Next-generation MCMC: theory, options, and practice for Bayesian inference in ADMB

Cole Monnahan, James Thorson, Ian Taylor

2/27/2013

Outline

Goal:

Detail options for MCMC in ADMB, and when and how they should be used.

Outline:

1. Brief introduction to theory of MCMC
2. Metropolis algorithm in ADMB, as well as built-in options
3. How to specify an arbitrary correlation matrix to ADMB, and an example of why this can be useful
4. Theory and example of MCMCMC
5. Theory and example of –hybrid MCMC

Markov Chains

A Markov chain (MC) is a stochastic process with the property that:

$$\Pr(X_n \mid X_1, \ldots, X_{n-1}) = \Pr(X_n \mid X_{n-1})$$

(i.e. to know where it's going, you only need to know where it is)

If this chain satisfies the conditions:

1. Can get from any state to any other (irreducible)
2. Mean return time to a state is finite (positive recurrent)

Then chain converges to a distribution as $n \to \infty$

Note: We generally don't need to worry about these conditions in practice.

Markov chain Monte Carlo

- A MCMC is a chain designed such that the **equilibrium distribution = posterior of interest**
- How to create such a chain?

Let: $c \cdot f()$= posterior density $X_{proposed}$ = proposed parameters

$X_{current}$ = current parameters $U \sim$ random uniform (0,1)

Then:

$$X_{new} = \begin{cases} X_{proposed} & \text{if } U \leq \dfrac{c \cdot f(X_{proposed})}{c \cdot f(X_{current})} \\ X_{current} & \text{otherwise} \end{cases}$$

← This is the Metropolis algorithm

Metropolis MCMC

A few comments on Metropolis MCMC:

- If the density of the proposed state is higher than the current state, the chain moves there (e.g. Pr(U<4/1)=1). If it is lower, if moves there with a probability proportional to the ratio (e.g. Pr(U<1/4)=1/4)
- The normalizing constant c cancels out in the ratio, and thus doesn't need to be known. Makes MCMC useful for Bayesian inference!
- The "**proposal function"** (or jump function) generates proposed states, given the current one. The proposal function is symmetric for the Metropolis. If it isn't, that is the Metropolis-Hastings algorithm.
- The rate of convergence depends on the proposal function. **The point of this Think Tank is to discuss "better" functions.**

Metropolis-Hastings

$$U \leq \frac{f(X_p)q(X_p \mid X_c)}{f(X_c)q(X_c \mid X_p)}$$

If q symmetric

Metropolis

$$U \leq \frac{f(X_{proposed})}{f(X_{current})}$$

MCMC in ADMB

- ADMB uses a MNV(0,Σ) proposal function (symmetric), where Σ is the covariance matrix calculated by inverting the Hessian.
- If the posterior is multivariate normal, then this Metropolis algorithm works very well.
- But fishery and ecological models exhibit:
  - Correlation and non-linear posteriors
  - Parameters with high support at boundary
  - Multi-modality, etc.
- To improve convergence we can:
  - Re-parameterize the model to make more normal
  - Change the covariance matrix (e.g. mcrb), proposal function (~~mcgrope~~ mcprobe), or acceptance rate
  - Abandon Metropolis and adopt a next-generation algorithm: MCMCMC or the hybrid method

-mcrb N algorithm

(1) $\boldsymbol{\Sigma}_{\text{old}} = \begin{bmatrix} 1 & \cdots & \rho_{1,n} \\ \vdots & \ddots & \vdots \\ \rho_{n,1} & \cdots & 1 \end{bmatrix}$ The original correlation matrix

(2) $\mathbf{L} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ L_{n,1} & \cdots & L_{n,n} \end{bmatrix}$ Lower Choleski decomposition of $\boldsymbol{\Sigma}_{\text{old}}$

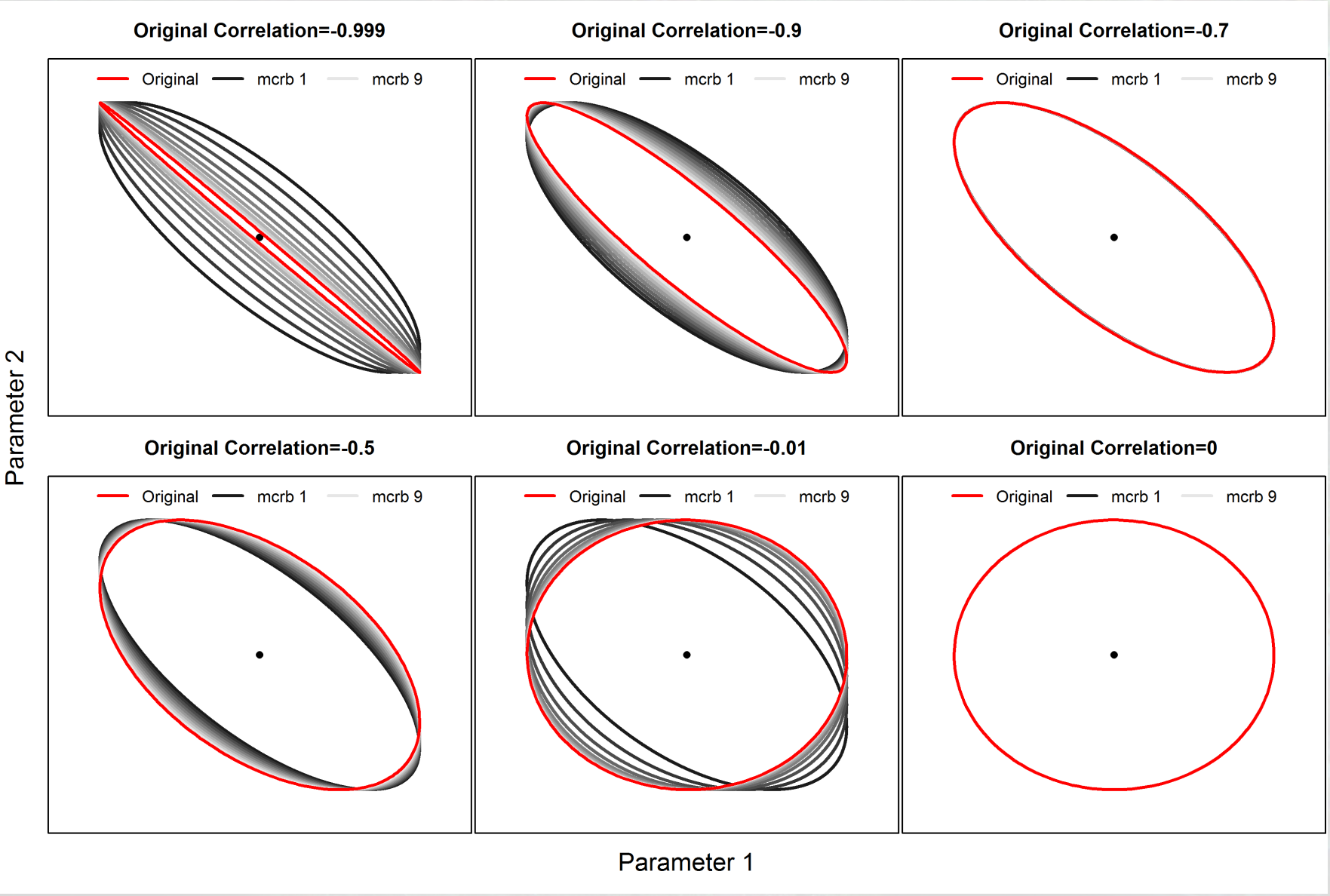
(3) $\hat{\mathbf{L}} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ L_{n,1}^{N/10} & \cdots & L_{n,n}^{N/10} \end{bmatrix}$ Raise elements to power user supplied N

