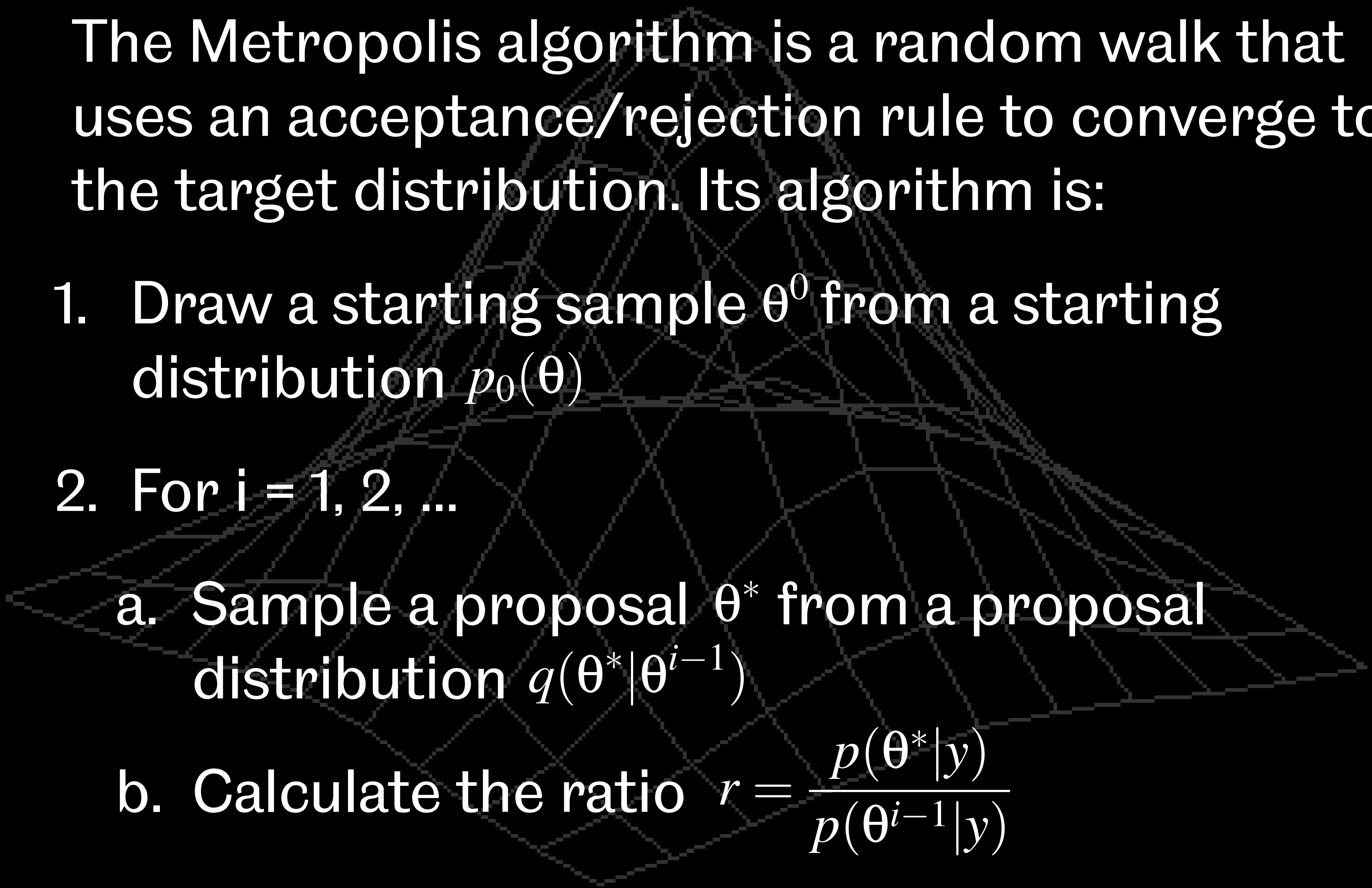
MH Algorithm



- Some history:
  - The Metropolis algorithm was first proposed in Metropolis *et al.* (1953)
  - It was then generalized by Hastings in Hastings (1970)
  - Made into mainstream statistics and engineering via the articles Gelfand and Smith (1990) and Gelfand *et al.* (1990) which presented the Gibbs sampler as used in Geman and Geman (1984)
- All other MCMC methods can be considered as special cases of the MH algorithm

Metropolis Algorithm

The Metropolis algorithm is a random walk that uses an acceptance/rejection rule to converge to the target distribution. Its algorithm is:



1. Draw a starting sample θ^0 from a starting distribution $p_0(\theta)$

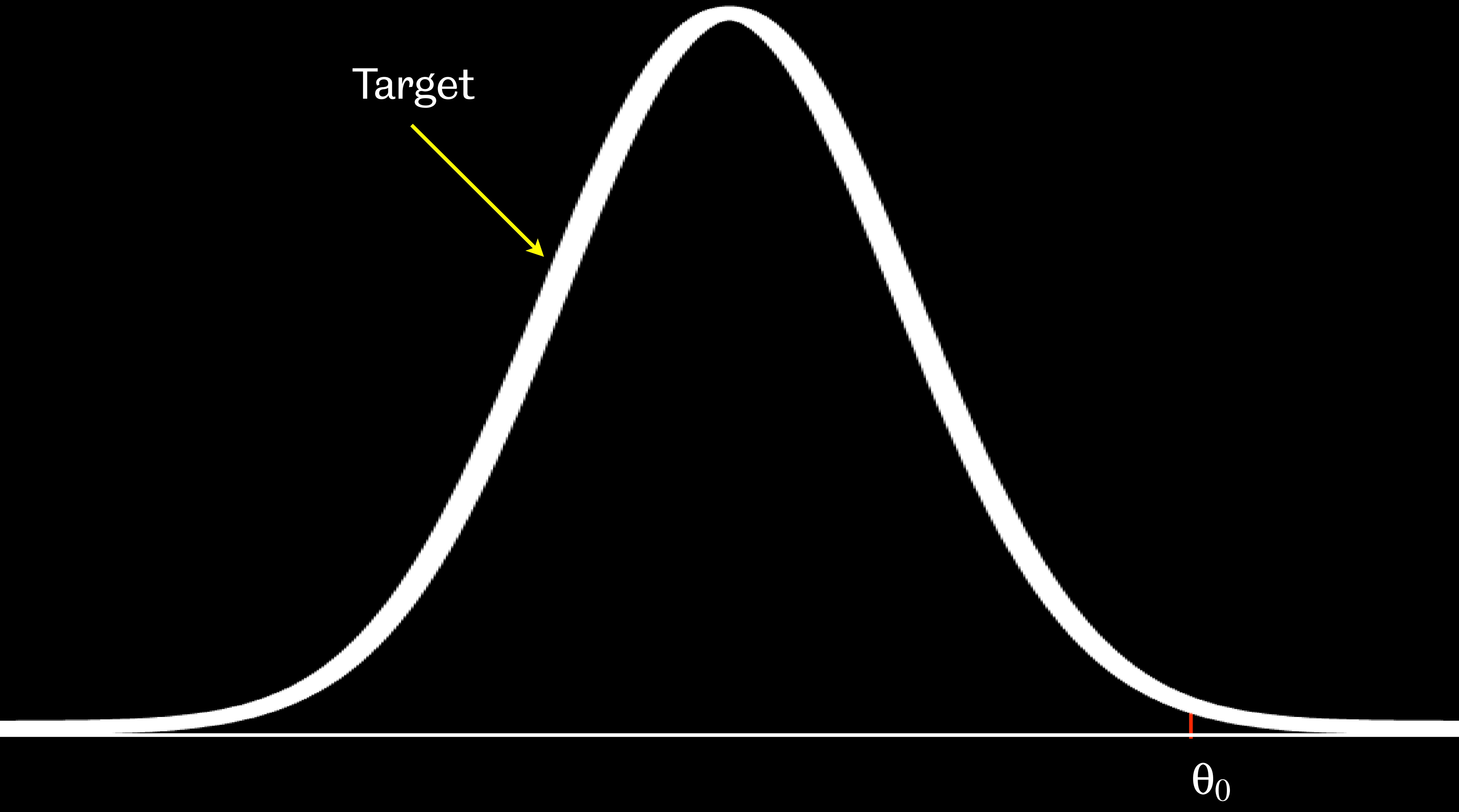
2. For i = 1, 2, ...

   a. Sample a proposal θ^* from a proposal distribution $q(\theta^*|\theta^{i-1})$

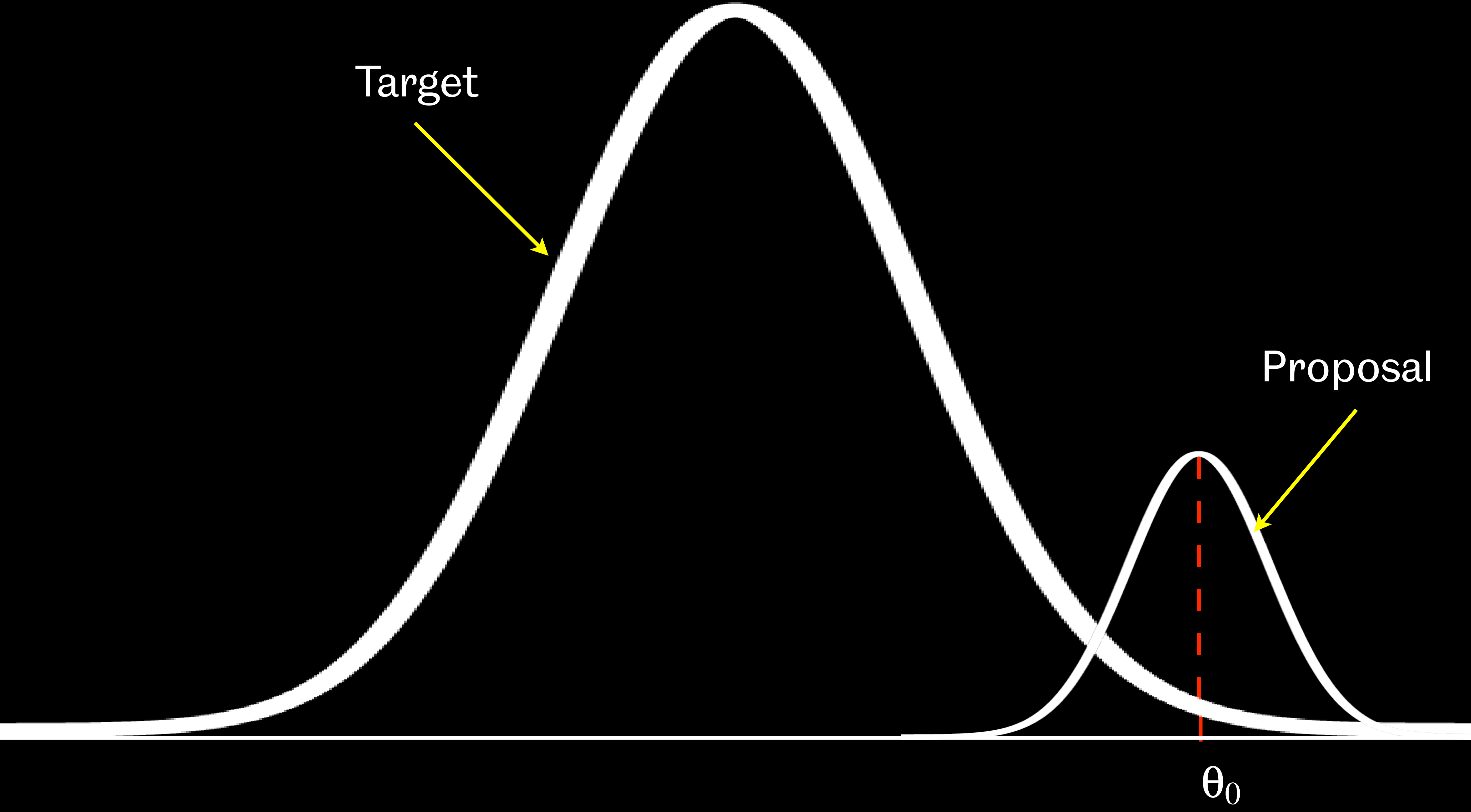
   b. Calculate the ratio $r = \frac{p(\theta^*|y)}{p(\theta^{i-1}|y)}$

   c. Set $\theta^i = \begin{cases} \theta^* & \text{with probability } min(r, 1) \\ \theta^{i-1} & \text{otherwise} \end{cases}$