(4) $\tilde{\mathbf{L}} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \frac{\hat{L}_{n,1}}{\left|\hat{L}_{n,\cdot}\right|} & \cdots & \frac{\hat{L}_{n,n}}{\left|\hat{L}_{n,\cdot}\right|} \end{bmatrix}$ Normalize rows of $\hat{L}$

(5) $\boldsymbol{\Sigma}_{\text{new}} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$ Calculate new correlation matrix

See `read_hessian_matrix_and_scale1()` for source code and note that the “corrtest” file contains these steps

-mcrb N examples



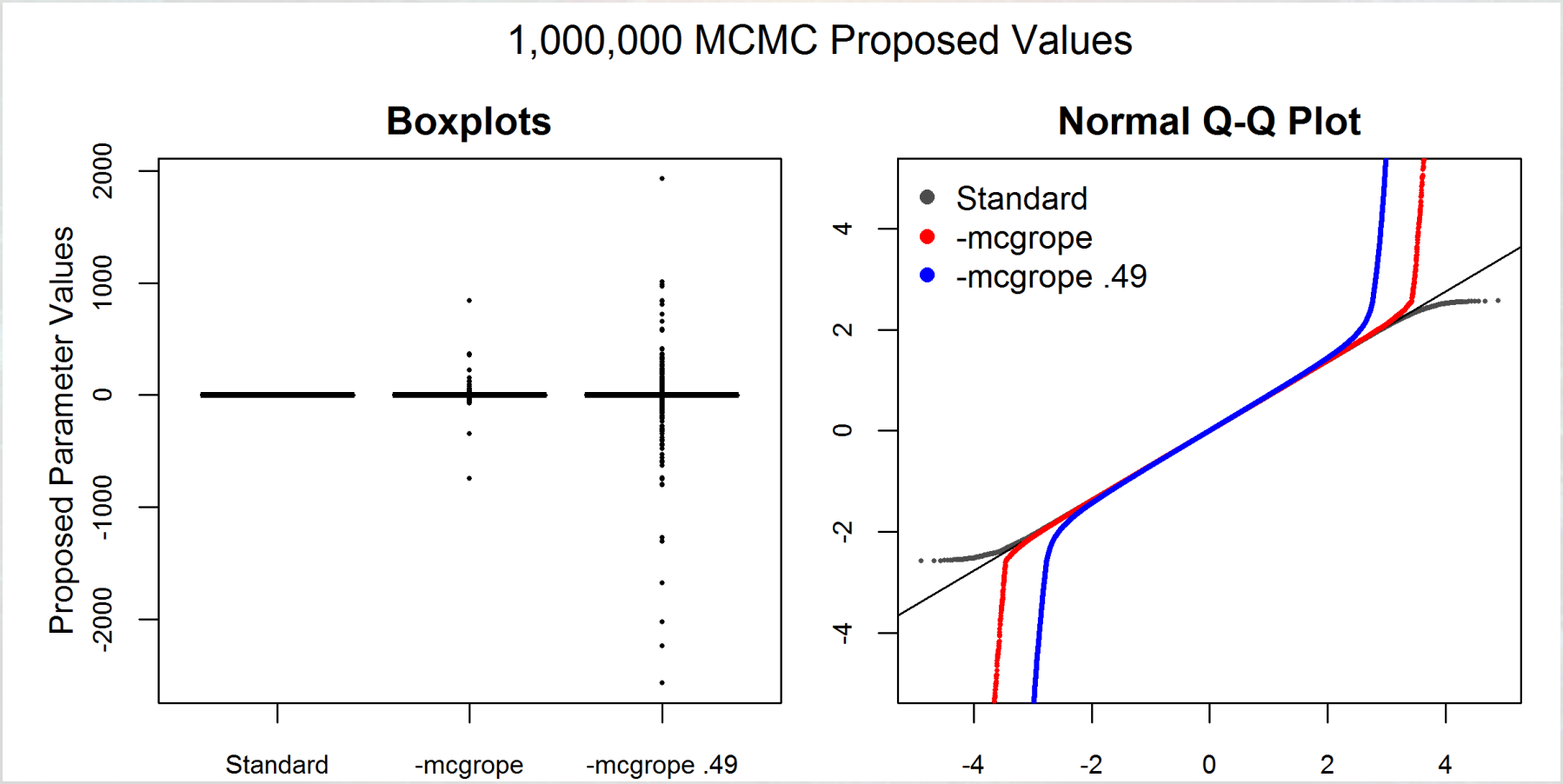
-mcprobe algorithm

- This option is designed to occasionally jump much further than typical, with the hope it will escape a local minimum. (Recently renamed from mcgrope).
- Note: N is optional value specified by user, N=.05 by default, but accepted range is .00001<N<.499. The higher the number, the more often it "probes" for another area of high density.
    1. Generate MVN sample. Calculate the cumulative density=D.
    2. If D>N, keep MVN draw. If D<N replace with draw from multivariate Cauchy distribution.
- I think? The code is more complicated than mcrb and I haven't recoded it in R as a check. Anyone else know?
- We can cheat by looking at the proposed values

See:
`new_probing_bounded_multivariate_normal()`
for details.

-mcprobe examples

- Took simple.tpl, fixed *a* at MLE, ran MCMC for *b*
- Collected proposals → infer proposal function



Specifying a correlation matrix

- The built-in ADMB options are quick and easy to try, but not very flexible.
- A more flexible approach is to force ADMB to use an arbitrary correlation matrix:
  - When running –mcmc, ADMB reads in admodel.cov and uses it in algorithm. So change this file and ADMB will use what we tell it.
  - This approach allows the user to change the correlation but also the acceptance rate by scaling the variances.
  - Unfortunately the admodel.cov file is in unbounded space (i.e. before the bounded transformation is done) and needs to be converted.
  - http://www.admb-project.org/examples/admb-tricks/covariance-calculations shows how to fix this

Specifying a correlation matrix

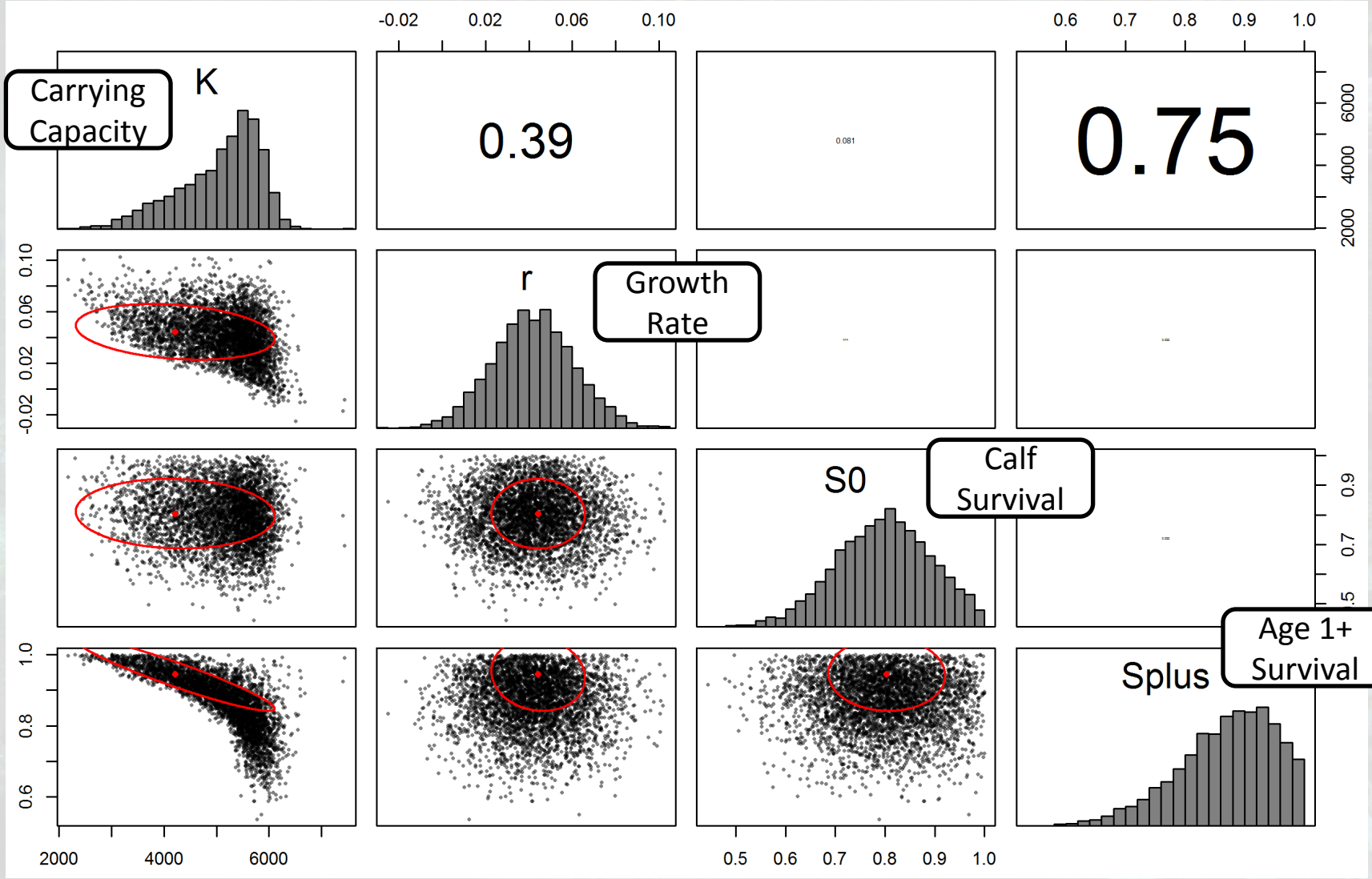
The admodel.cov file:

Content	Description	Type	Size
n	Number of parameters (not including `sd_variables`)	Integer	1
cov	The Covariance matrix, as vector of elements	Numeric	n^2
hbf	The `hybrid_bounded_flag`, dictating bounding function	Integer	1
scale	The "scale" used in the Delta method	Numeric	n

Example code for reading into R:

```
filename <- file("admodel.cov", "rb")
num.pars <- readBin(filename, "integer", 1)
cov.vec <- readBin(filename, "numeric", num.pars^2)
cov <- matrix(cov.vec, ncol=num.pars, nrow=num.pars)
hybrid_bounded_flag <- readBin(filename, "integer", 1)
scale <- readBin(filename, "numeric", num.pars)
cov.bounded <- cov *(scale %o% scale)  # the bounded Cov
se <- sqrt(diag(cov.bounded))
cor <- cov.bounded/(se %o% se)         # bounded Cor
```

Simple age-structured model



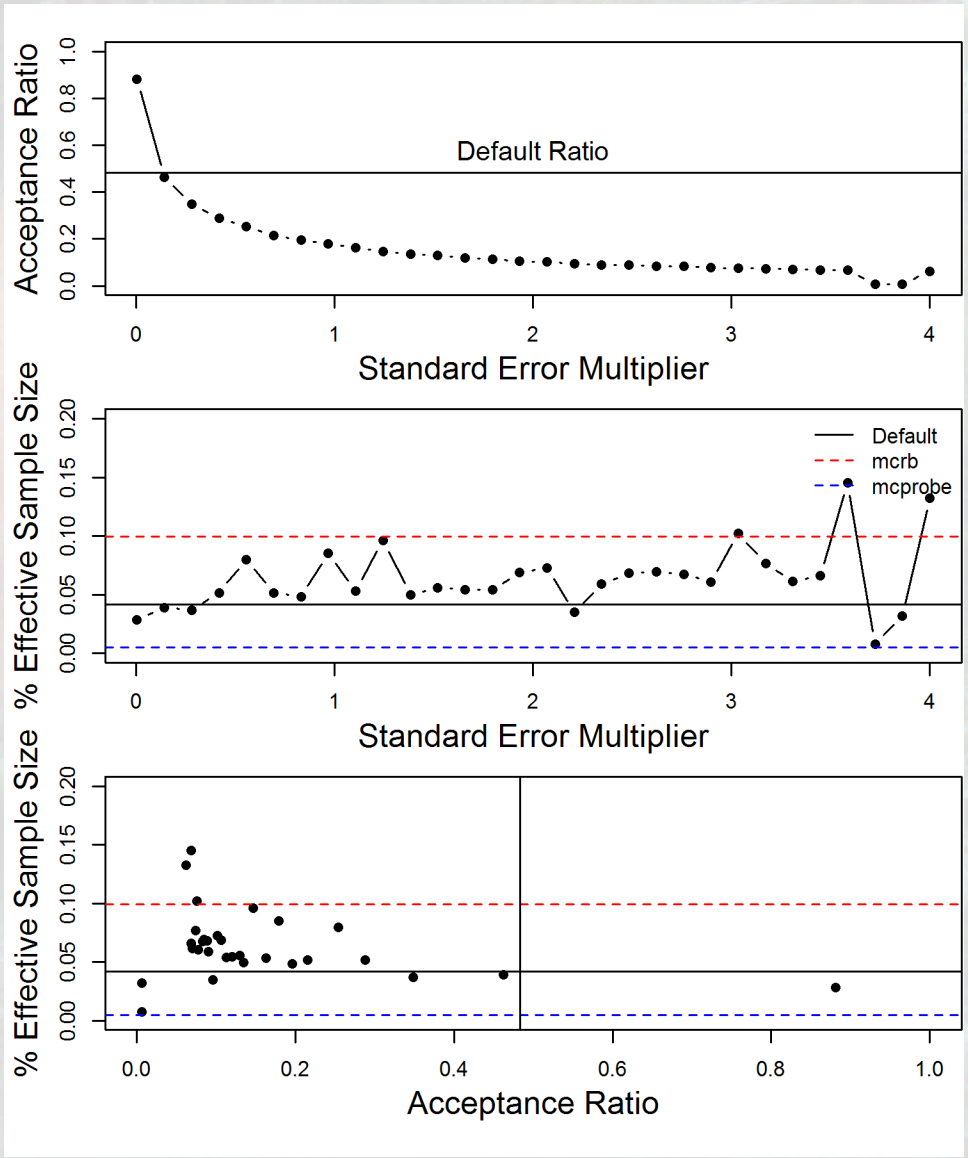
Optimizing acceptance ratio (AR)

Note: ADMB scales so that .15<AR<.4 during first 2500 iterations, unless –mcnoscale

To optimize AR:

1. Change SEs, write to admodel.cov, turn off scaling
2. Repeat for different SEs
3. Look for faster convergence
4. Be wary of %EFS for chains that haven't fully explored the parameter-space!