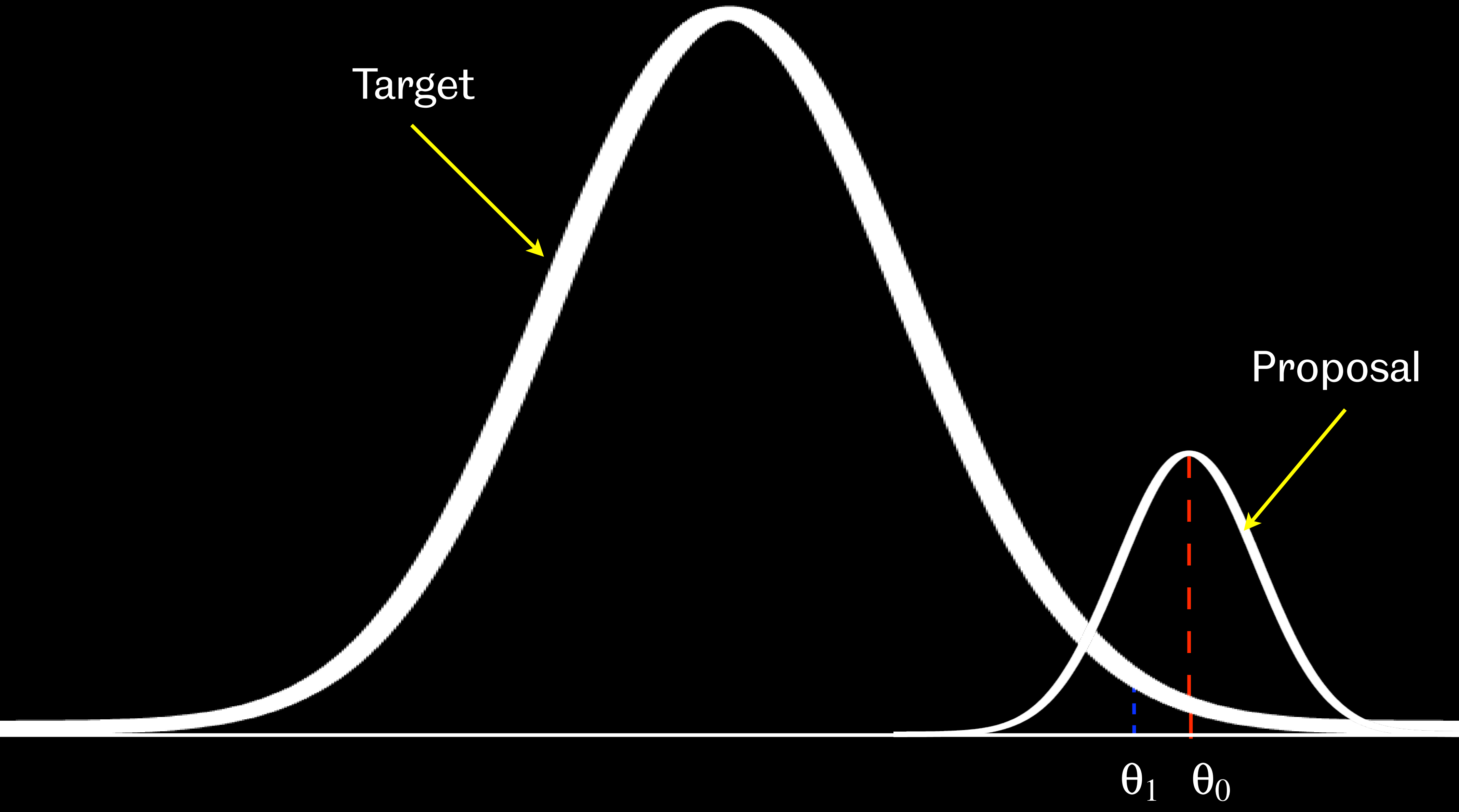
Metropolis Algorithm



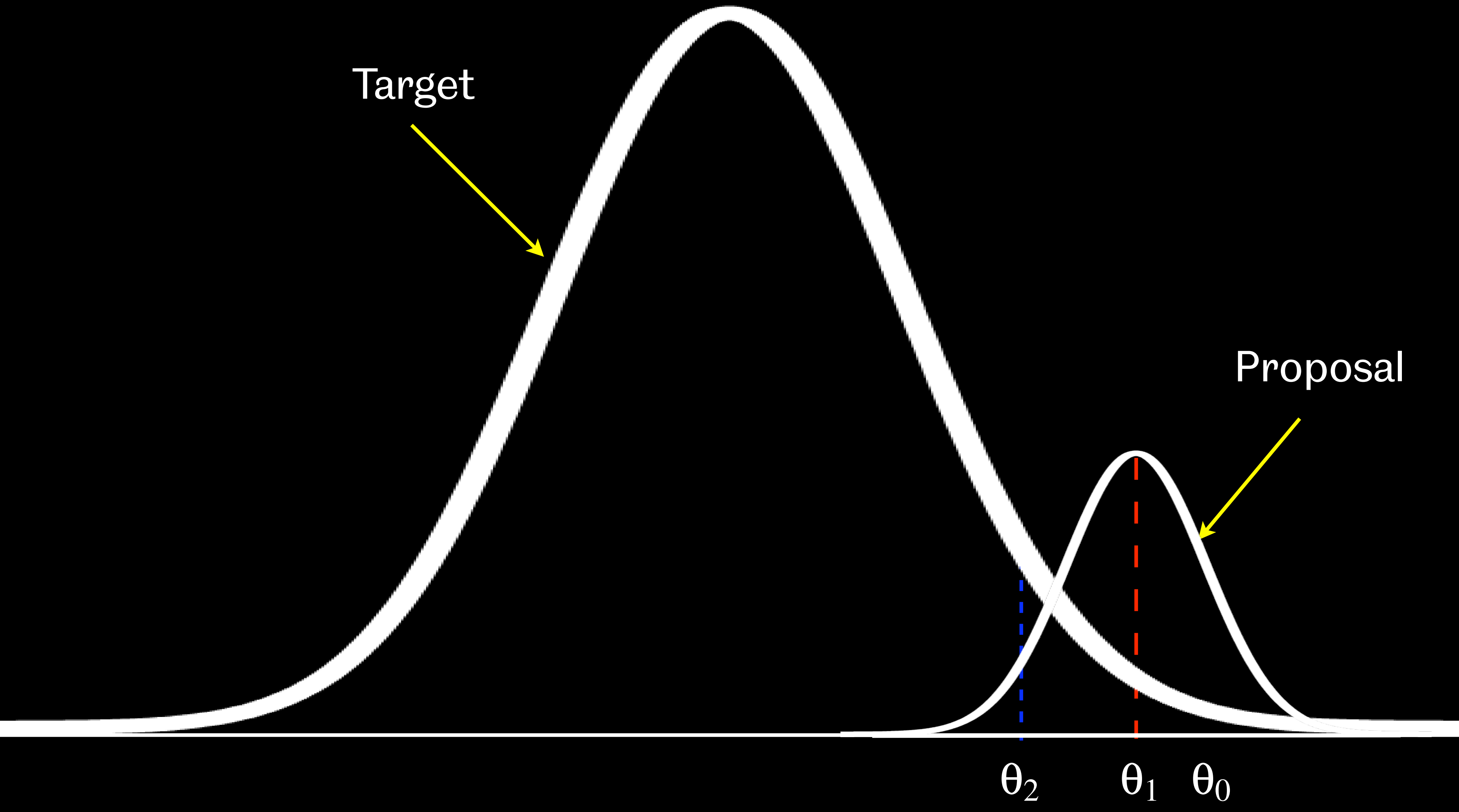
Metropolis Algorithm



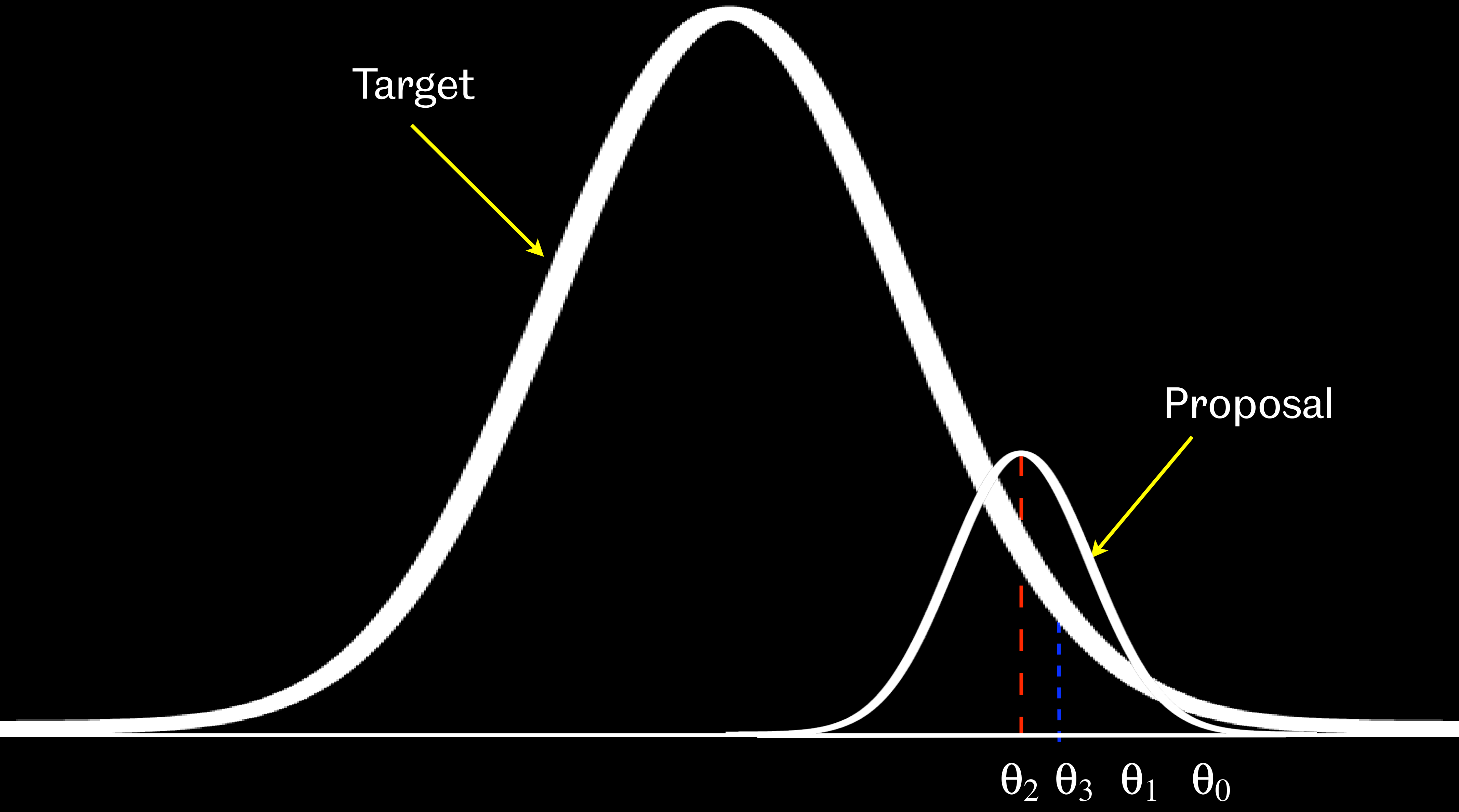
Metropolis Algorithm



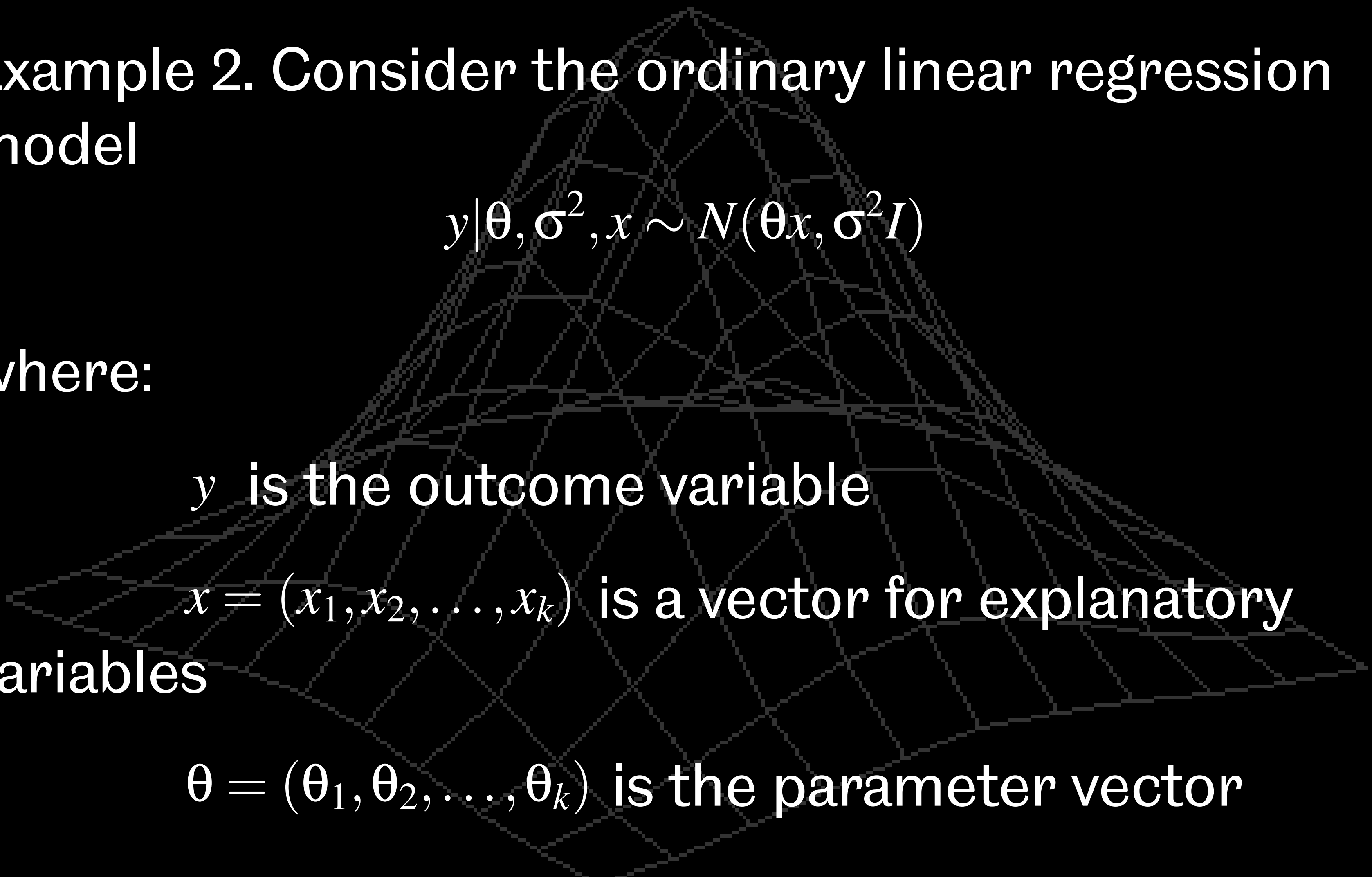
Metropolis Algorithm



Metropolis Algorithm



Metropolis Algorithm



Example 2. Consider the ordinary linear regression model

$$y|\theta,\sigma^2,x \sim N(\theta x,\sigma^2 I)$$

where:

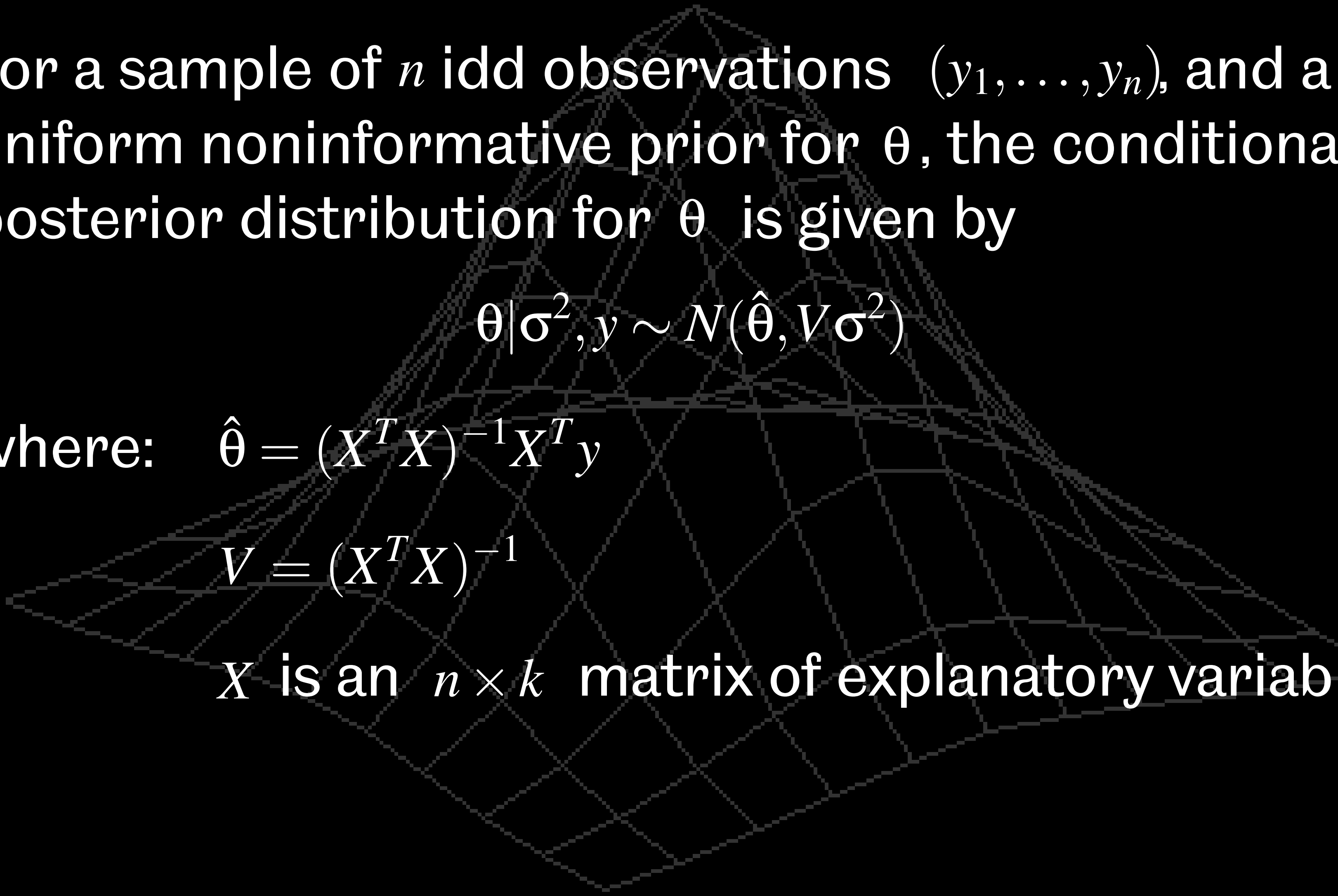
y is the outcome variable

$x = (x_1, x_2, \ldots, x_k)$ is a vector for explanatory variables

$\theta = (\theta_1, \theta_2, \ldots, \theta_k)$ is the parameter vector

σ is the independent observation error standard deviation

Metropolis Algorithm



For a sample of n idd observations $(y_1, \ldots, y_n)$, and a uniform noninformative prior for θ, the conditional posterior distribution for θ is given by

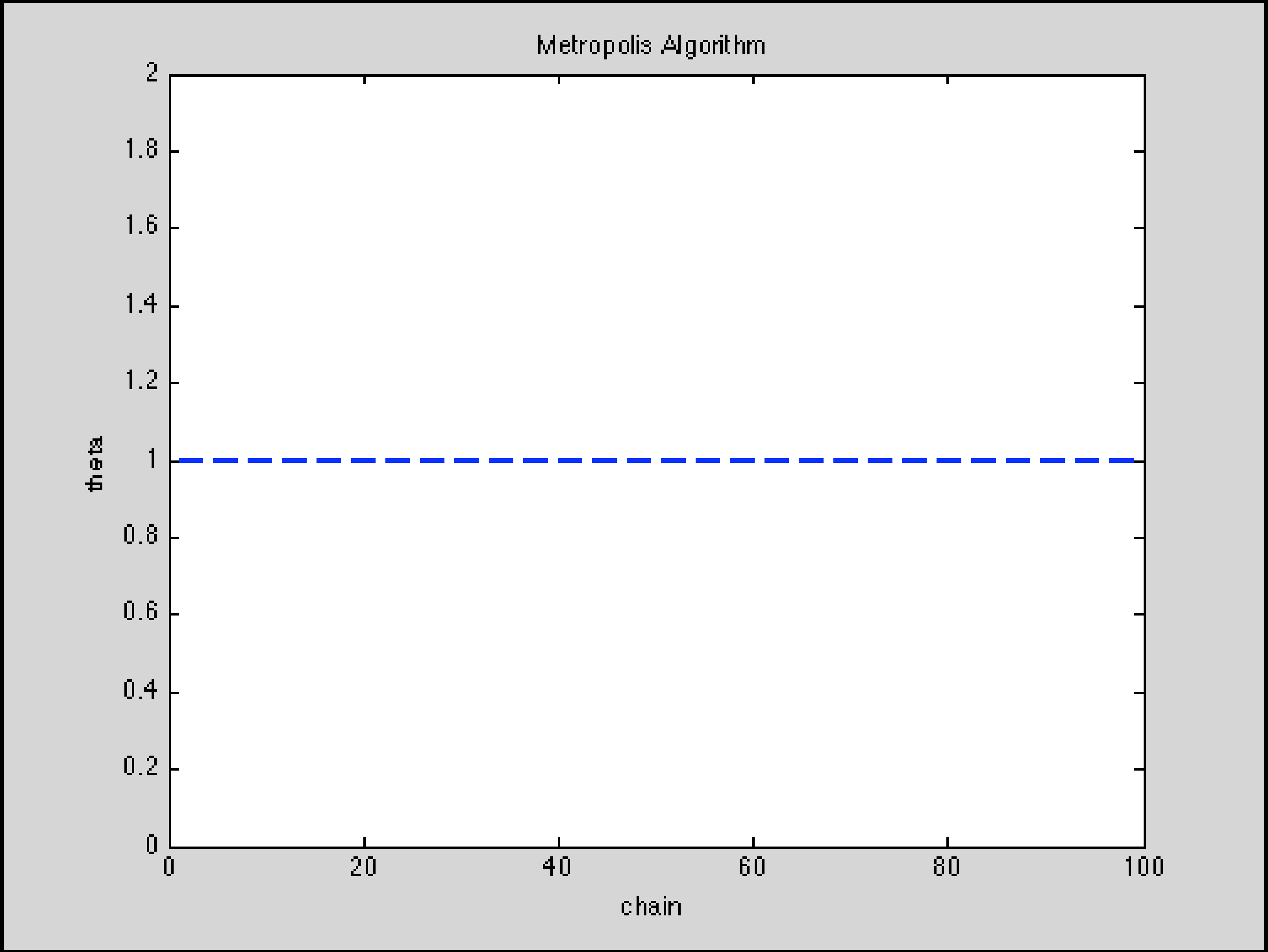
$$\theta | \sigma^2, y \sim N(\hat{\theta}, V\sigma^2)$$