For this model, no major improvement by changing AR



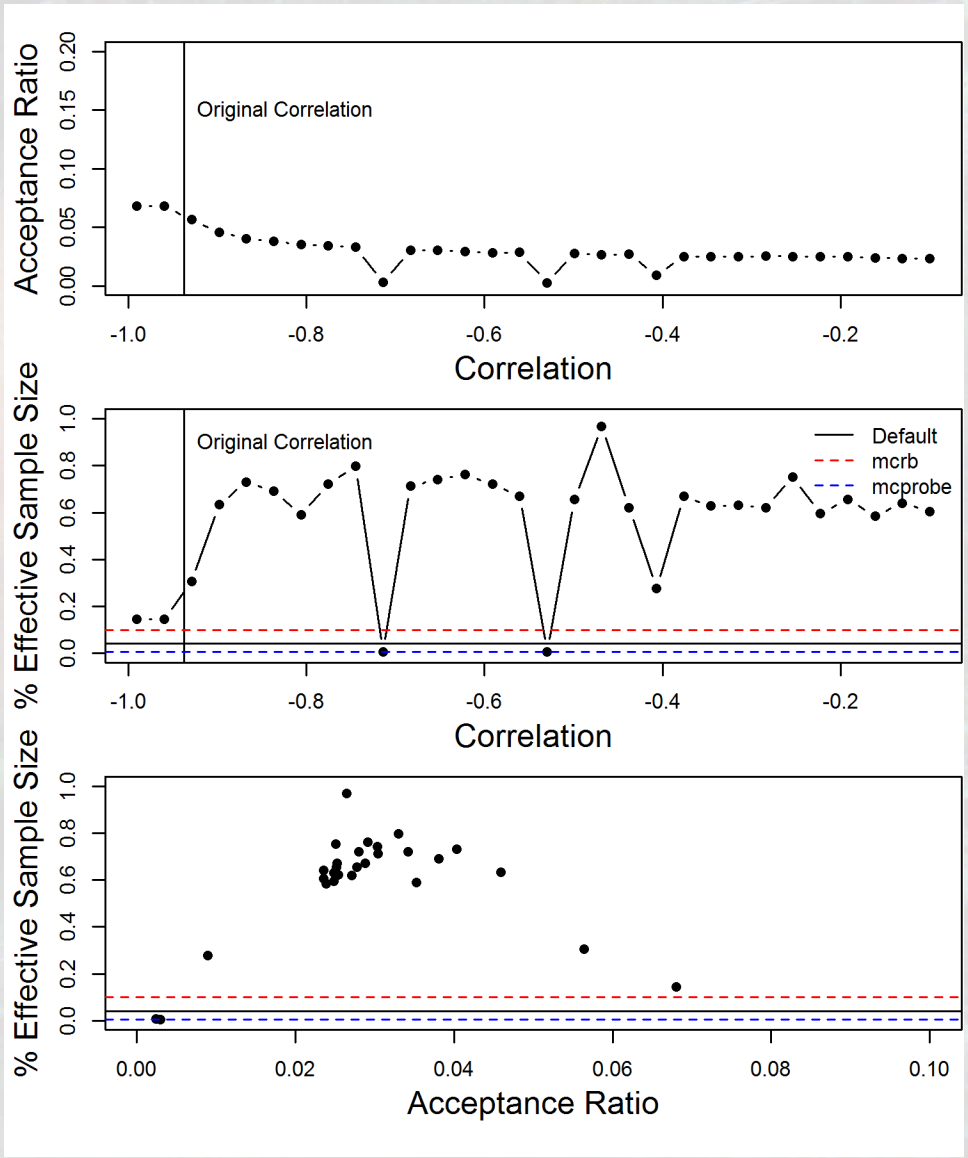
Optimizing the correlation matrix

We suspect suboptimal correlation for Age 1+ survival parameter.

Steps to Optimize Cor:

1. Examine preliminary posterior pairs()
2. Write admodel.cov w/ desired matrix
3. Turn off scaling (?)
4. Repeat for different matrices
5. Look for faster convergence

By optimizing the AR and correlation of one parameter, the **model converges >20 times faster than default**!