where: $\hat{\theta} = (X^T X)^{-1} X^T y$

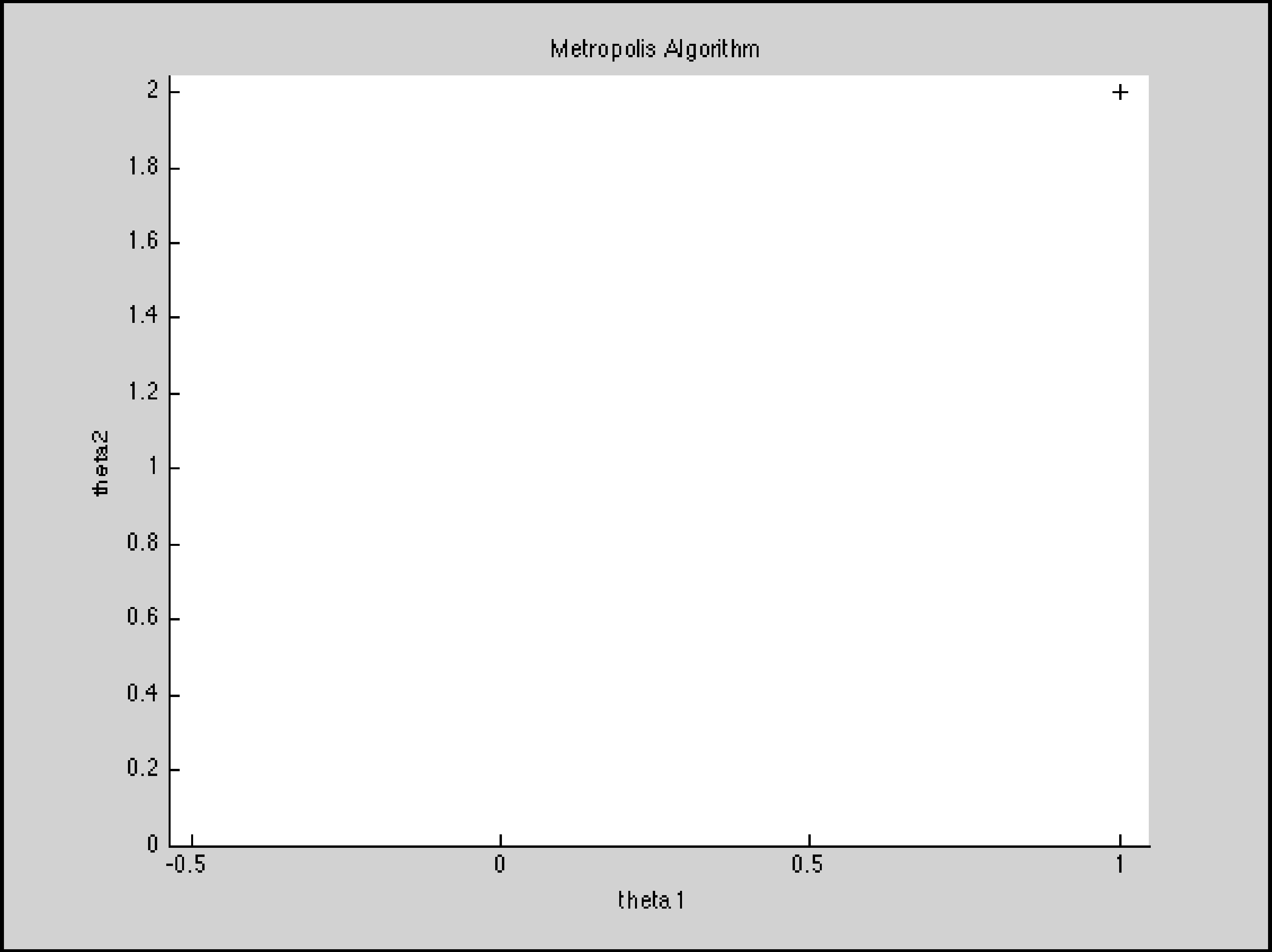
$V = (X^T X)^{-1}$

X is an $n \times k$ matrix of explanatory variables

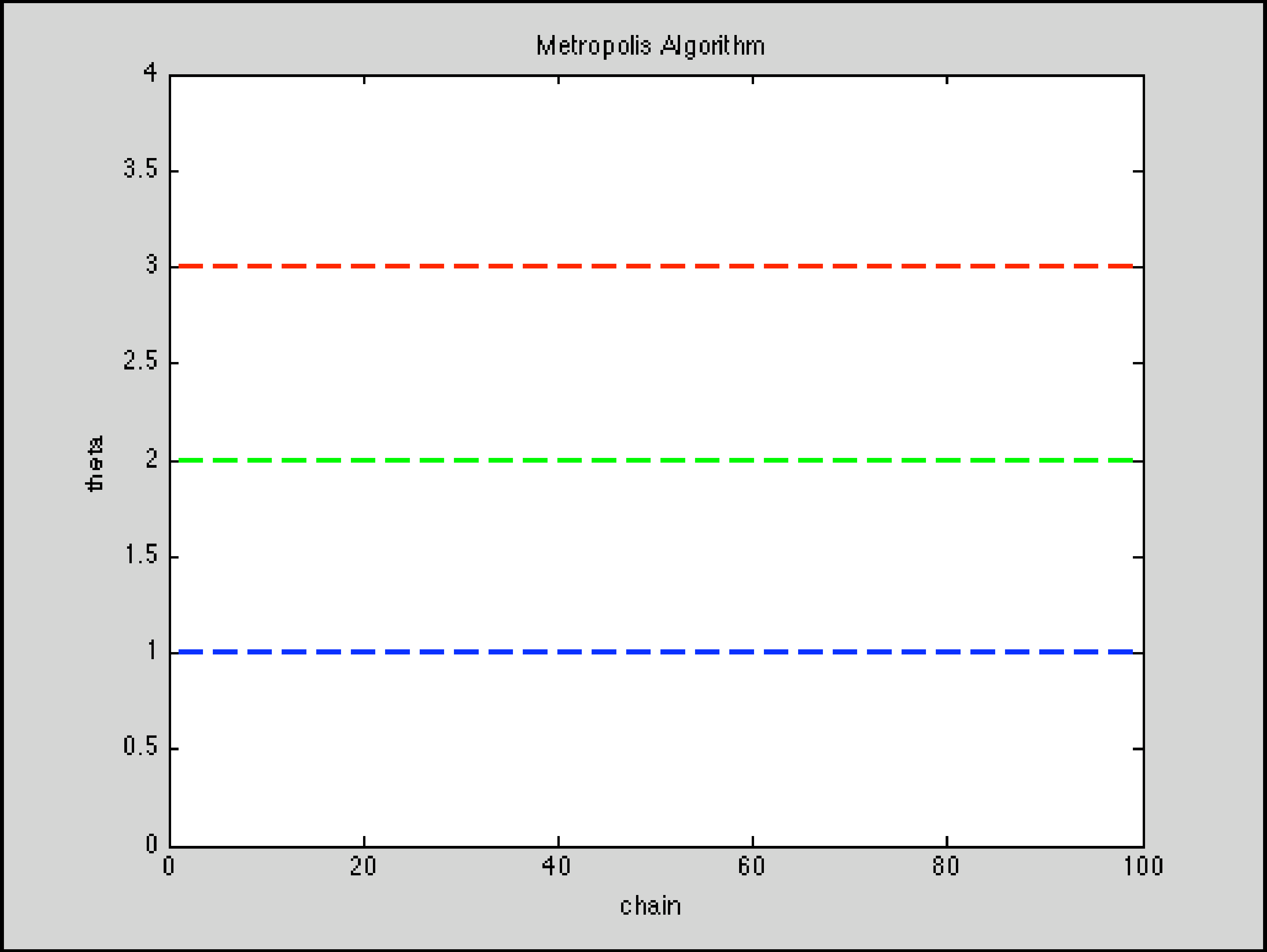
Metropolis Algorithm



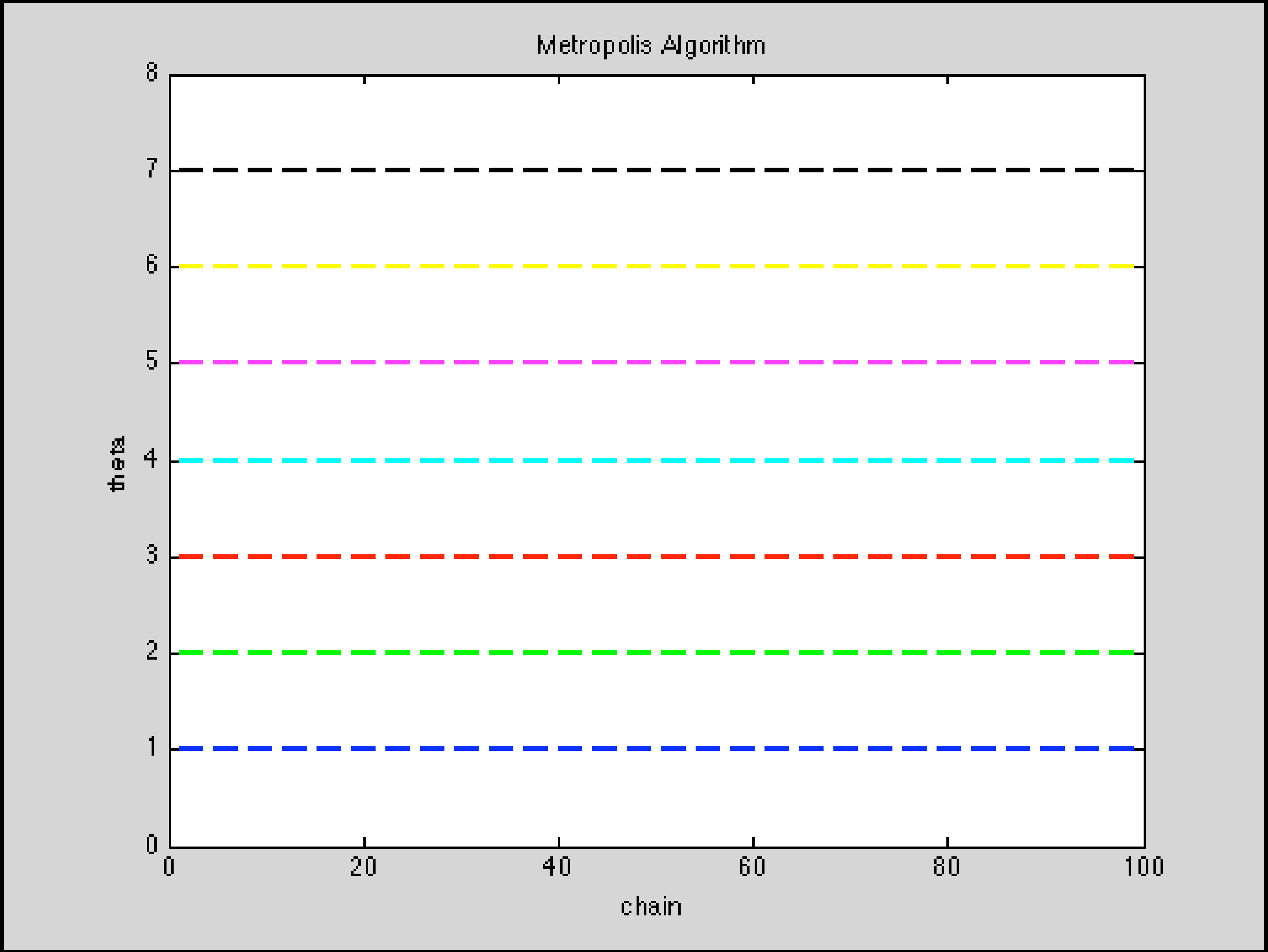
Metropolis Algorithm



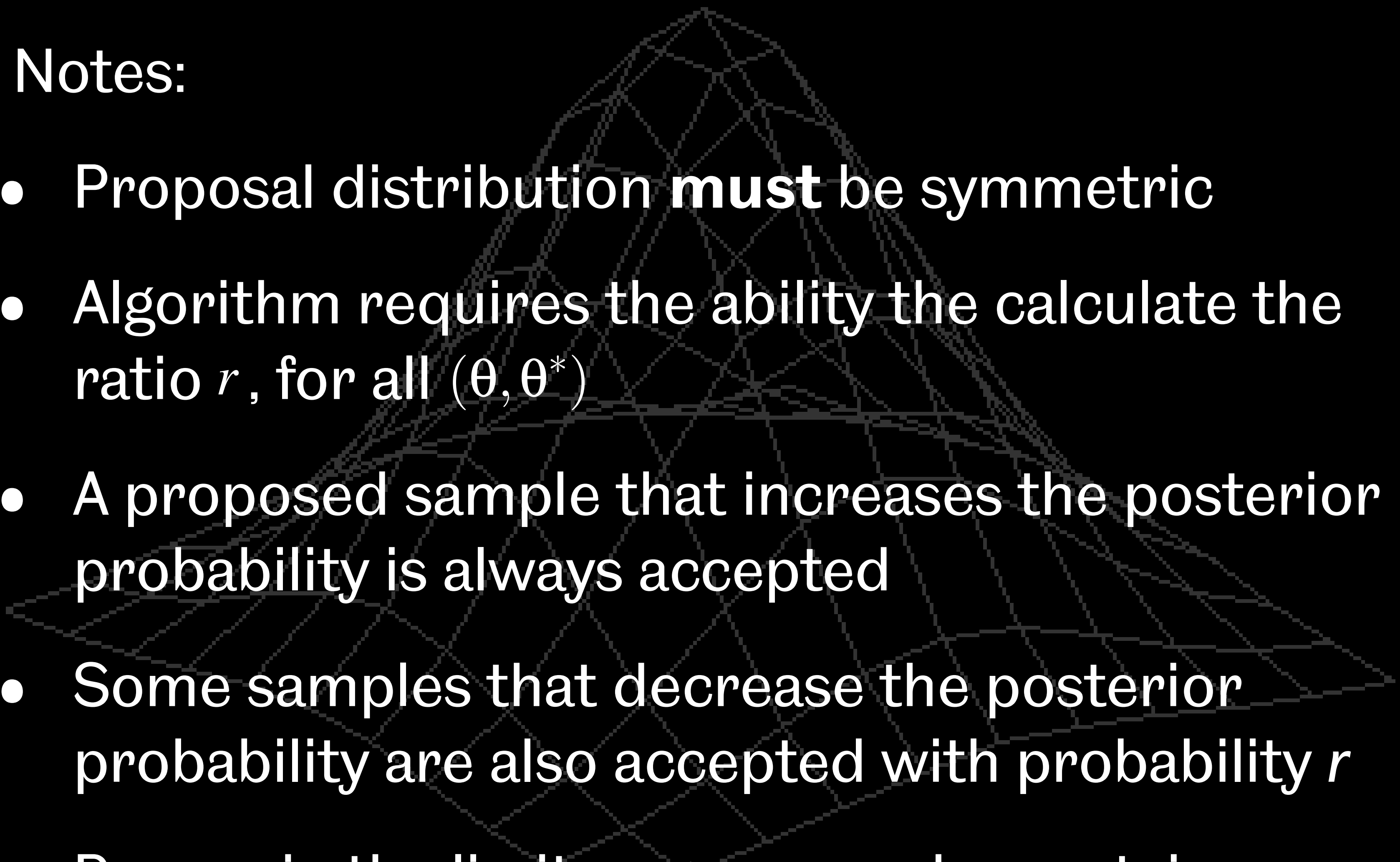
Metropolis Algorithm



Metropolis Algorithm



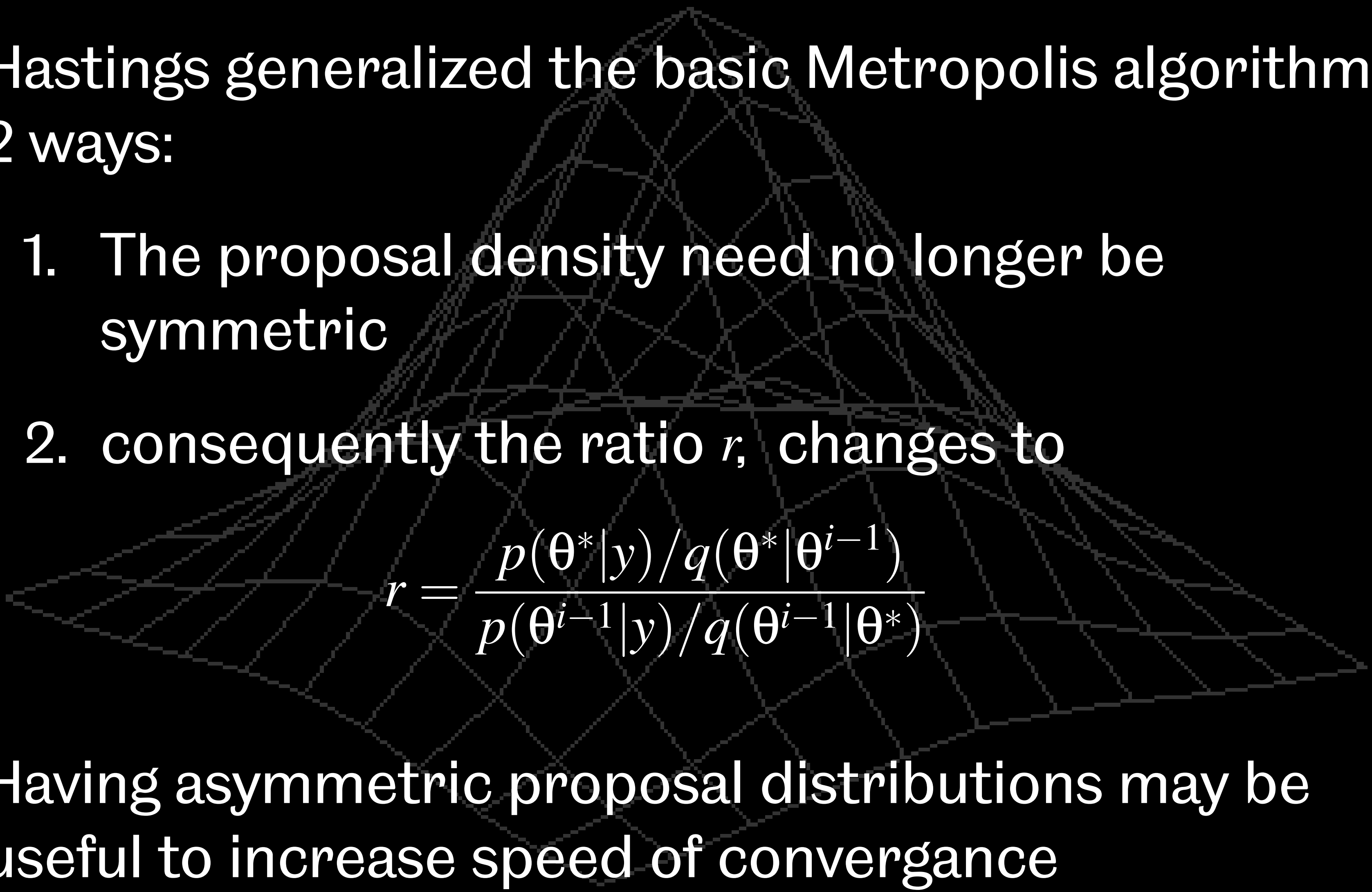
Metropolis Algorithm



Notes:

- Proposal distribution **must** be symmetric
- Algorithm requires the ability the calculate the ratio r, for all (θ, θ^*)
- A proposed sample that increases the posterior probability is always accepted
- Some samples that decrease the posterior probability are also accepted with probability *r*
- Proven in the limit as $i \rightarrow \infty$ under certain conditions

MH Algorithm



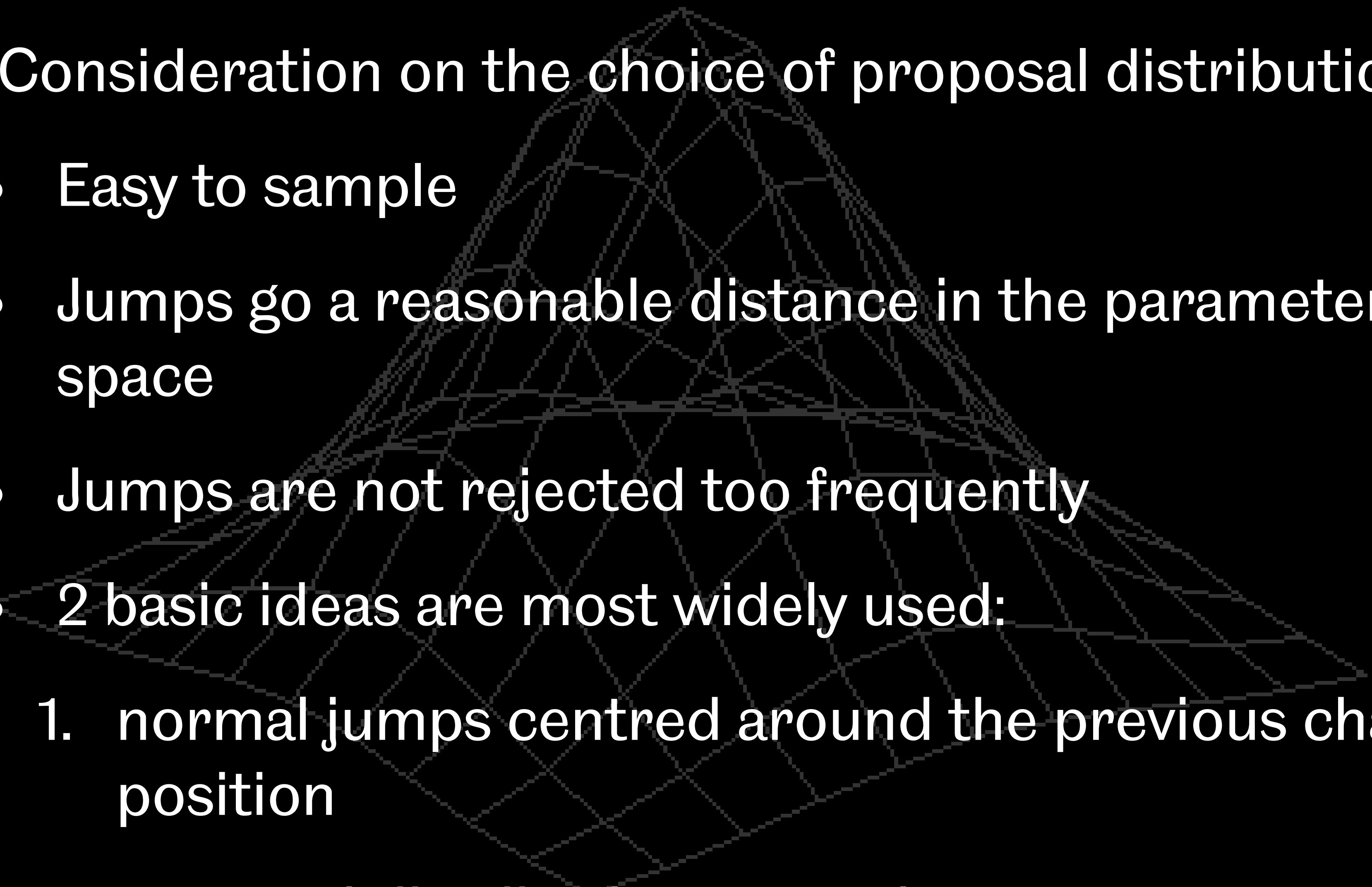
Hastings generalized the basic Metropolis algorithm in 2 ways:

1. The proposal density need no longer be symmetric
2. consequently the ratio r, changes to

$$r = \frac{p(\theta^{*}|y)/q(\theta^{*}|\theta^{i-1})}{p(\theta^{i-1}|y)/q(\theta^{i-1}|\theta^{*})}$$

Having asymmetric proposal distributions may be useful to increase speed of convergance

MH Algorithm

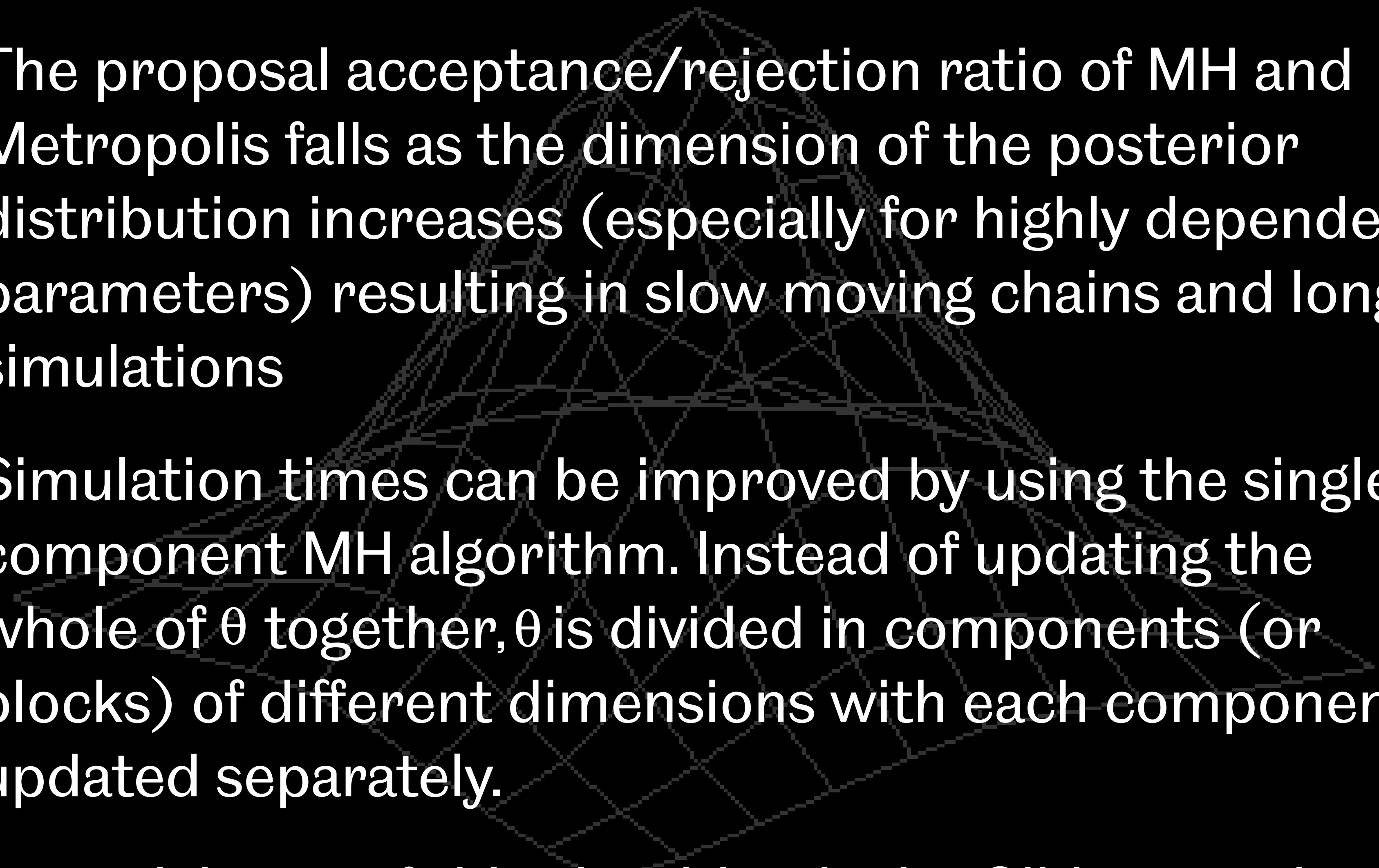


Consideration on the choice of proposal distribution:

- Easy to sample
- Jumps go a reasonable distance in the parameter space
- Jumps are not rejected too frequently
- 2 basic ideas are most widely used:

1. normal jumps centred around the previous chain position
2. proposal distributions very close to target distribution; resulting in high acceptance ratios

Single Component MH

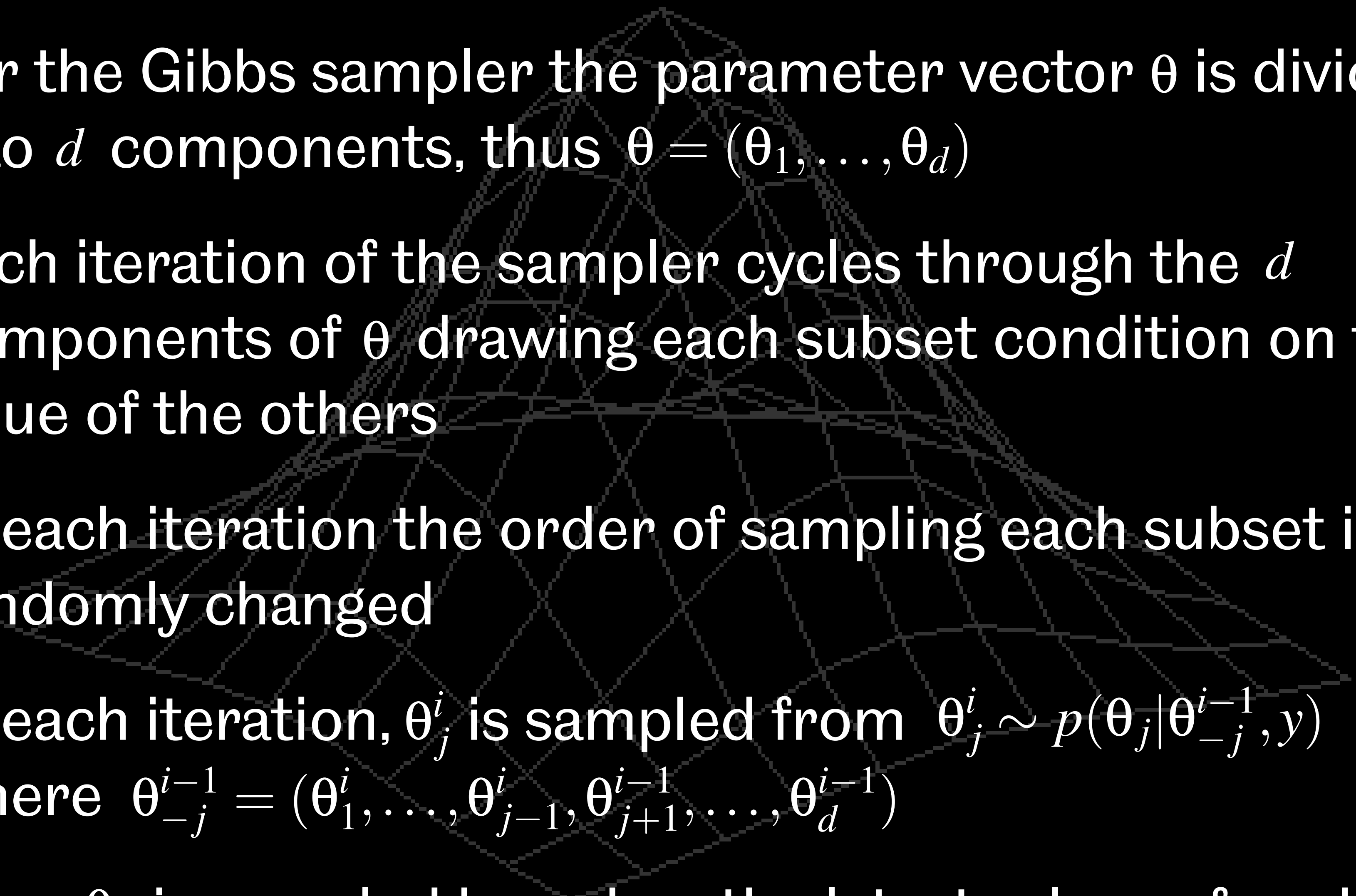


The proposal acceptance/rejection ratio of MH and Metropolis falls as the dimension of the posterior distribution increases (especially for highly dependent parameters) resulting in slow moving chains and long simulations

Simulation times can be improved by using the single component MH algorithm. Instead of updating the whole of θ together, θ is divided in components (or blocks) of different dimensions with each component updated separately.

A special case of this algorithm is the Gibbs sampler

Gibbs Sampler



For the Gibbs sampler the parameter vector θ is divided into d components, thus $\theta = (\theta_1, \ldots, \theta_d)$

Each iteration of the sampler cycles through the d components of θ drawing each subset condition on the value of the others

At each iteration the order of sampling each subset is randomly changed

At each iteration, θ_j^i is sampled from $\theta_j^i \sim p(\theta_j | \theta_{-j}^{i-1}, y)$ where $\theta_{-j}^{i-1} = (\theta_1^i, \ldots, \theta_{j-1}^i, \theta_{j+1}^{i-1}, \ldots, \theta_d^{i-1})$

Thus θ_j is sampled based on the latest values of each component of θ

Gibbs Sampler

Example 3. Consider the model with an observable vector y of d components

$$y|\theta,\Sigma \sim N(\theta,\Sigma)$$

where

μ is a vector of means

Σ is the variance matrix

Assuming a normal prior distribution for θ, given by

$$\theta \sim N(\theta_0,\Sigma_0)$$

Gibbs Sampler

The marginal conditional distribution required for the Gibbs sampler is given by

$$\theta^{(1)}|\theta^{(2)}, y \sim N\big(\theta_n^{(1)} + \beta^{1|2}(\theta^{(2)} - \theta_n^{(2)}), \Sigma^{1|2}\big)$$

where

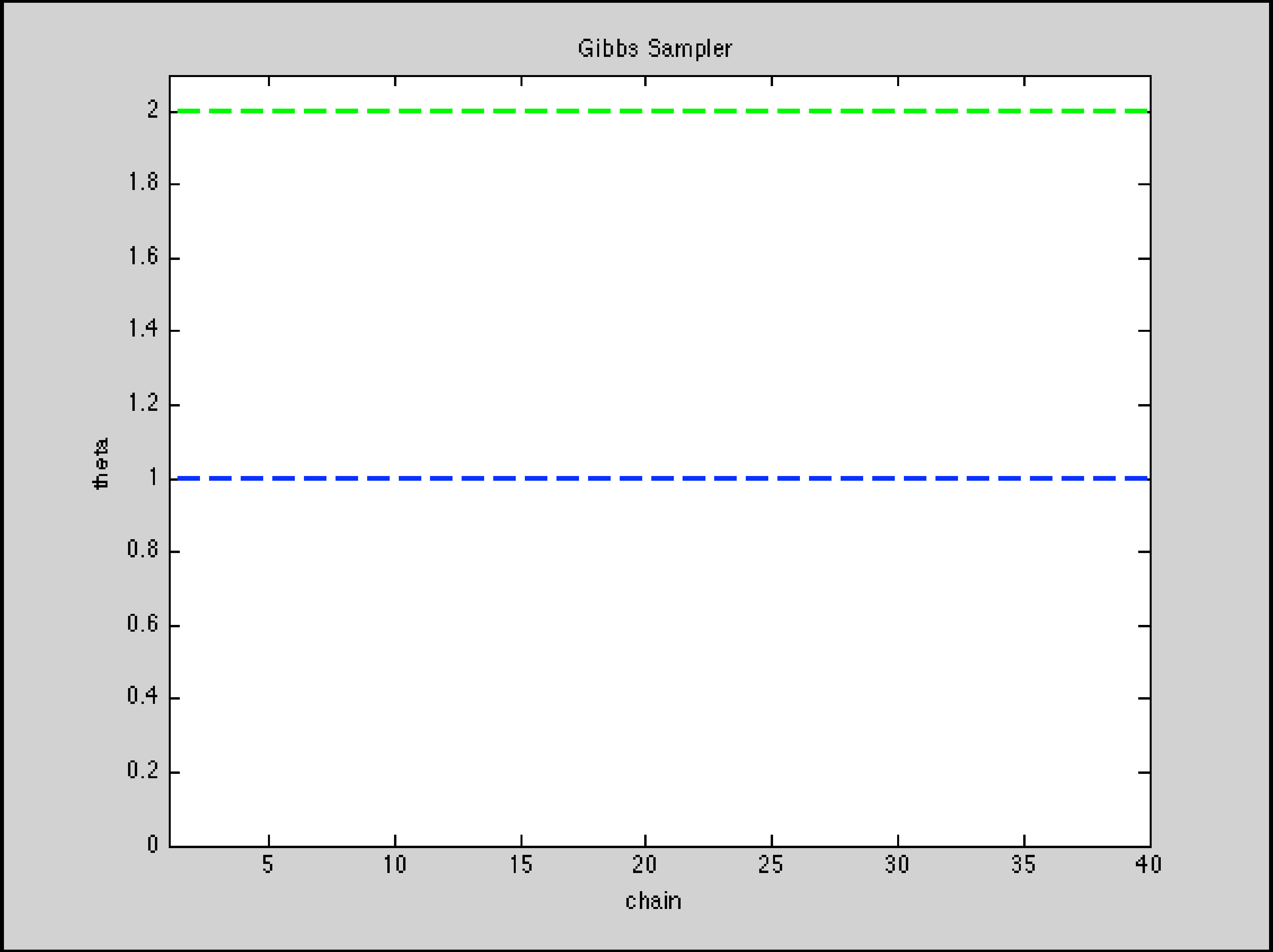
$$\Sigma_n^{-1} = \Sigma_0^{-1} + n\Sigma^{-1}$$

$$\theta_n = (\Sigma_0^{-1} + n\Sigma^{-1})^{-1}(\Sigma_0^{-1}\theta_0 + n\Sigma^{-1}\bar{y})$$

$$\beta^{1|2} = \Sigma_n^{(12)}(\Sigma_n^{(22)})^{-1}$$

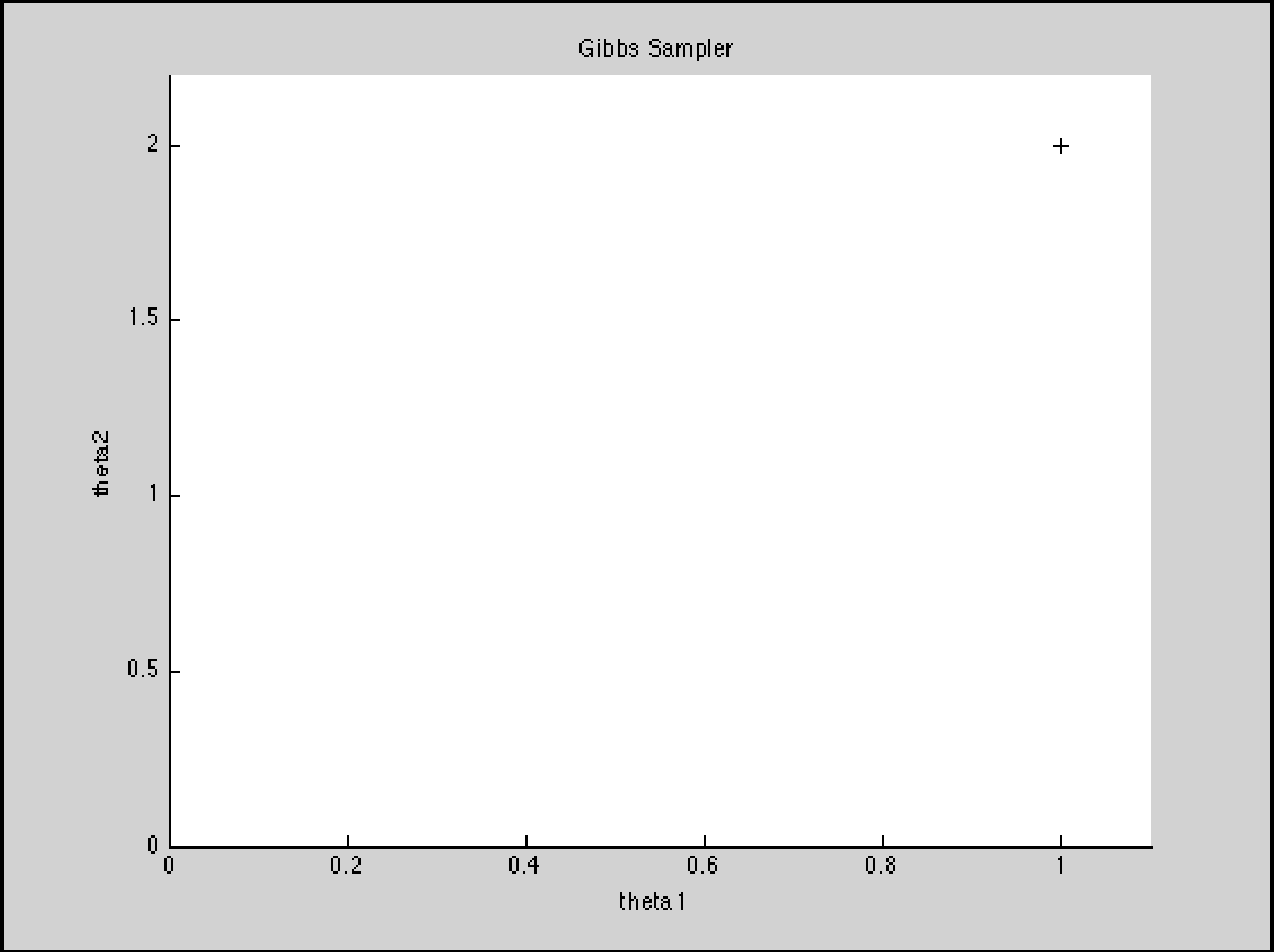
$$\Sigma^{1|2} = \Sigma_n^{(11)} - \Sigma_n^{(12)}(\Sigma_n^{(22)})^{-1}\Sigma_n^{(21)}$$