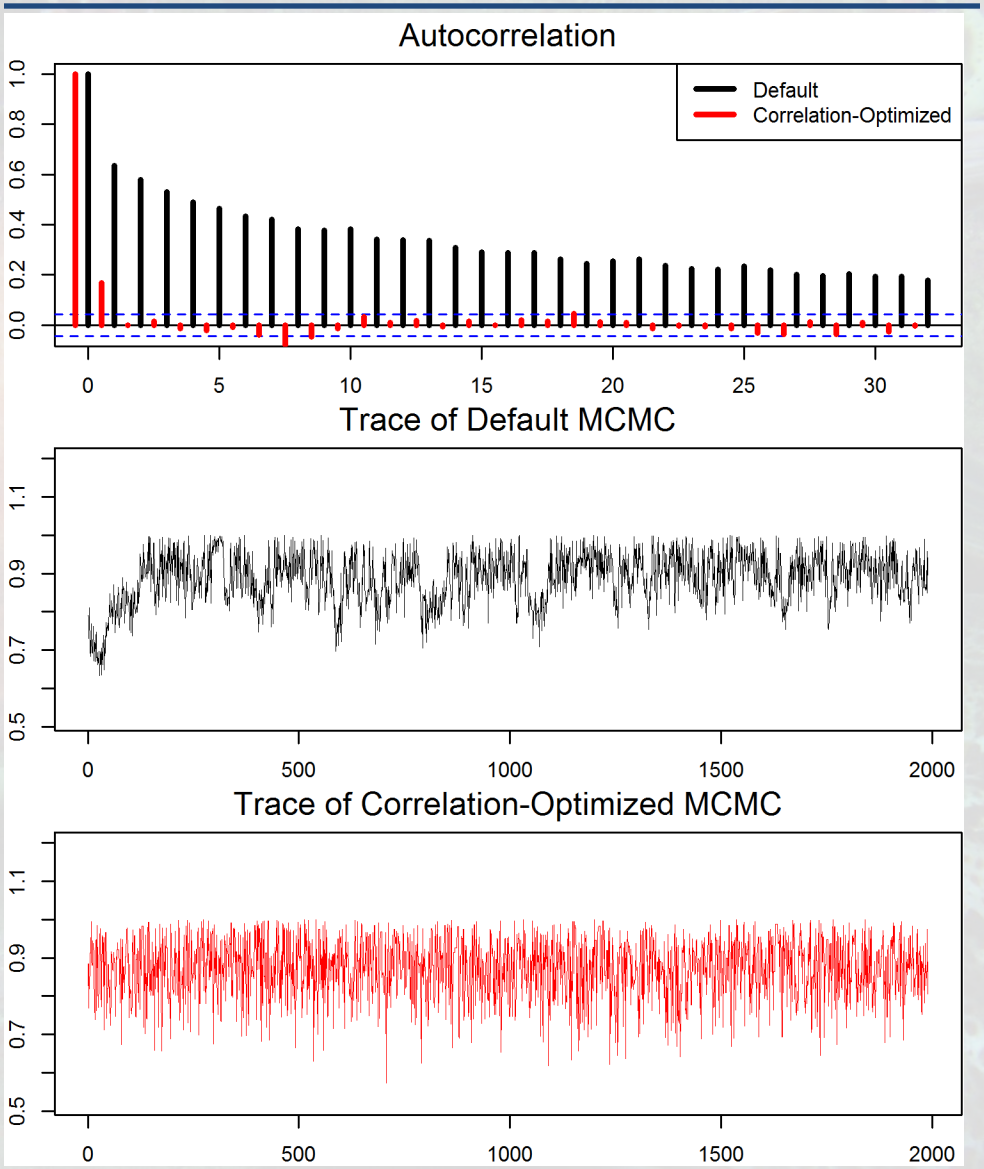
Optimizing the correlation matrix

We suspect suboptimal correlation for Age 1+ survival parameter.

Steps to Optimize Cor:

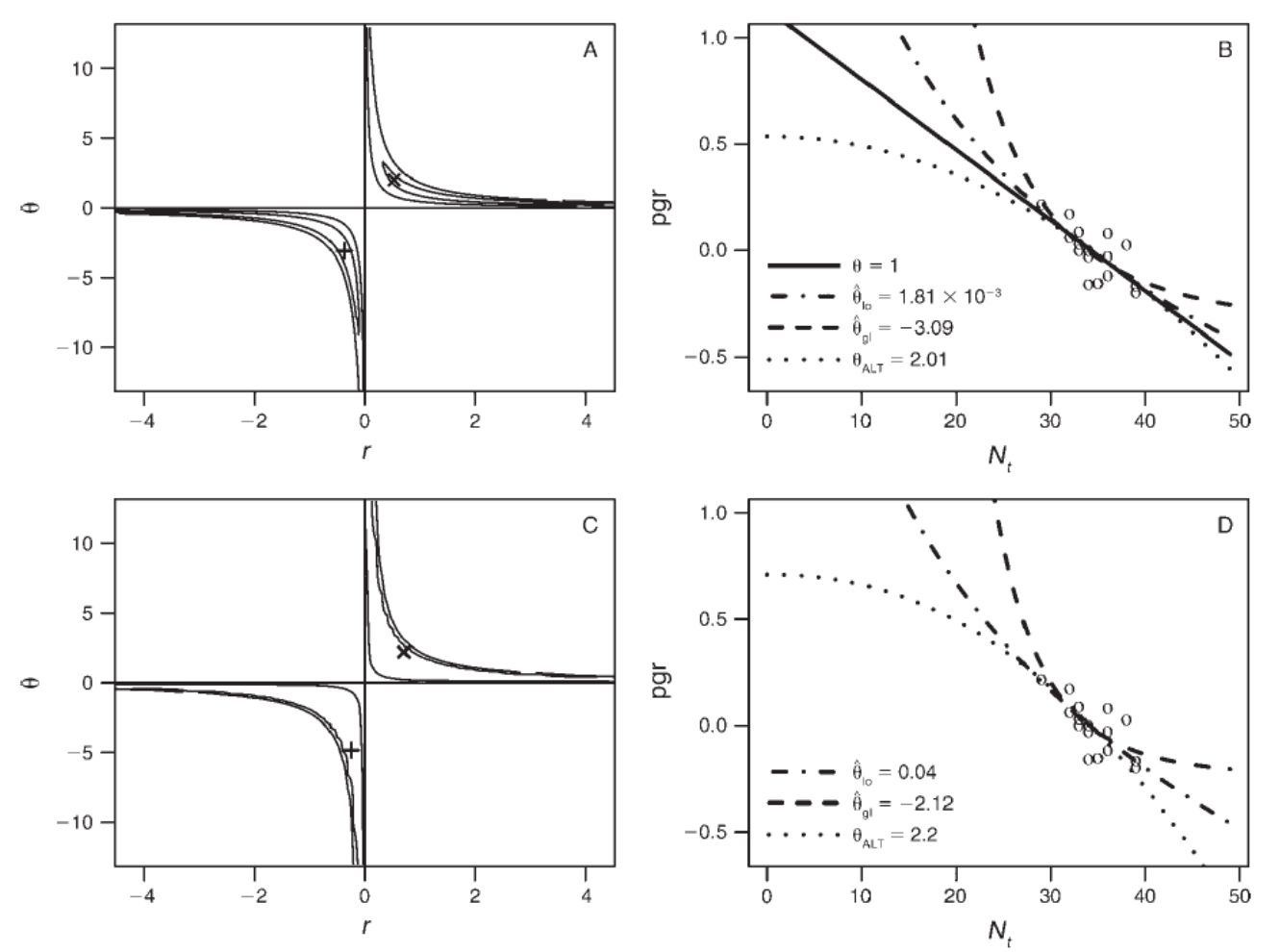
1. Examine preliminary posterior pairs()
2. Write admodel.cov w/ desired matrix
3. Turn off scaling (?)
4. Repeat for different matrices
5. Look for faster convergence

By optimizing the AR and correlation of one parameter, the **model converges >20 times faster than default**!



Metropolis-Coupled MCMC

Goal: improved sampling of multimodal surfaces



Polansy et al. 2009. Ecology 90(8): 2313-2320.

Metropolis-Coupled MCMC

Fishery examples

- Ecosystem models
- Mixture distribution models
- Genetics studies (“Mr. Bayes”)

MLE solutions

- Multiple starting points
- “Heuristic” optimizers (e.g., particle-swarm)
  - Tashkova et al. 2012. Ecol. Model. 226:36-61.

Metropolis-Coupled MCMC

Algorithm:

1. Run standard + 'heated' chains (e.g., 1000 samples)
   - Heated chain = chain using $\pi(\theta)/N$, where N = {2,4,6}
2. Propose swapping chains
   - Probability of swapping normal chain $\theta^{(i)}$ and heated chain $\theta^{(j)}$ = $$\frac{\pi(\theta^{(j)}) \cdot \pi(\theta^{(i)})/N}{\pi(\theta^{(i)}) \cdot \pi(\theta^{(j)})/N}$$
3. Repeat (e.g., 1000 times)

Metropolis-Coupled MCMC

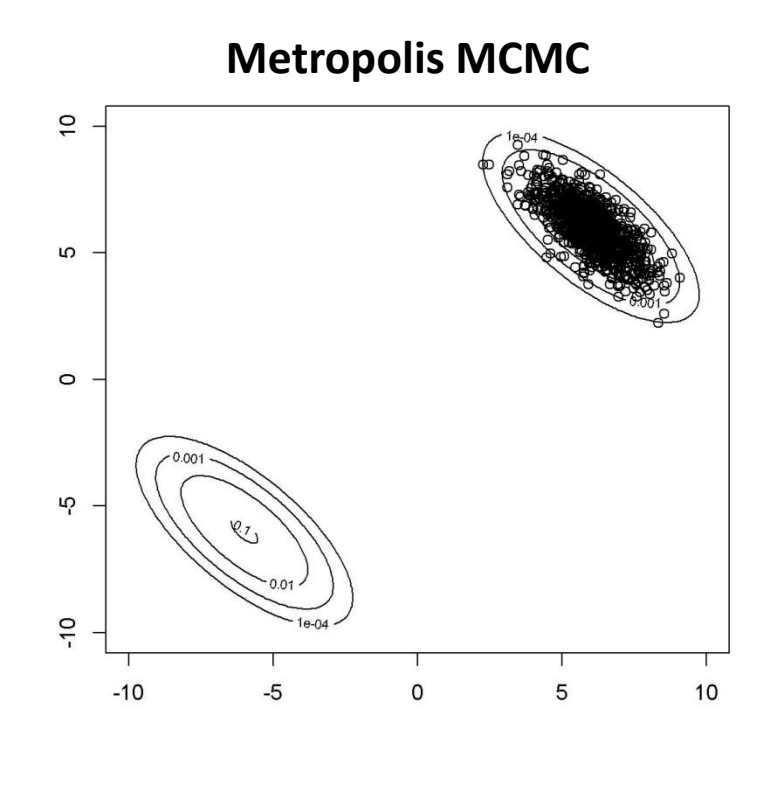
Theory

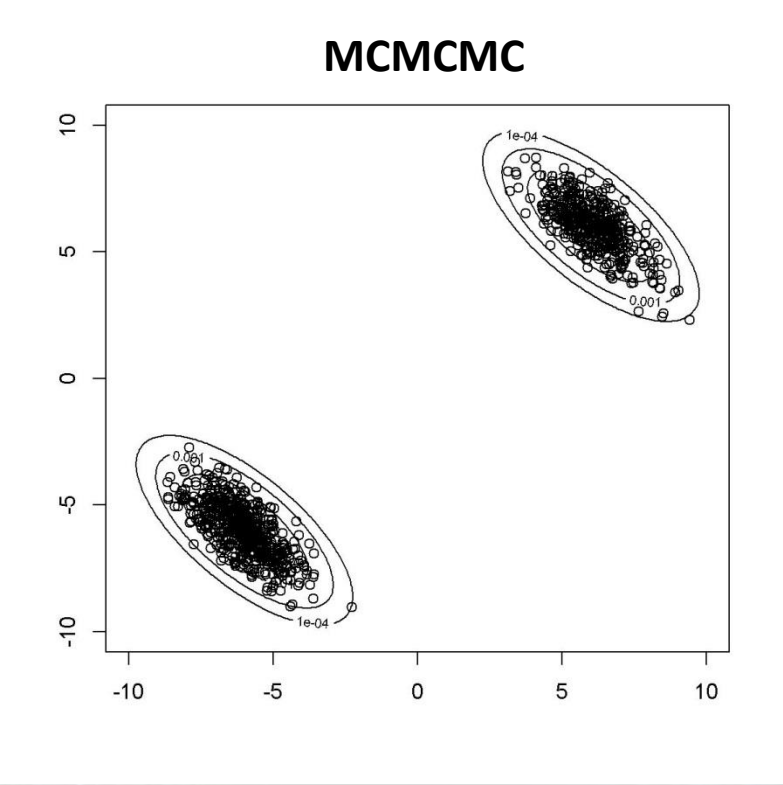
- Accept-rejection for swap is a Metropolis-Hastings step
  - Proposal ratio $\dfrac{\pi(\theta^{(i)})/N}{\pi(\theta^{(j)})/N}$
  - Likelihood ratio $\dfrac{\pi(\theta^{(j)})}{\pi(\theta^{(i)})}$
- Heated chain serves as adaptive sample for proposal density

Metropolis-Coupled MCMC

Software application

- JTT built a generic tool using ADMB
  - http://www.admb-project.org/examples/r-stuff/mcmcmc





Hamiltonian MCMC

Goal: improved convergence with irregular posteriors

- Metropolis works poorly with non-stationary covariance
  - E.g., most age-structured models
- Gibbs works poorly with high covariance
  - E.g., most hierarchical models

Hamiltonian MCMC

Sample from *H*

$$H = -\ln(\pi(\theta)) + \sum_i \frac{p_i^2}{2m_i}$$

-ln(π(θ)) = potential energy

p_i = kinetic energy

m_i = mass for parameter *i*

Steps:

1. Draw from ***p***
2. Trajectory maintains fixed H
3. Marginalize across ***p*** because ***p*** and **θ** and independent

Outcome – Draws from π(θ) (see Beskos et al. 2010 for proof; requires knownledge of statistical mechanics)

Hamiltonian MCMC

Algorithm (https://sites.google.com/site/thorsonresearch/code/hybrid)

1. $\phi(x) = -\ln(\pi(x))$; choose *m*, *t*, *n*
2. Randomly draw p(t) from Gaussian with mean=0, sd=m
3. Leapfrog projection
    1. $p(t+\tau/2)=p(t) - \tau/2\ \phi'(x)$
    2. $x(t+\tau) = x(t)+\tau(p(t+\tau/2) /m)$
    3. $p(t+\tau) = p(t+\tau/2) - \tau/2\ \phi'(x+\tau)$
4. Repeat the Step-3 *n* times
5. Accept-reject based on $\phi(x_0)/ \phi(x_1)$
6. Repeat Step 2-5, e.g., 1000 times

Please see example on personal website

Hamiltonian efficiency

Efficiency = number of IID draws from PDF / number of MCMC draws to achieve same variance

Assume multivariate normal with *d* independent dimensions:

Gibbs: efficiency = 1/d (Gelman et al. 2004, pg. 306)

Metropolis: efficiency = 0.3/d (ibid)

Hamiltonian: efficiency ~ $0.3/\tau$ (Hanson 2001)

where τ is number of steps

and $\tau_{opt} \sim d^{(1/4)}$ (Beskos et al. 2010)