Gibbs Sampler



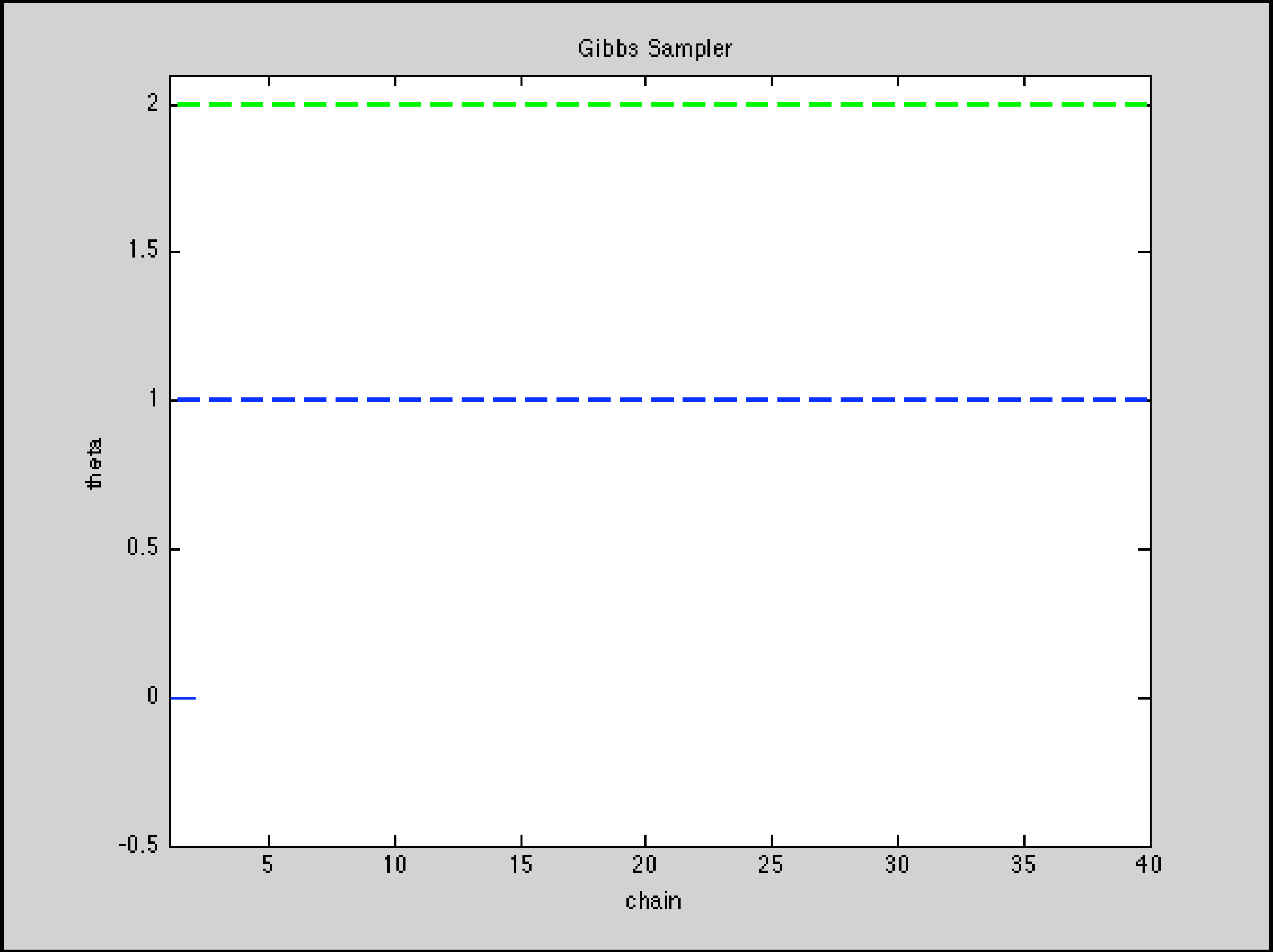
No correlation between parameters

Gibbs Sampler



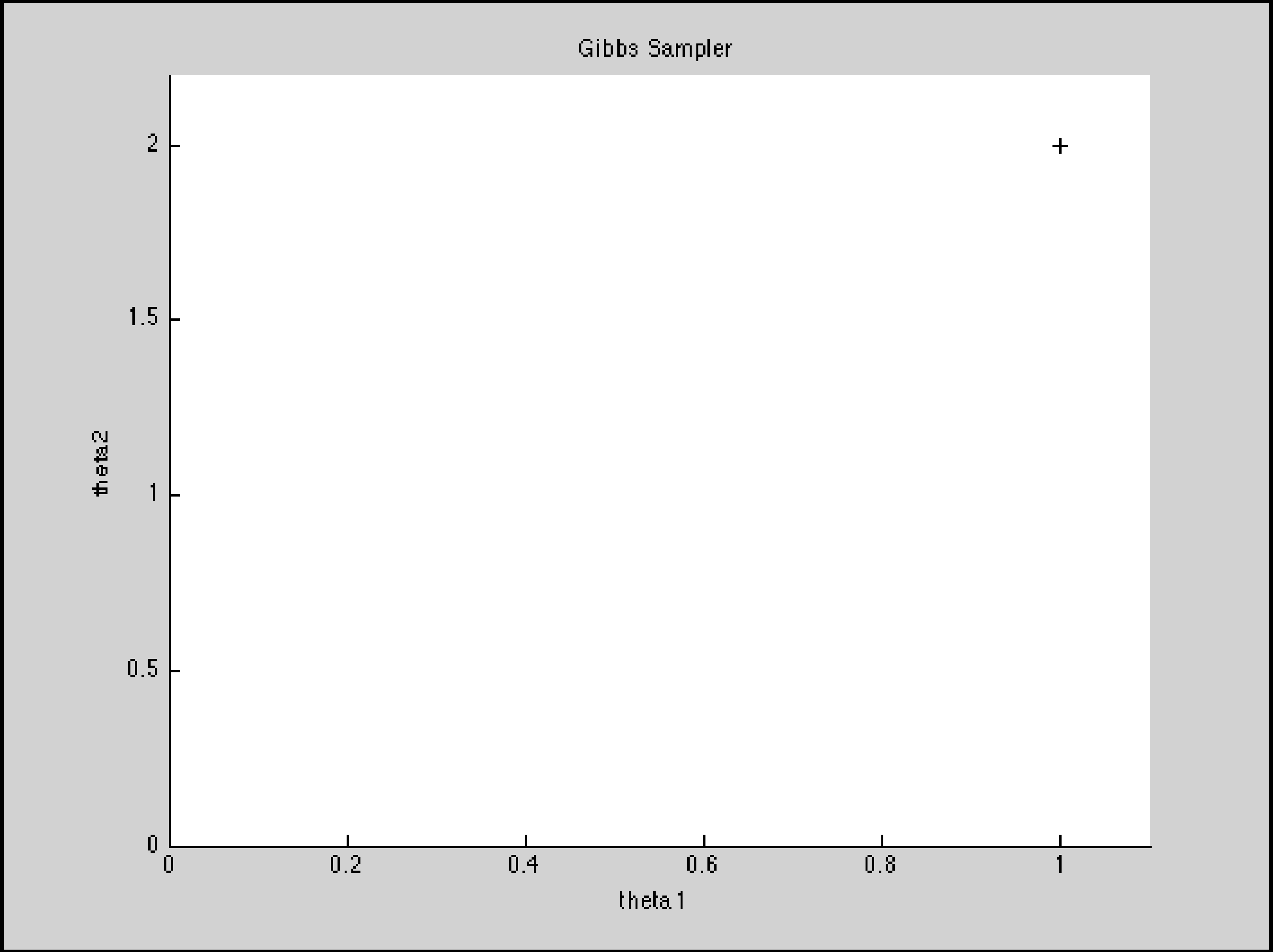
No correlation between parameters

Gibbs Sampler



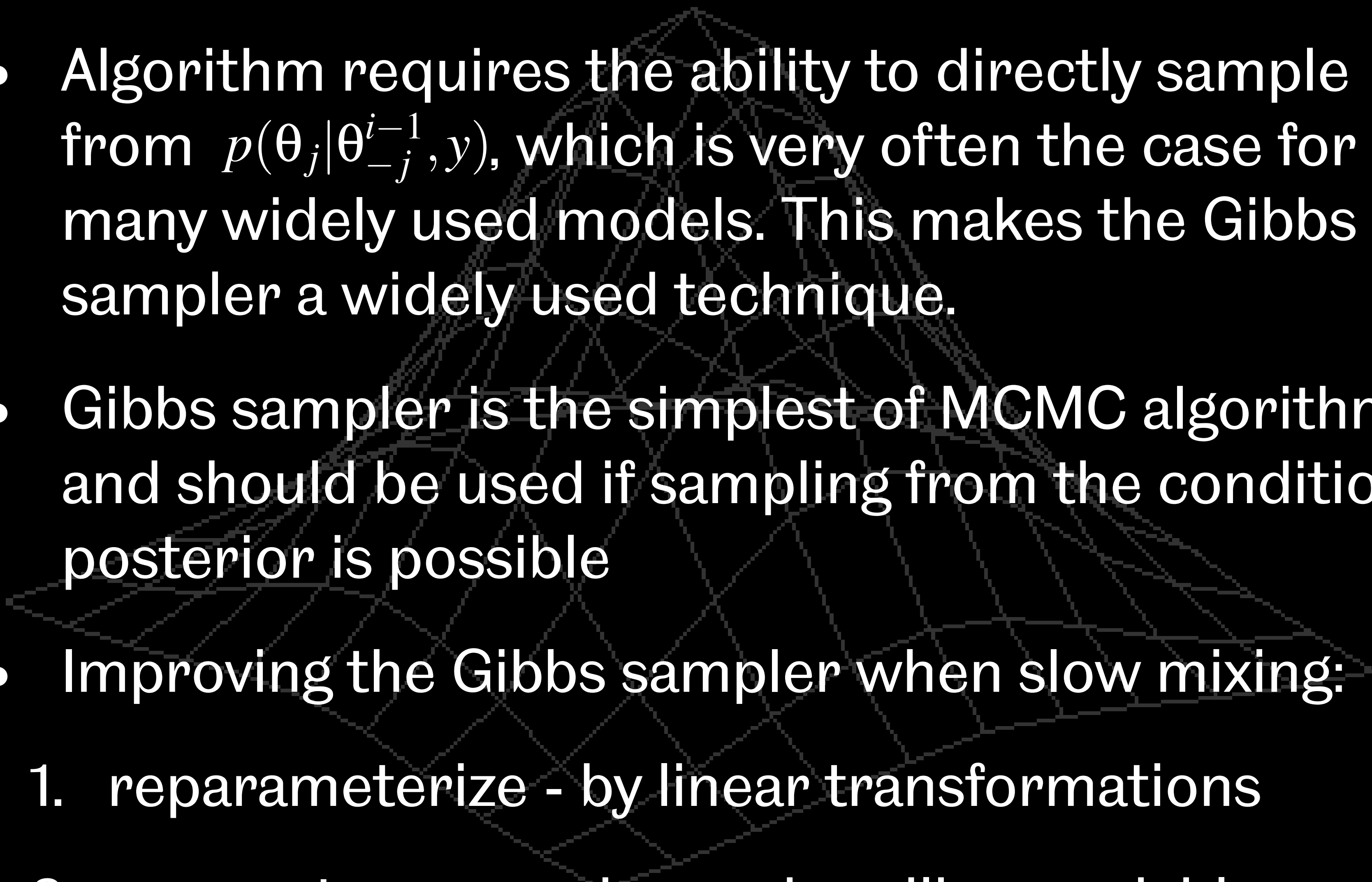
Correlated parameters

Gibbs Sampler



Correlated parameters

Gibbs Sampler



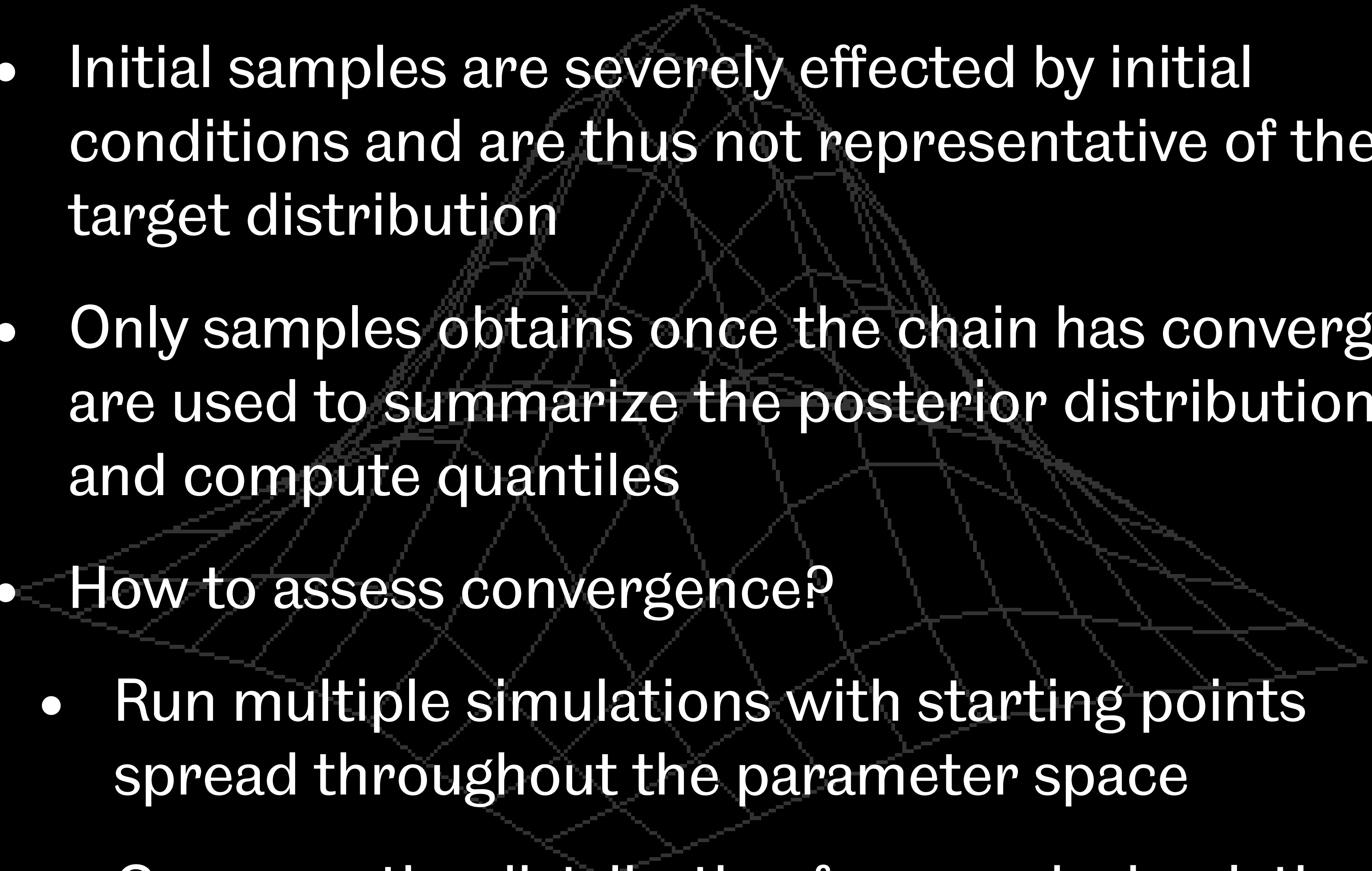
- Algorithm requires the ability to directly sample from $p(\theta_j|\theta_{-j}^{i-1}, y)$, which is very often the case for many widely used models. This makes the Gibbs sampler a widely used technique.

- Gibbs sampler is the simplest of MCMC algorithms and should be used if sampling from the conditional posterior is possible

- Improving the Gibbs sampler when slow mixing:

1. reparameterize - by linear transformations
2. parameter expansion and auxiliary variables
3. better blocking

Inference



- Initial samples are severely effected by initial conditions and are thus not representative of the target distribution

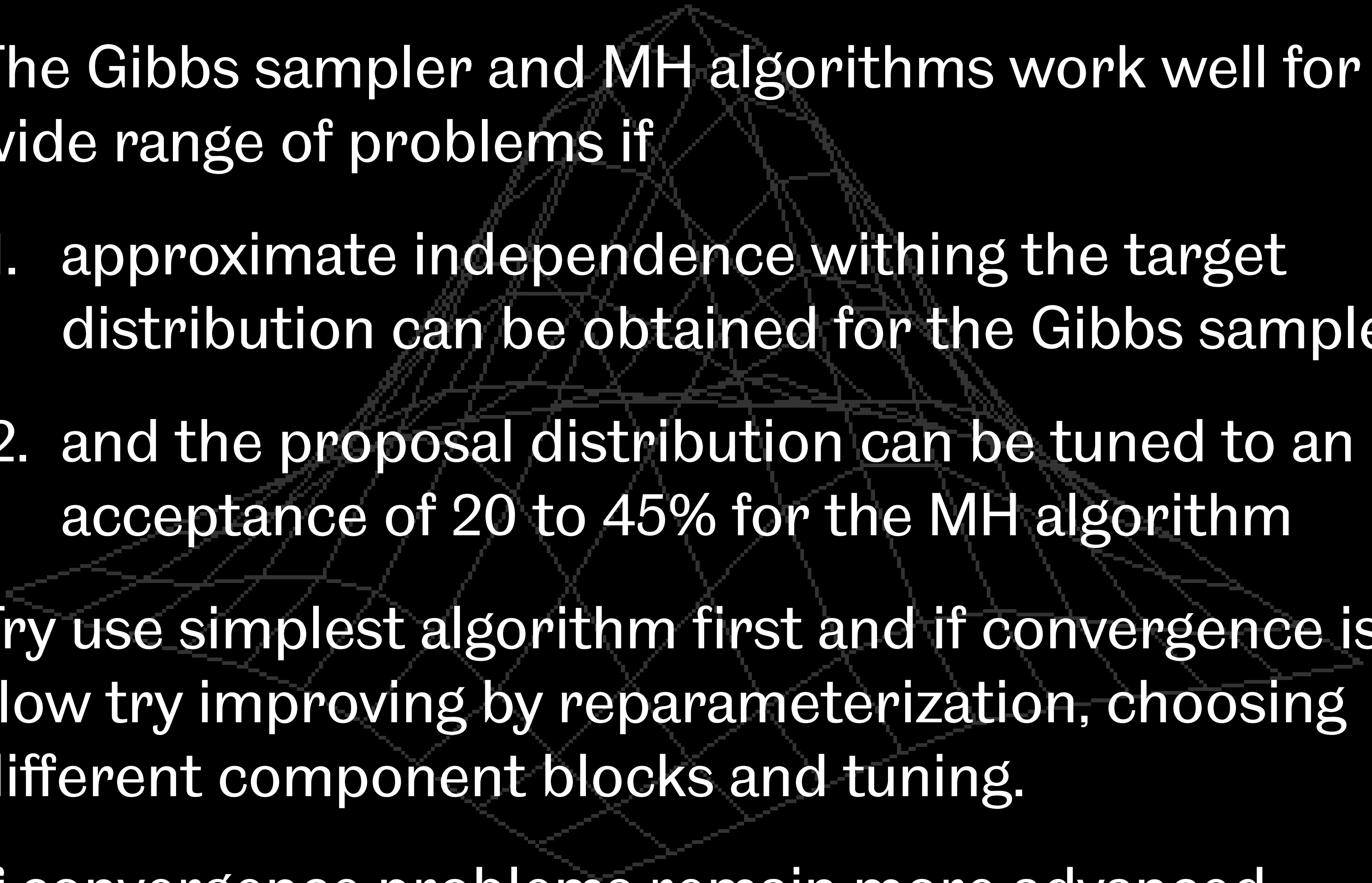
- Only samples obtains once the chain has converged are used to summarize the posterior distribution and compute quantiles

- How to assess convergence?

  - Run multiple simulations with starting points spread throughout the parameter space

  - Compare the distribution from each simulation to the mixed results

Recommendation



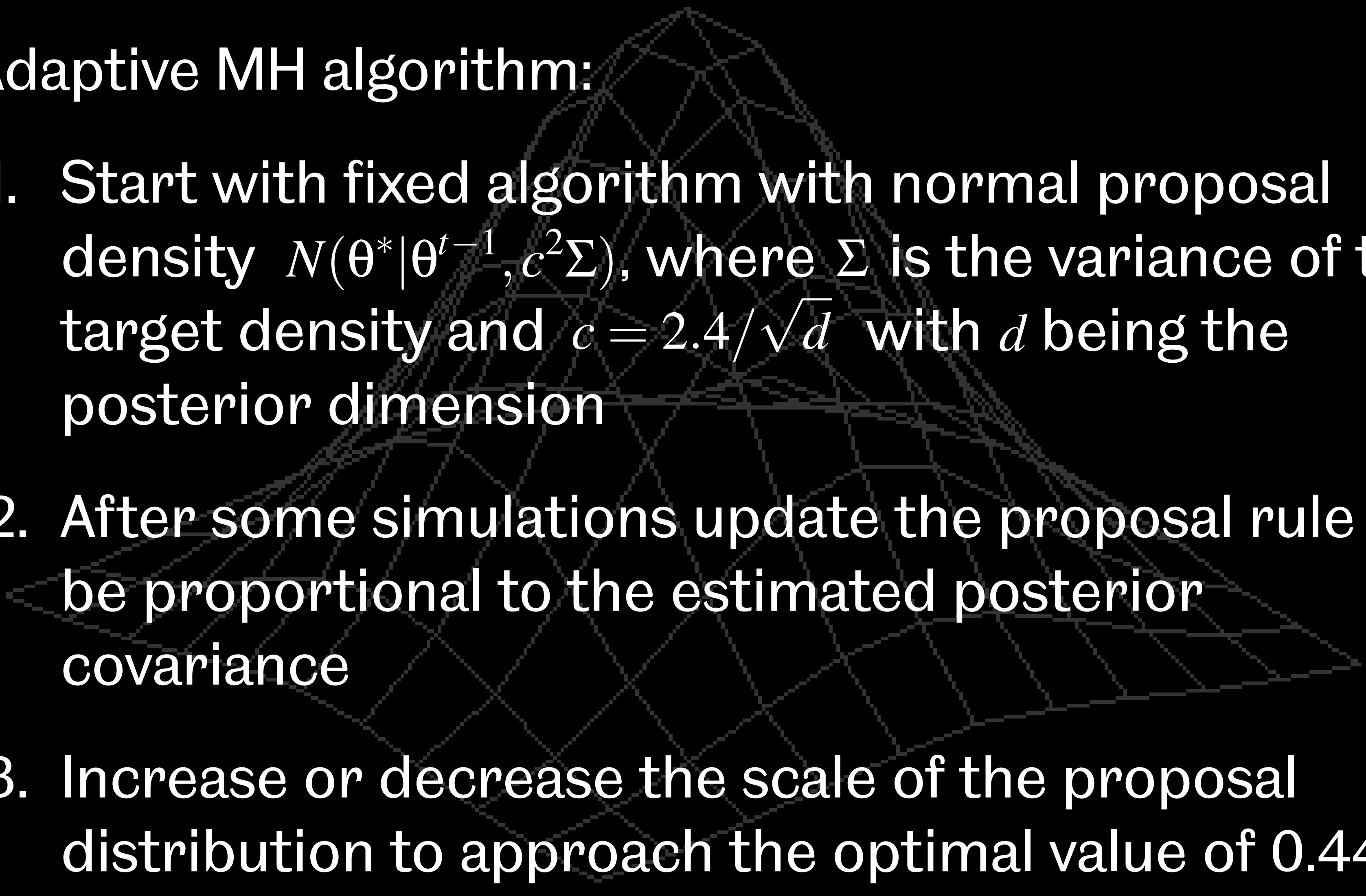
The Gibbs sampler and MH algorithms work well for a wide range of problems if

1. approximate independence withing the target distribution can be obtained for the Gibbs sampler
2. and the proposal distribution can be tuned to an acceptance of 20 to 45% for the MH algorithm

Try use simplest algorithm first and if convergence is slow try improving by reparameterization, choosing different component blocks and tuning.

If convergence problems remain more advanced algorithms should be used

MH Algorithm Variations



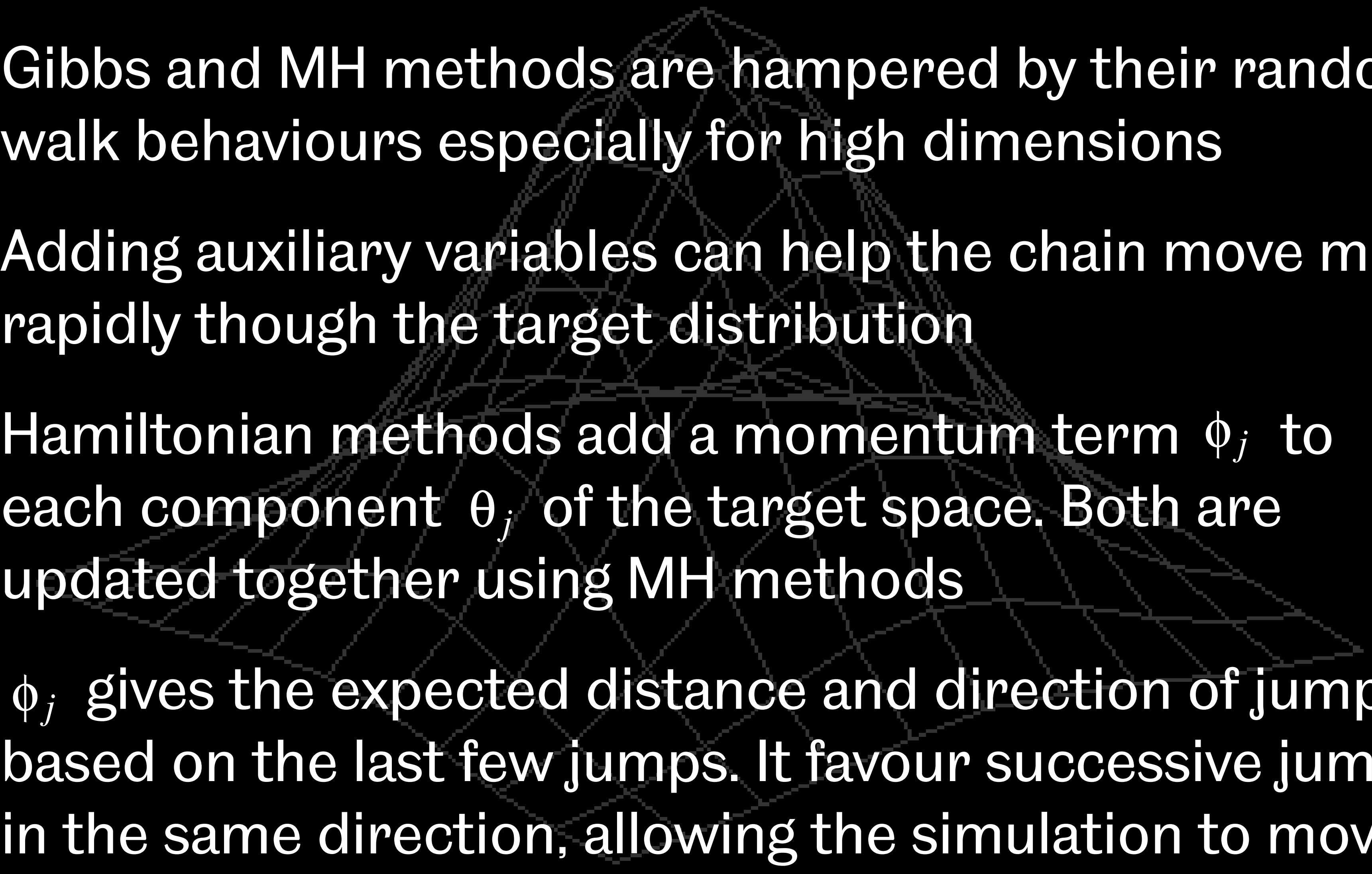
Adaptive MH algorithm:

1. Start with fixed algorithm with normal proposal density $N(\theta^*|\theta^{t-1}, c^2\Sigma)$, where Σ is the variance of the target density and $c = 2.4/\sqrt{d}$ with d being the posterior dimension

2. After some simulations update the proposal rule to be proportional to the estimated posterior covariance

3. Increase or decrease the scale of the proposal distribution to approach the optimal value of 0.44 (for one dimension) or 0.23 (for multidimensional jumps)

Hamiltonian Monte Carlo



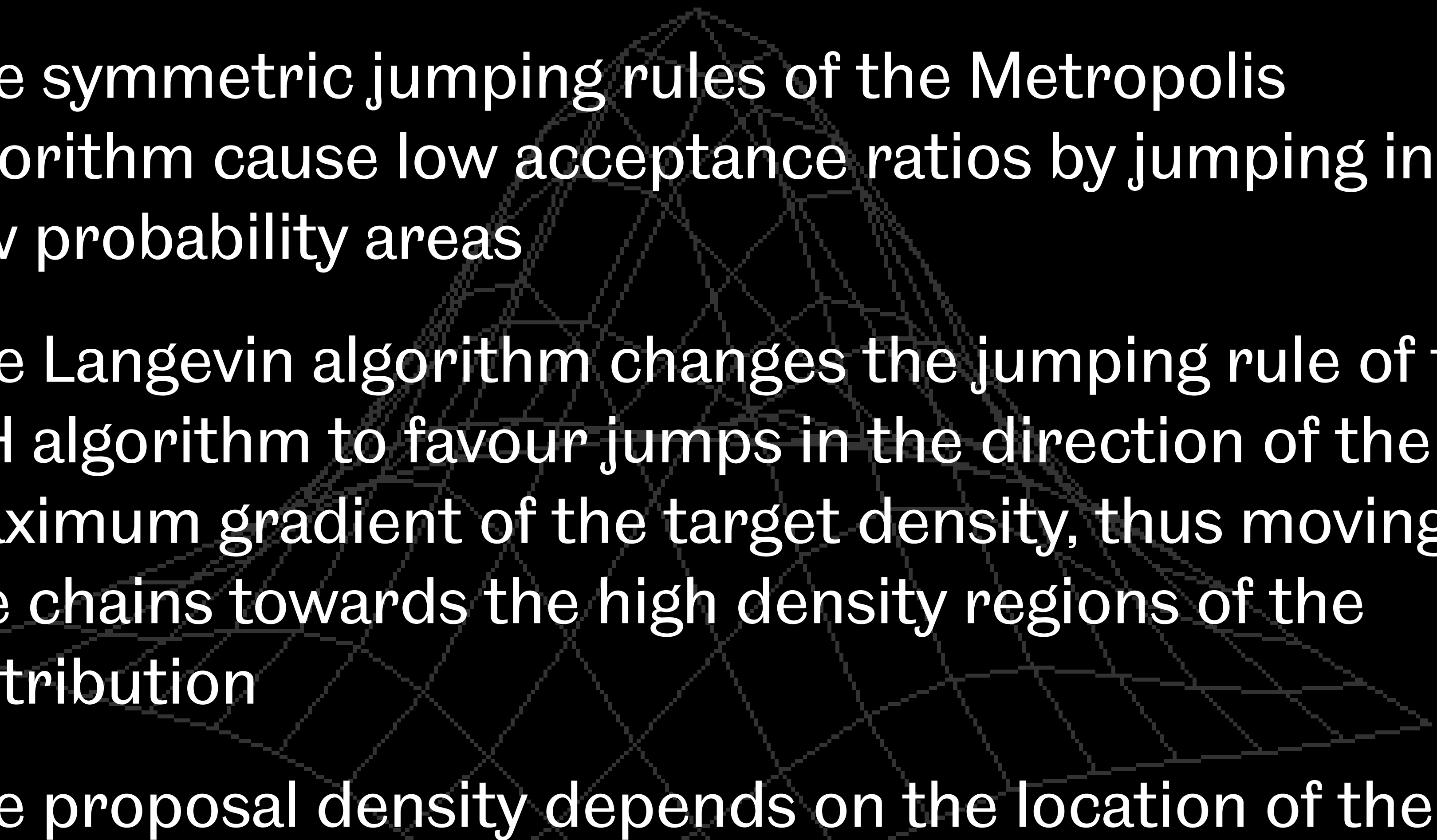
Gibbs and MH methods are hampered by their random walk behaviours especially for high dimensions

Adding auxiliary variables can help the chain move more rapidly though the target distribution

Hamiltonian methods add a momentum term ϕ_j to each component θ_j of the target space. Both are updated together using MH methods

ϕ_j gives the expected distance and direction of jump of θ_j based on the last few jumps. It favour successive jump in the same direction, allowing the simulation to move rapidly through the space of θ_j

Langevin Rule

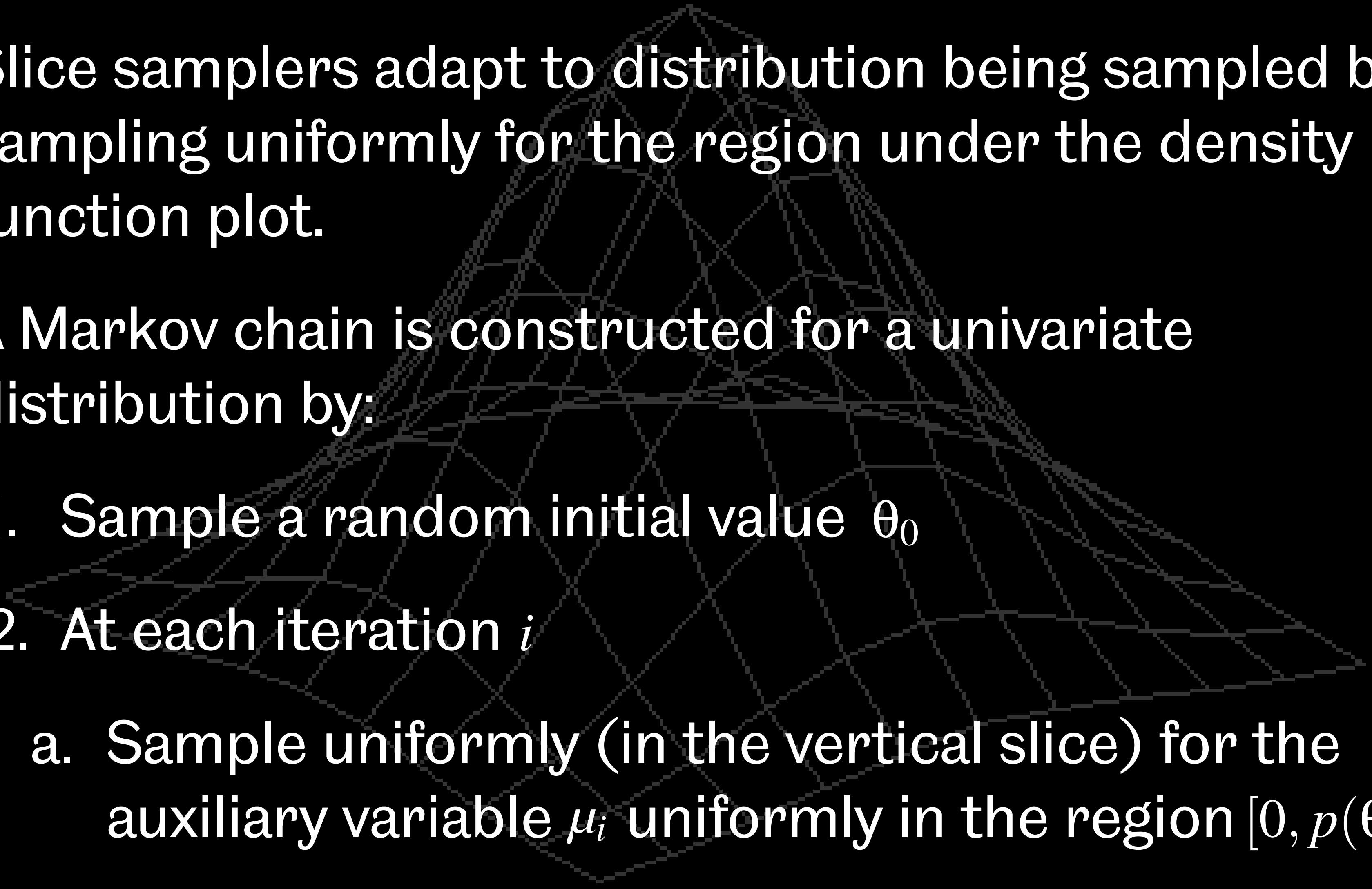


The symmetric jumping rules of the Metropolis algorithm cause low acceptance ratios by jumping in low probability areas

The Langevin algorithm changes the jumping rule of the MH algorithm to favour jumps in the direction of the maximum gradient of the target density, thus moving the chains towards the high density regions of the distribution

The proposal density depends on the location of the current sample and this is not symmetric. The MH algorithm is thus required

Slice Sampler

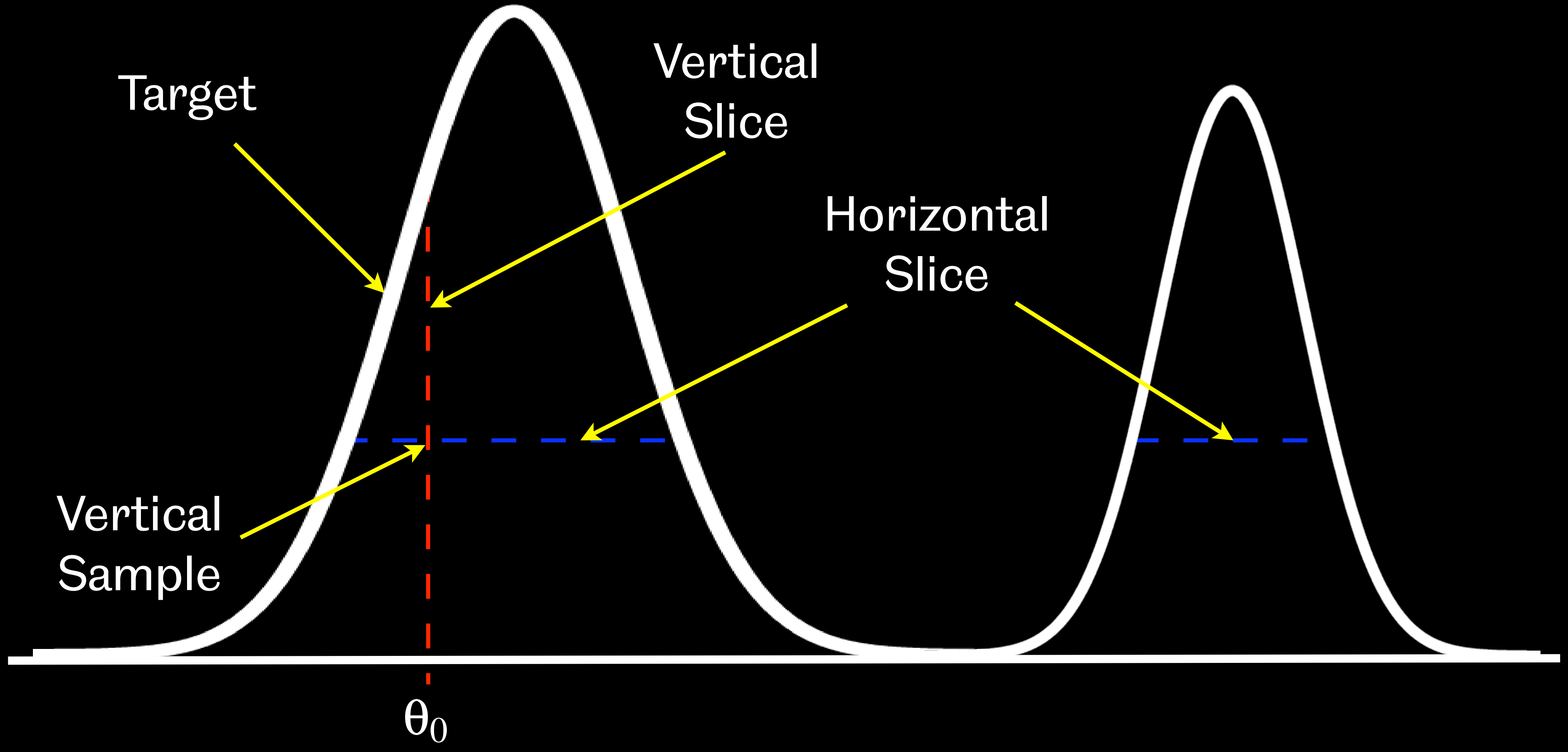


Slice samplers adapt to distribution being sampled by sampling uniformly for the region under the density function plot.

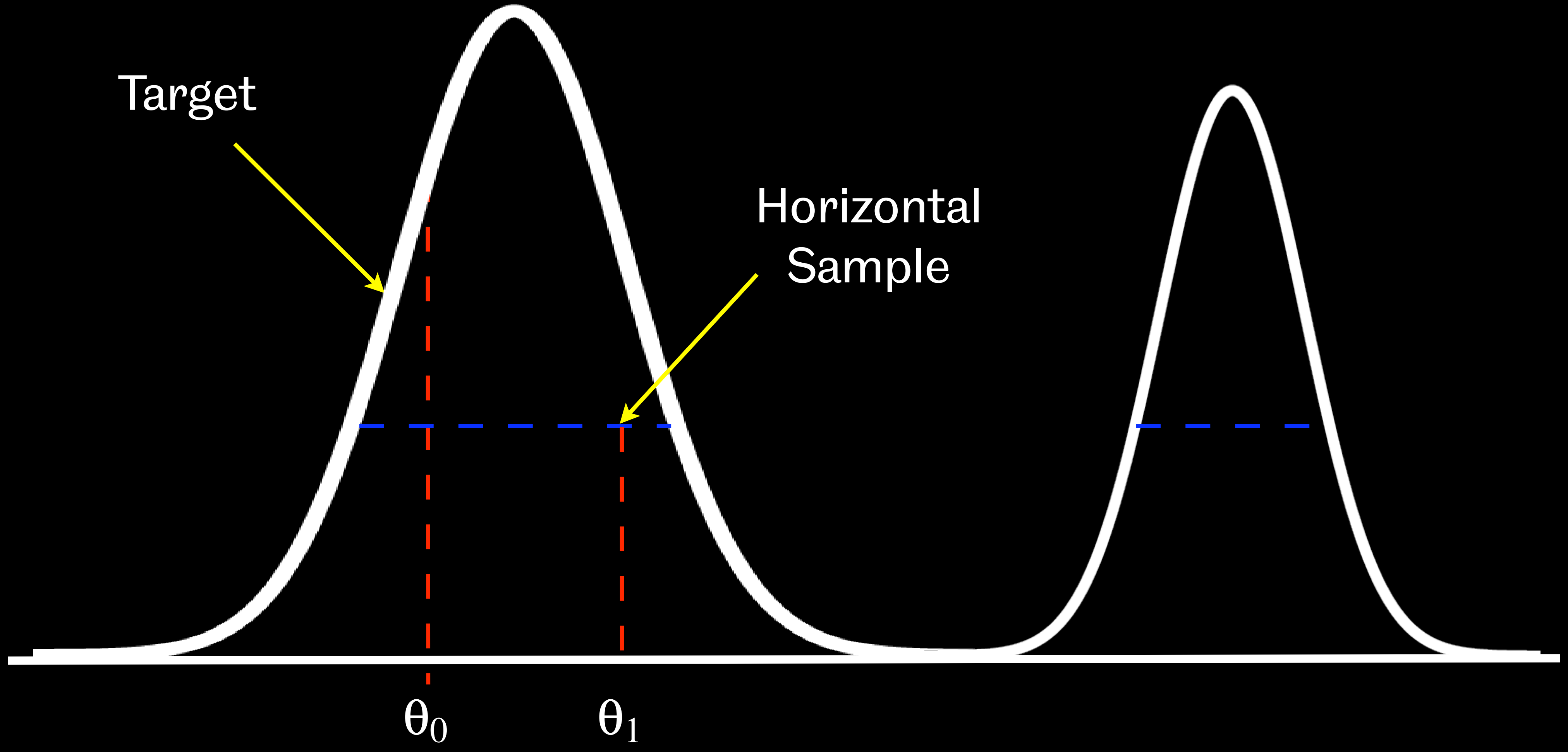
A Markov chain is constructed for a univariate distribution by:

1. Sample a random initial value θ_0
2. At each iteration i
    a. Sample uniformly (in the vertical slice) for the auxiliary variable μ_i uniformly in the region $[0, p(\theta_{i-1}|y)]$
    b. Sample uniformly (in the horizontal slice) for θ_i

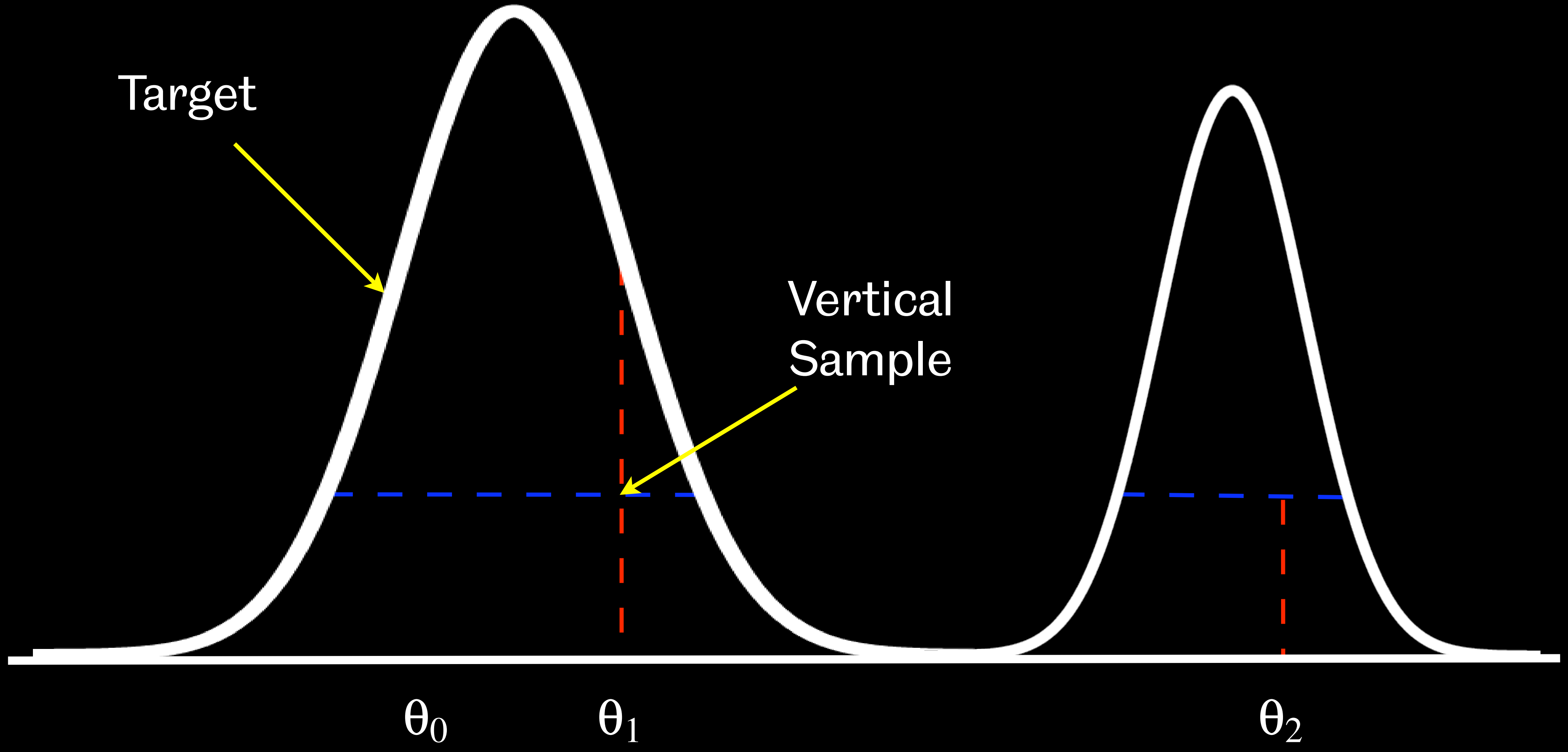
Slice Sampler



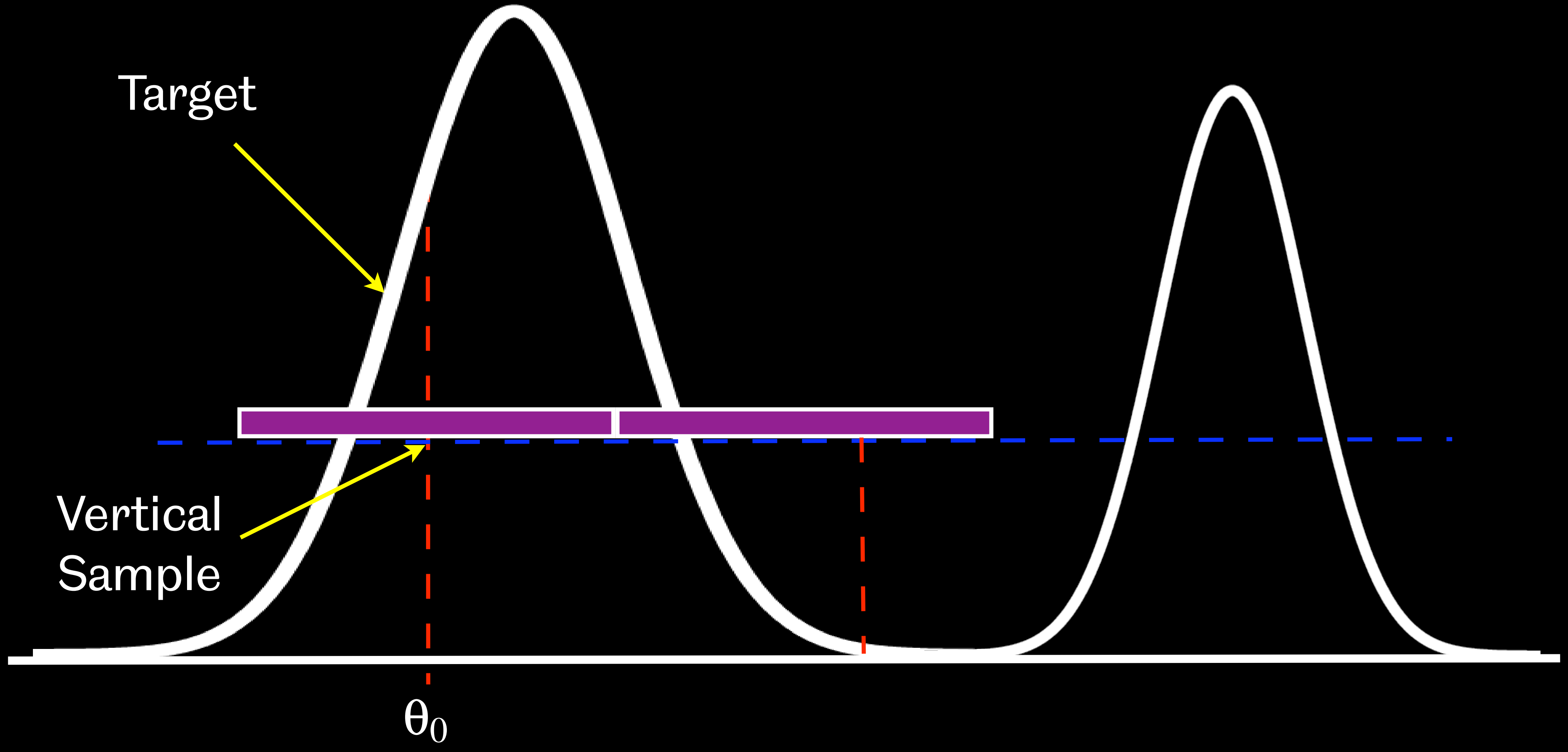
Slice Sampler



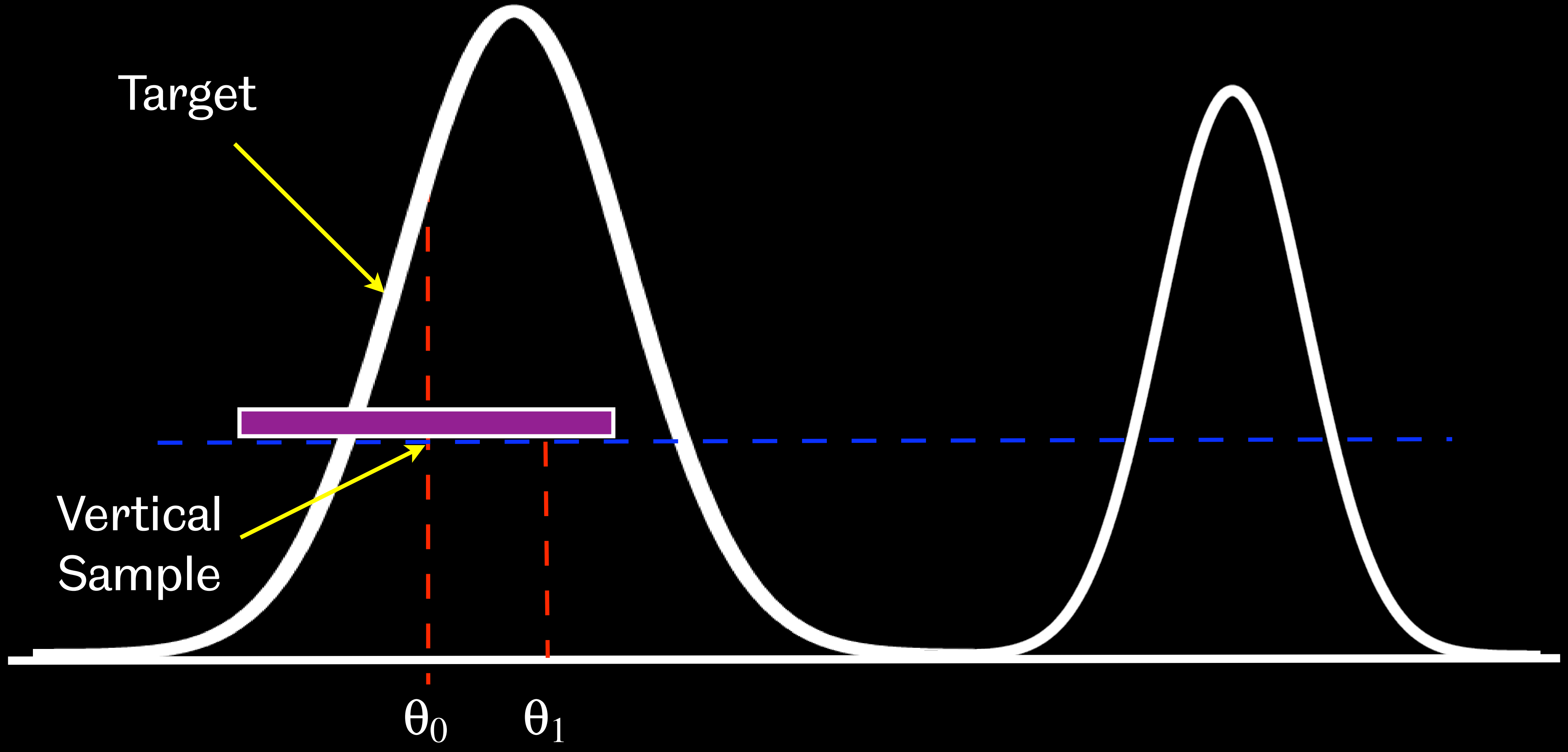
Slice Sampler



Slice Sampler



Slice Sampler

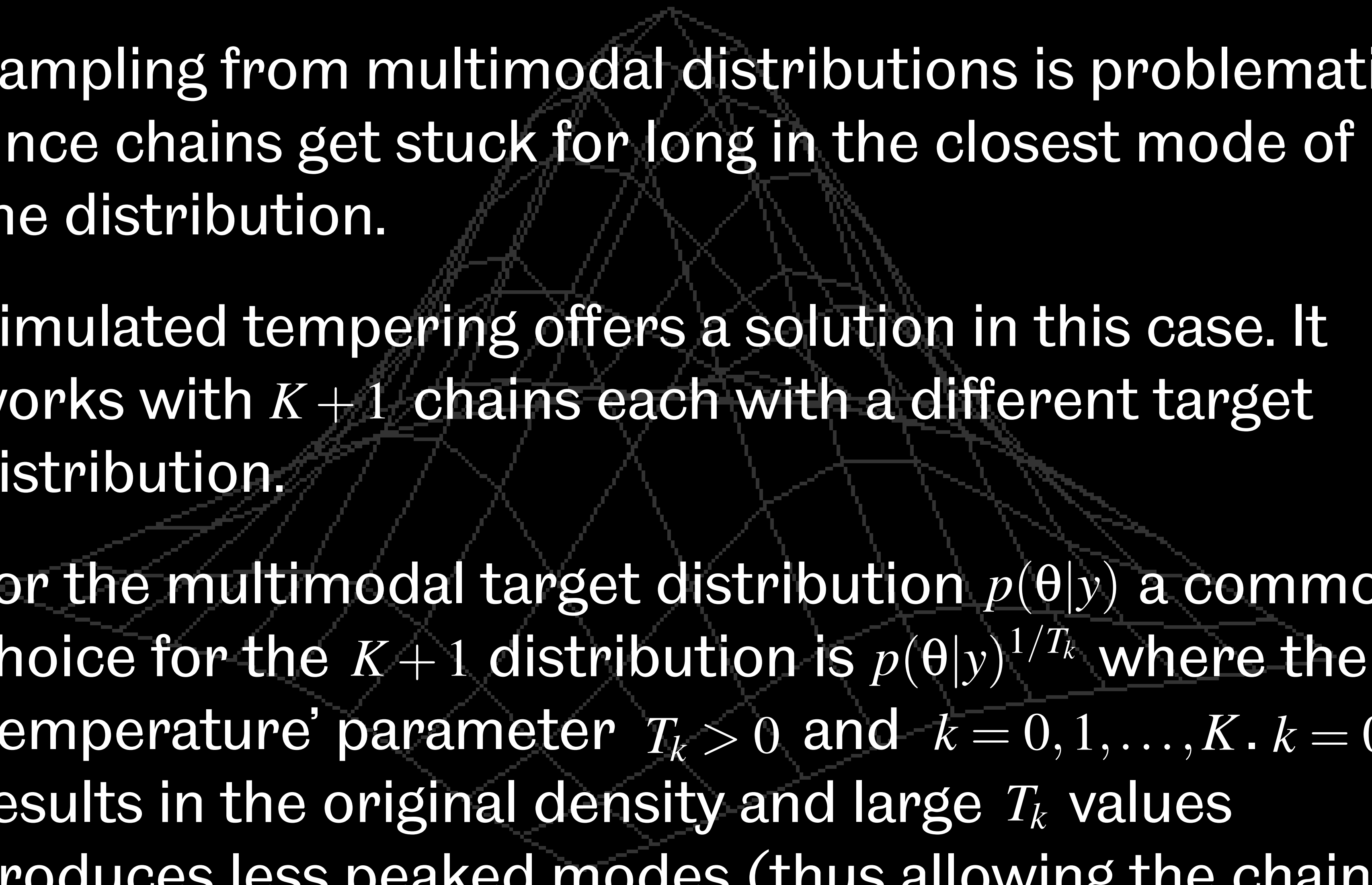


Slice Sampler

Notes:

- The slice sampler can be used for multivariate distributions by sampling one-dimensional conditional distributions in a Gibbs structure
- As opposed to the Gibbs sampler it does not require to sampled for the marginal conditional distribution but needs to be able to evaluate them
- The slice sampler can deal with distributions exhibiting multiple modes if the modes are not separated by large regions of low probability

Simulated Tempering

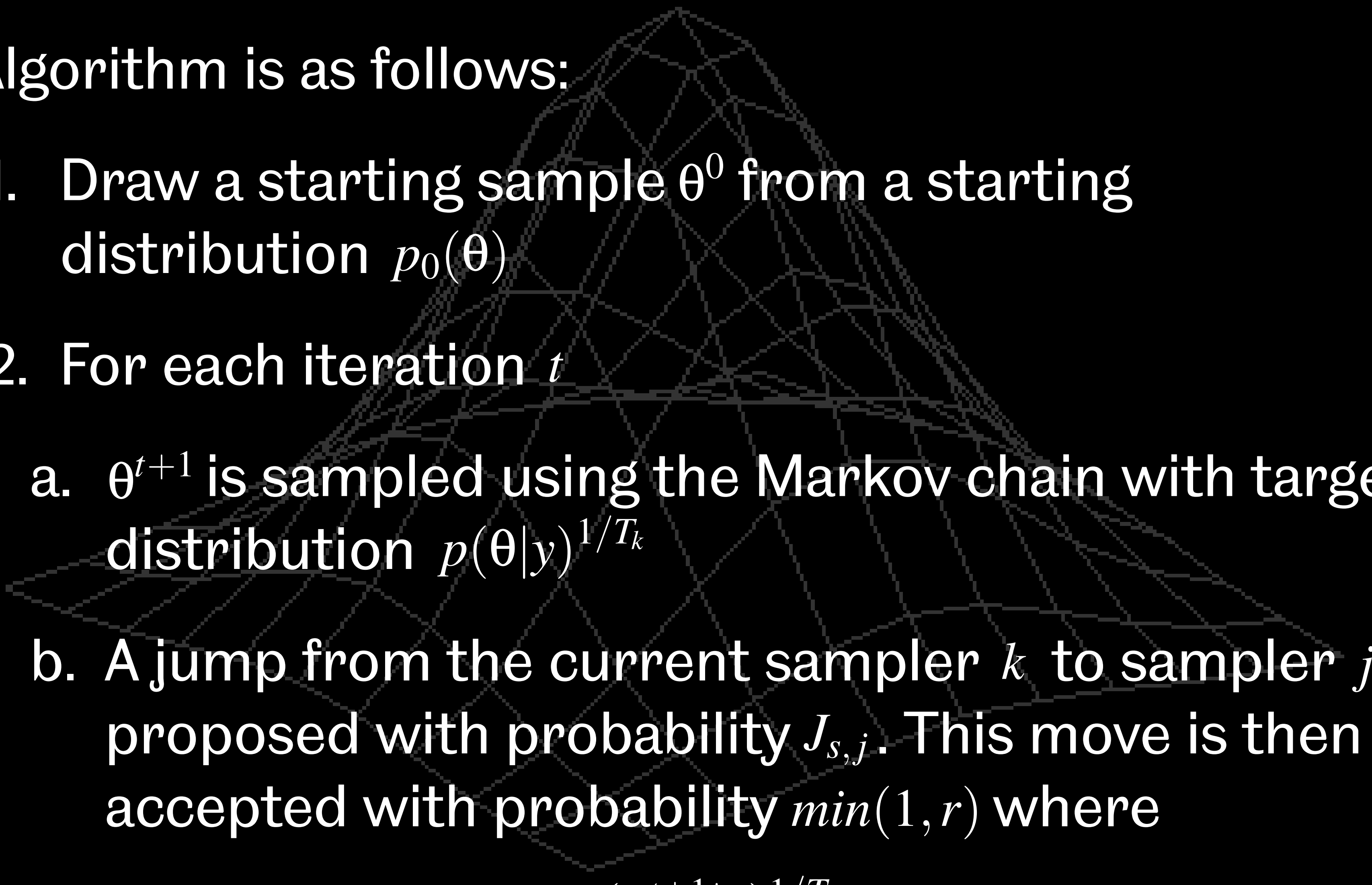


Sampling from multimodal distributions is problematic since chains get stuck for long in the closest mode of the distribution.

Simulated tempering offers a solution in this case. It works with $K+1$ chains each with a different target distribution.

For the multimodal target distribution $p(\theta|y)$ a common choice for the $K+1$ distribution is $p(\theta|y)^{1/T_k}$ where the 'temperature' parameter $T_k > 0$ and $k = 0, 1, \ldots, K$. $k = 0$ results in the original density and large T_k values produces less peaked modes (thus allowing the chain to move among all modes)

Simulated Tempering



Algorithm is as follows:

1. Draw a starting sample θ^0 from a starting distribution $p_0(\theta)$

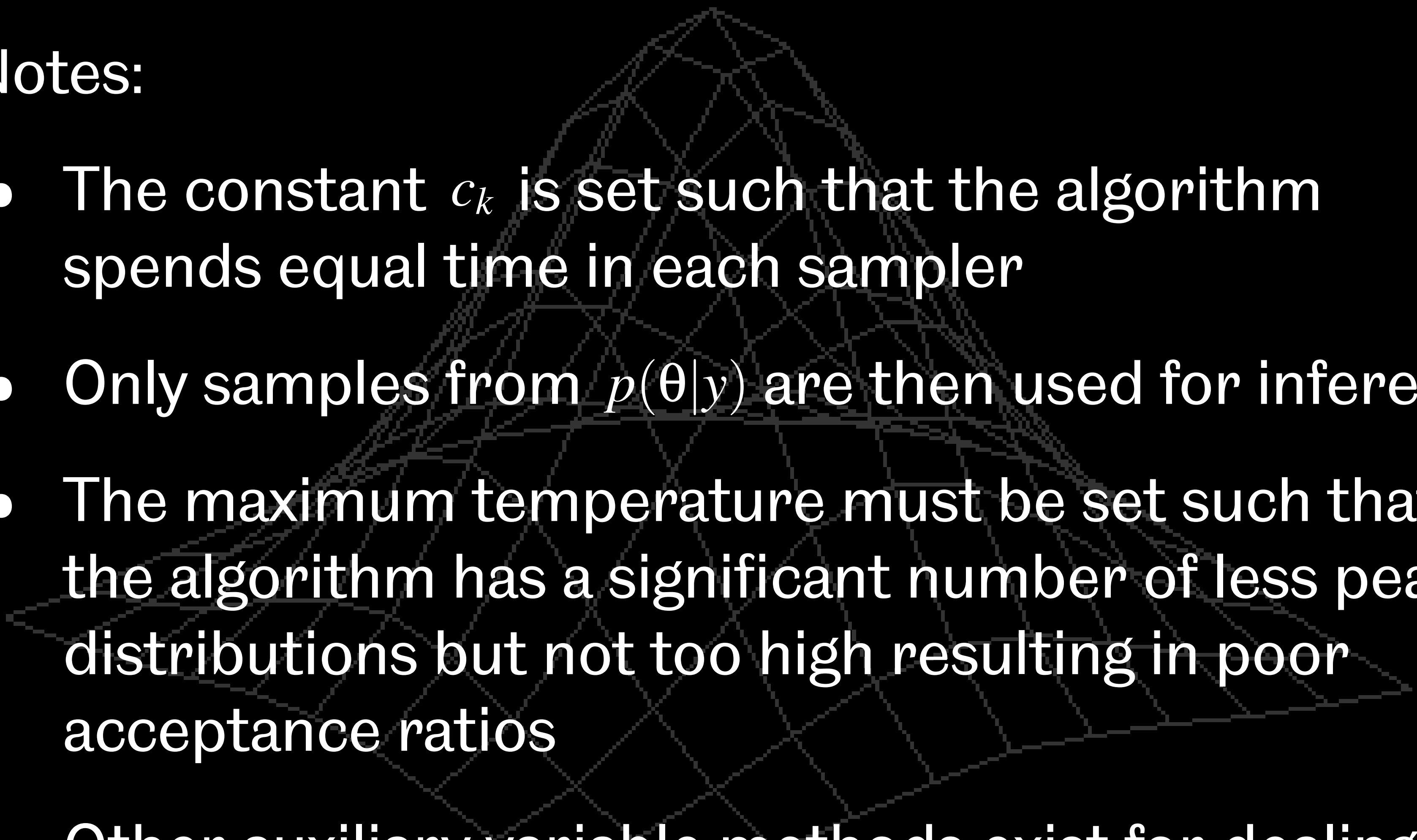
2. For each iteration t

   a. θ^{t+1} is sampled using the Markov chain with target distribution $p(\theta|y)^{1/T_k}$

   b. A jump from the current sampler k to sampler j is proposed with probability $J_{s,j}$. This move is then accepted with probability $min(1, r)$ where

$$r = \frac{c_j p(\theta^{t+1}|y)^{1/T_j} J_{j,k}}{c_k p(\theta^{t+1}|y)^{1/T_k} J_{k,j}}$$

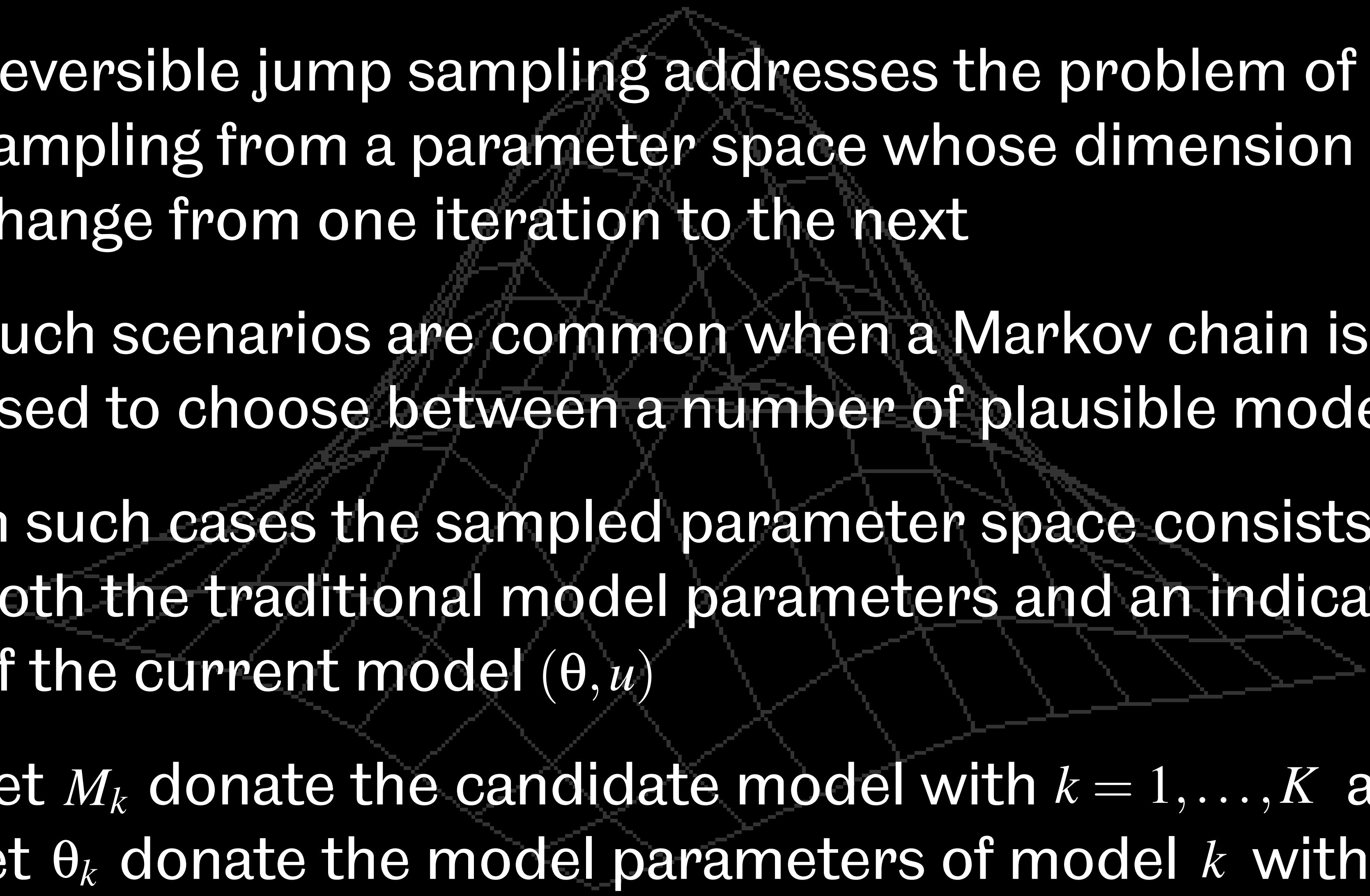
Simulated Tempering



Notes:

- The constant c_k is set such that the algorithm spends equal time in each sampler
- Only samples from $p(\theta|y)$ are then used for inference
- The maximum temperature must be set such that the algorithm has a significant number of less peaky distributions but not too high resulting in poor acceptance ratios
- Other auxiliary variable methods exist for dealing with multimodel distributions

Reversible Jump Sampling



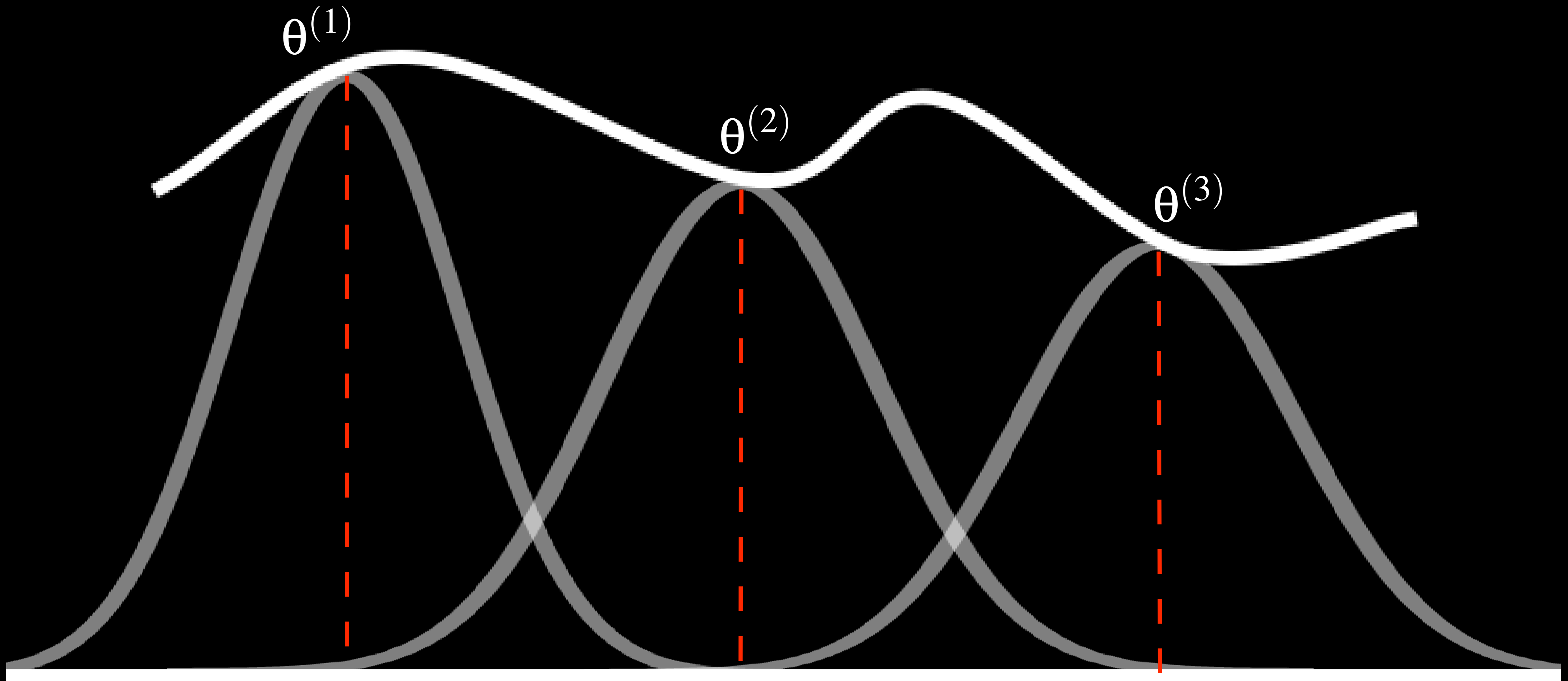
- Reversible jump sampling addresses the problem of sampling from a parameter space whose dimension can change from one iteration to the next

- Such scenarios are common when a Markov chain is used to choose between a number of plausible models

- In such cases the sampled parameter space consists of both the traditional model parameters and an indication of the current model (θ, u)

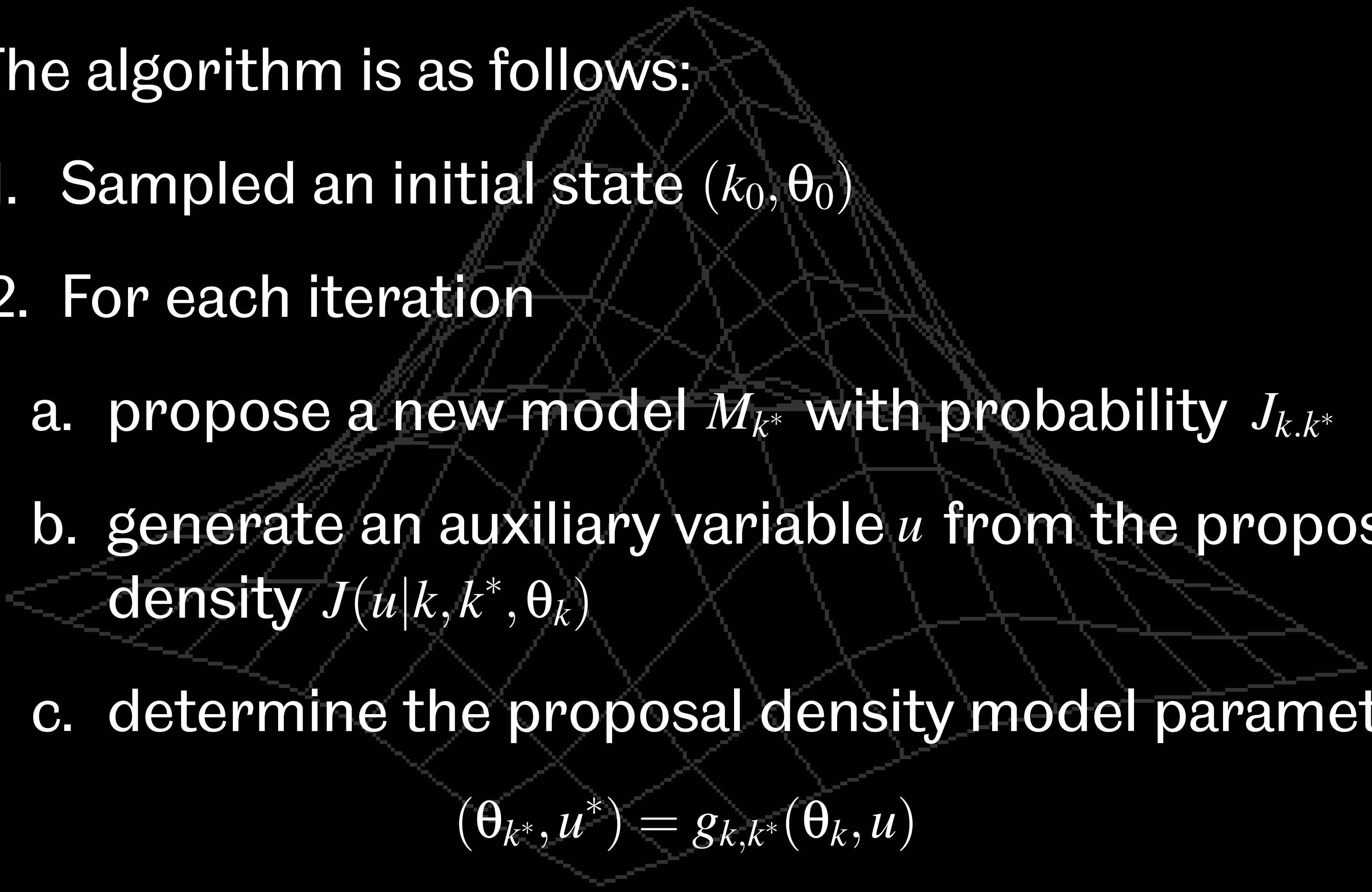
- Let M_k donate the candidate model with $k = 1, \ldots, K$ and let θ_k donate the model parameters of model k with dimension d_k

Reversible Jump Sampling



$k = 3$

Reversible Jump Sampling



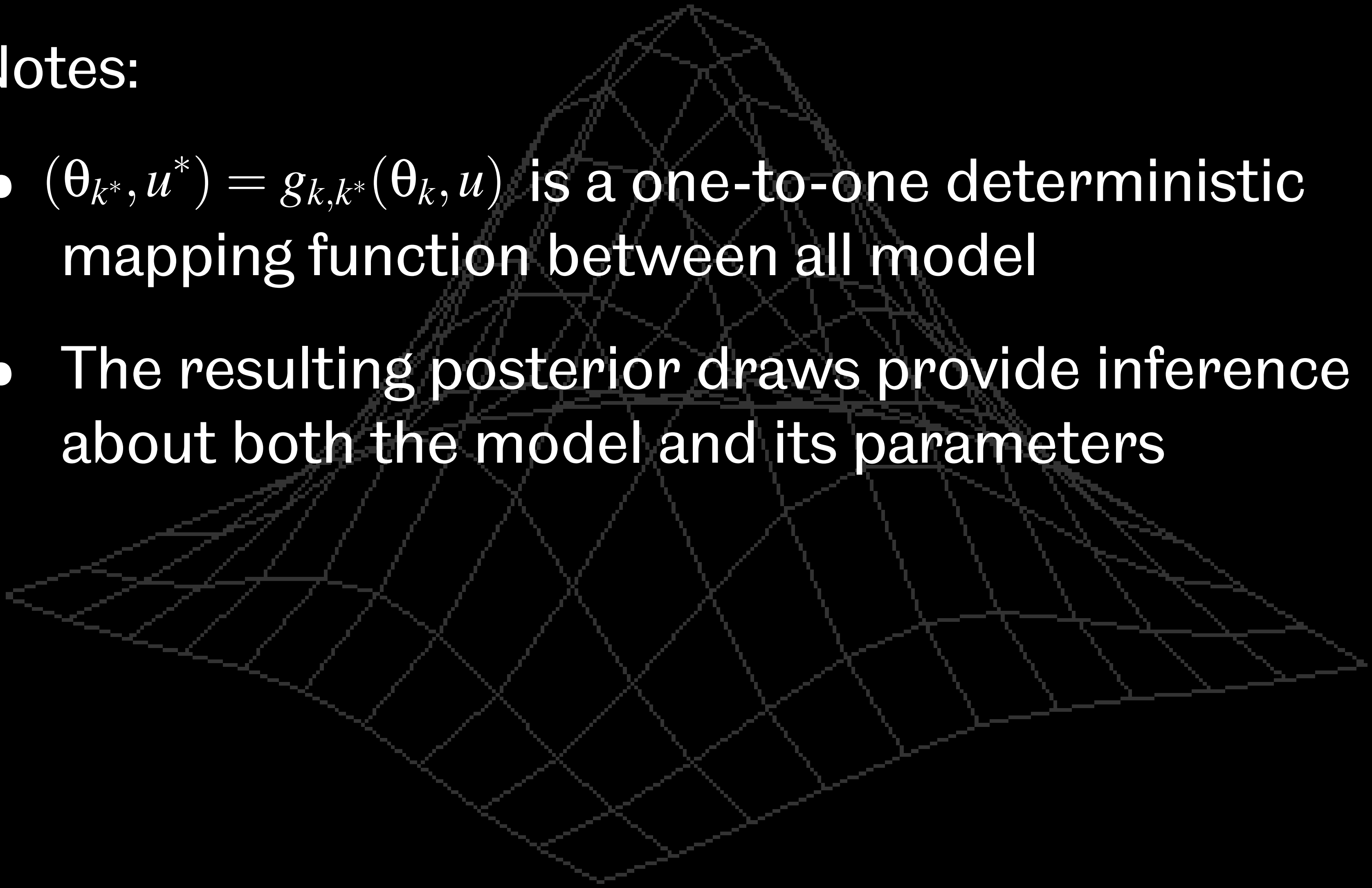
The algorithm is as follows:

1. Sampled an initial state (k_0, θ_0)
2. For each iteration
   a. propose a new model M_{k^*} with probability $J_{k.k^*}$
   b. generate an auxiliary variable u from the proposal density $J(u|k, k^*, \theta_k)$
   c. determine the proposal density model parameters
   $$(\theta_{k^*}, u^*) = g_{k,k^*}(\theta_k, u)$$
   d. accept the new model with probability $min(r, 1)$

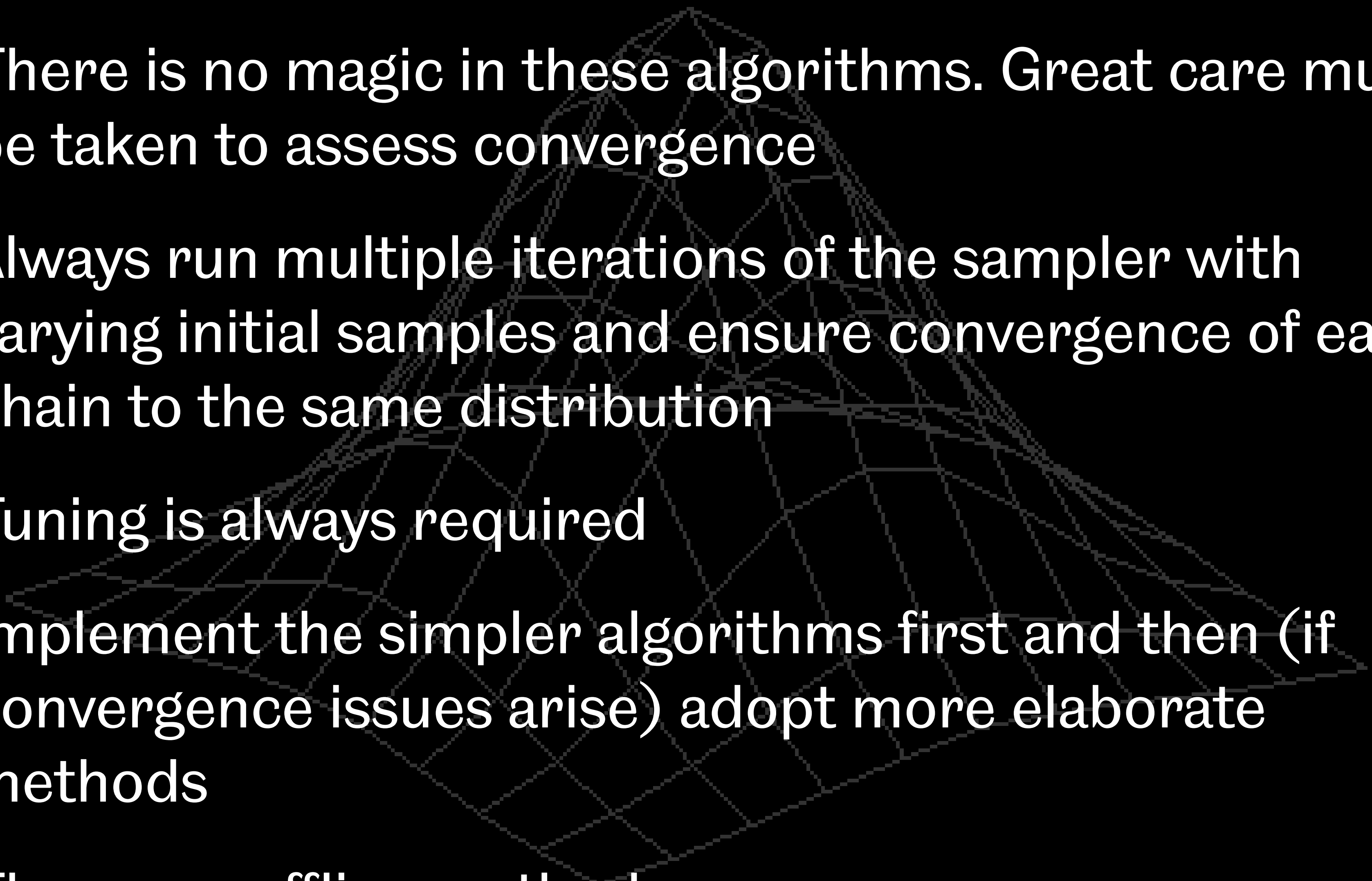
Reversible Jump Sampling

Notes:

- $(\theta_{k^*}, u^*) = g_{k,k^*}(\theta_k, u)$ is a one-to-one deterministic mapping function between all model
- The resulting posterior draws provide inference about both the model and its parameters



Conclusions



- There is no magic in these algorithms. Great care must be taken to assess convergence
- Always run multiple iterations of the sampler with varying initial samples and ensure convergence of each chain to the same distribution
- Tuning is always required
- Implement the simpler algorithms first and then (if convergence issues arise) adopt more elaborate methods
- These are offline methods

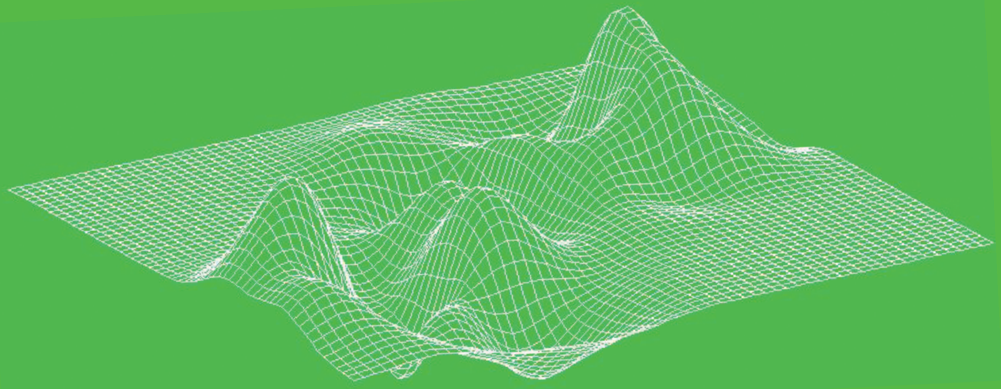
The
University
Of
Sheffield.

2007 SYMPOSIUM ON

DATA MODELLING

for new researchers

27th March 2007



Are you a data modeller? Do you use data to generate a mathematical description of a system? Do you use data driven models to investigate the learning process? Do you create models to mimic a system for tasks such as classification, filtering or prediction? If any of this describes your work then this symposium is for you!

The 2007 symposium on data modelling is a one day event aimed at postgraduate and postdoctoral researchers. It aims to bring together new researchers in data modelling from various departments in the University of Sheffield. Any new researcher on models, methods and applications of data driven modelling is welcomed.

We are looking for one page abstracts of recent work or work in progress for peer review for both oral and poster presentations. The deadline for submission is 1 February 2007. Registration will be FREE for presenters and will be open in the new year.

datamodelling.group.shef.ac.uk/symposium