Recommendations

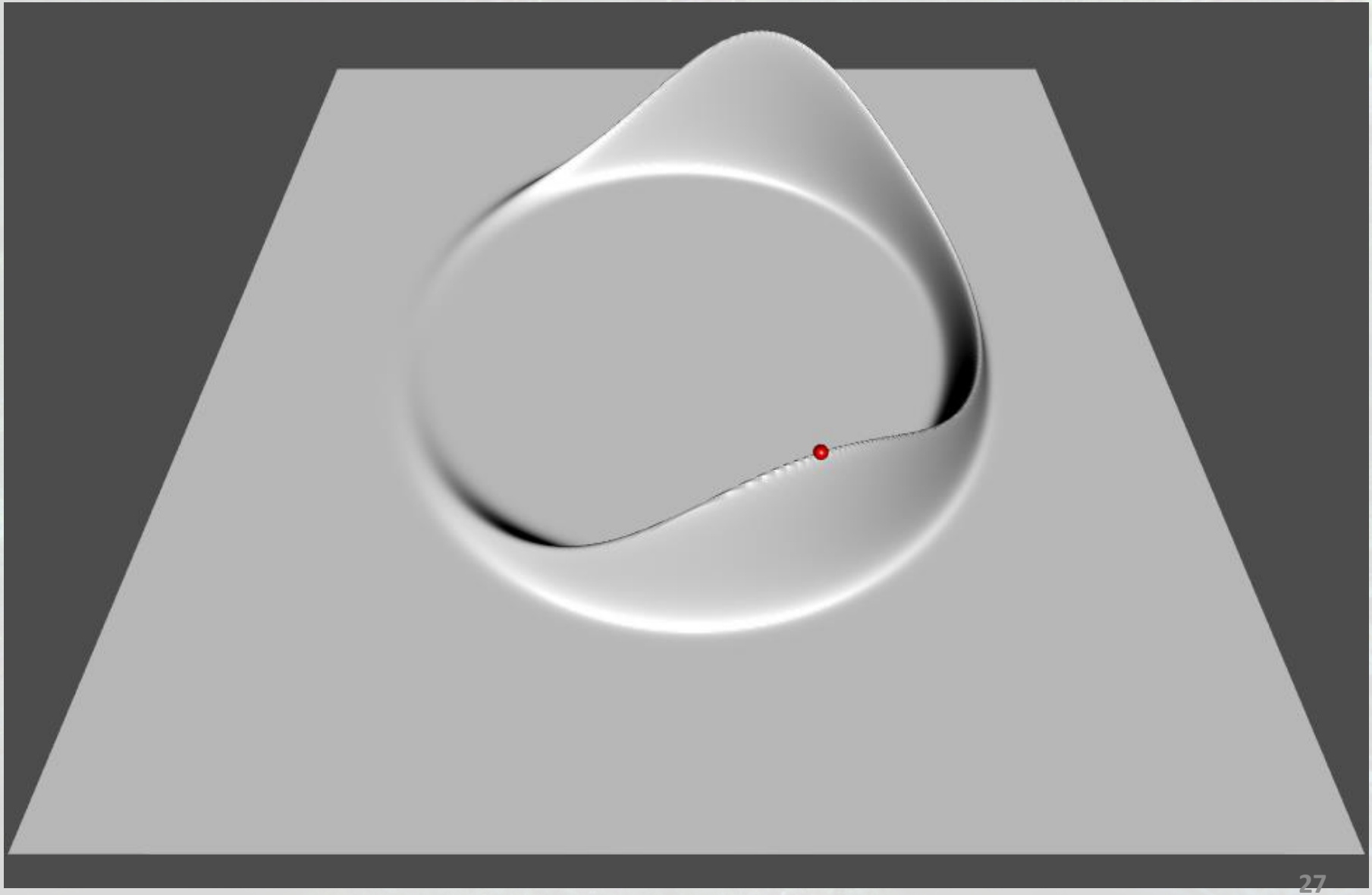
Tuning:

1. Fix τ = ceiling($d^{(1/4)}$) (-hynstep)
2. Fix ***M*** at estimated covariance (Hofman and Gelman 2011)
3. Tune *l* to achieve acceptance ~ 0.651 (-hyeps) (Beskos et al. 2010)

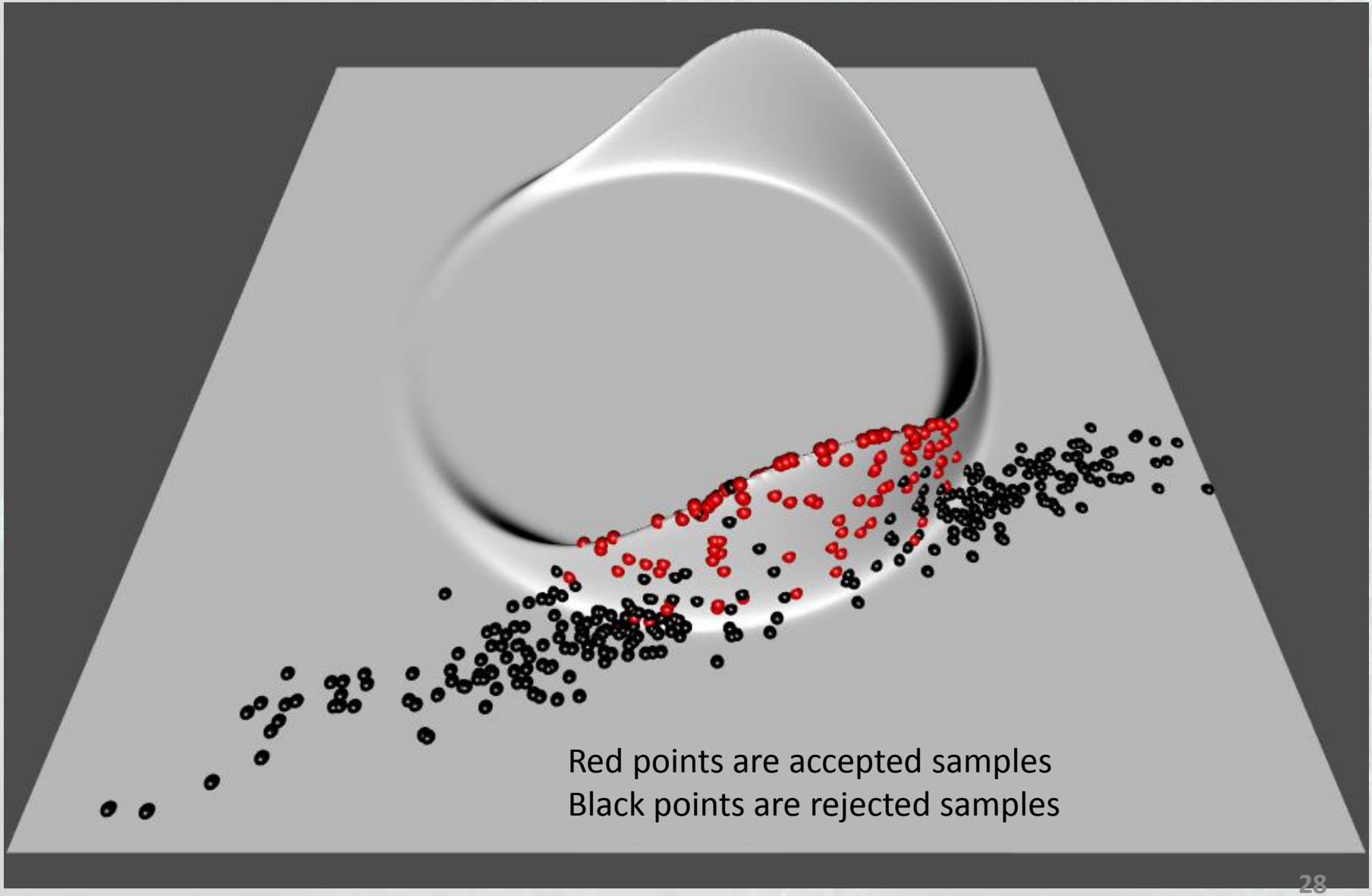
Notes:

If low dimension, τ=1 and Hamiltonian becomes Langevin sampling

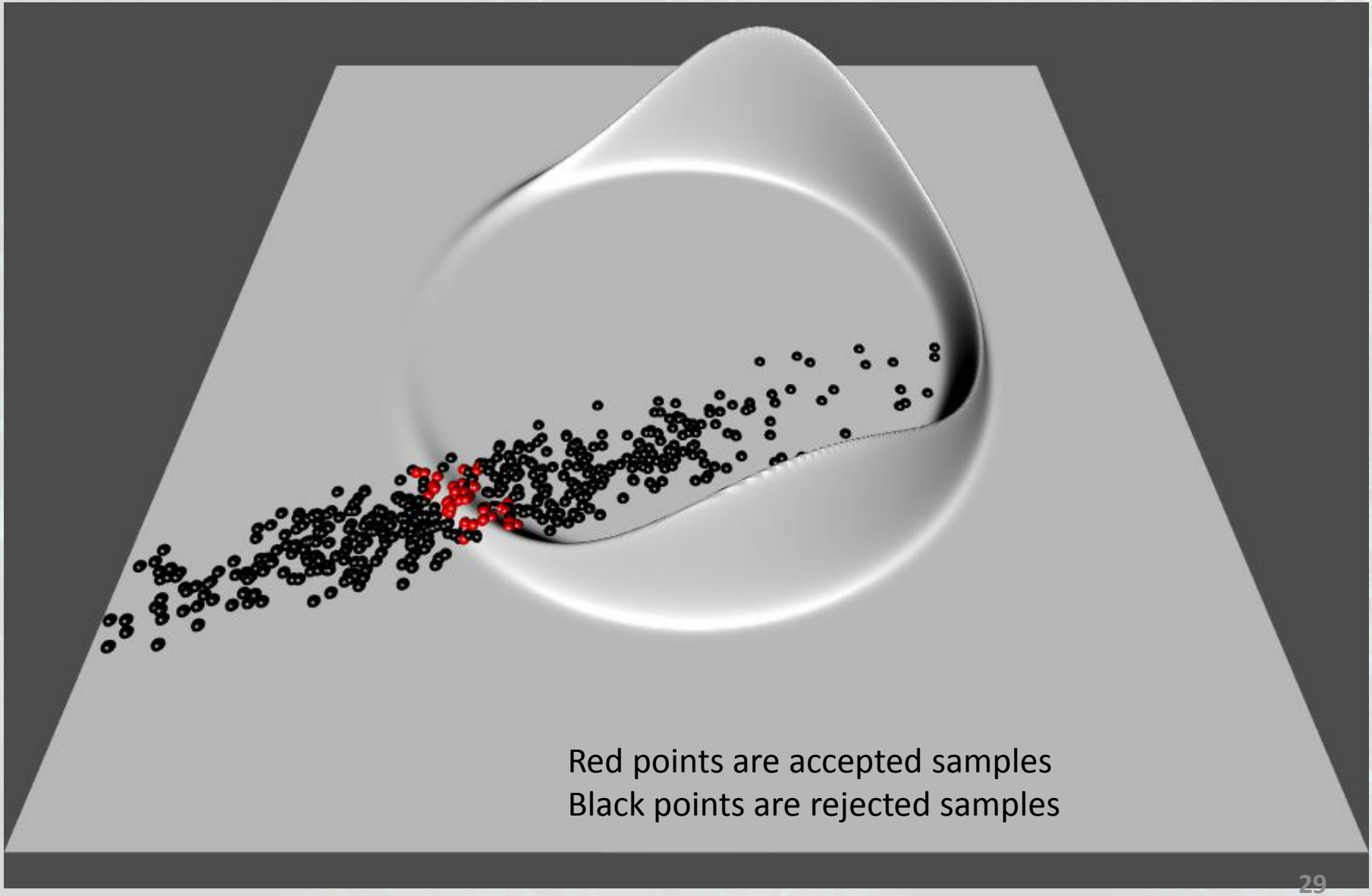
ADMB's standard MCMC



ADMB's standard MCMC

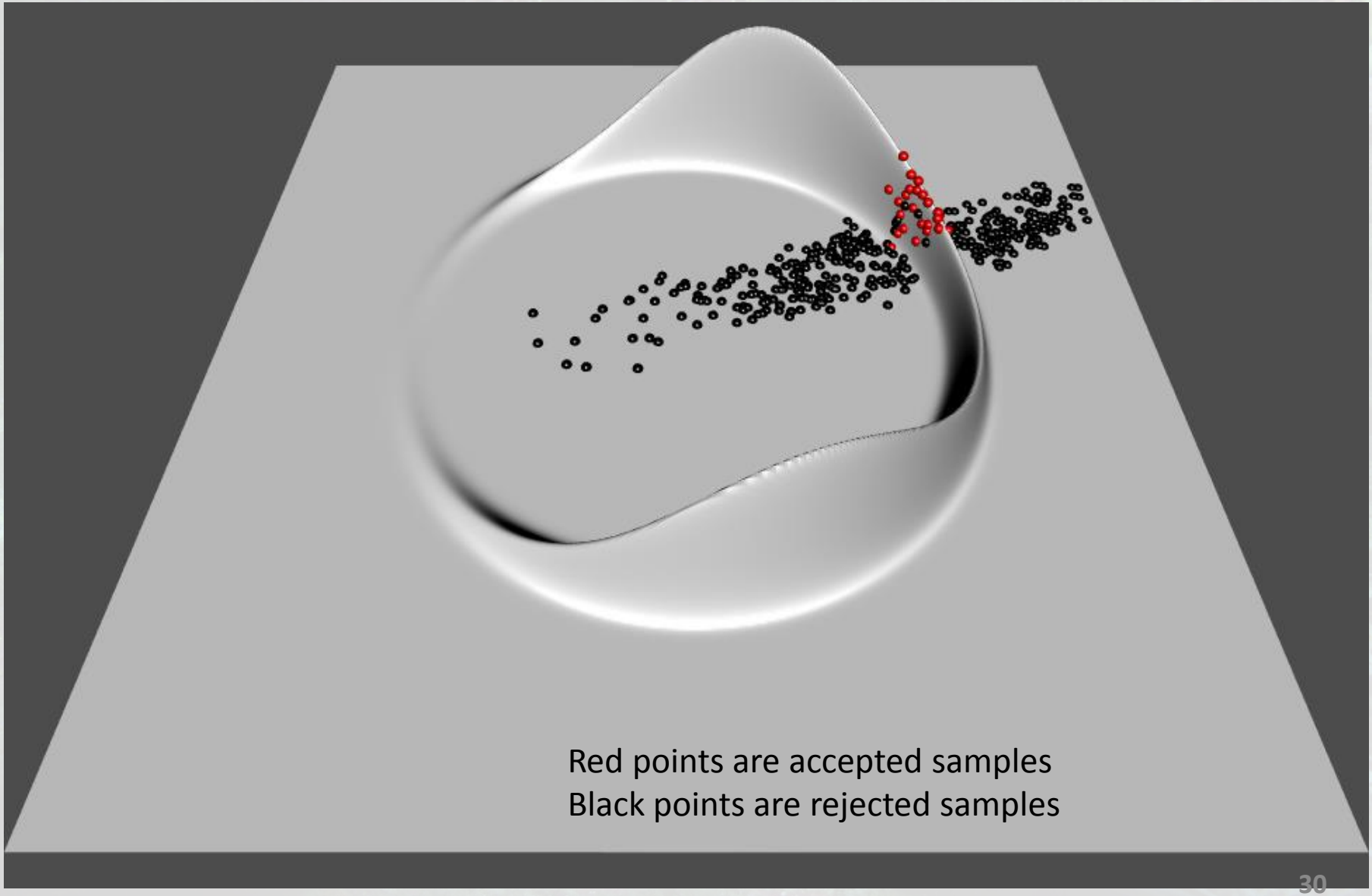


ADMB's standard MCMC



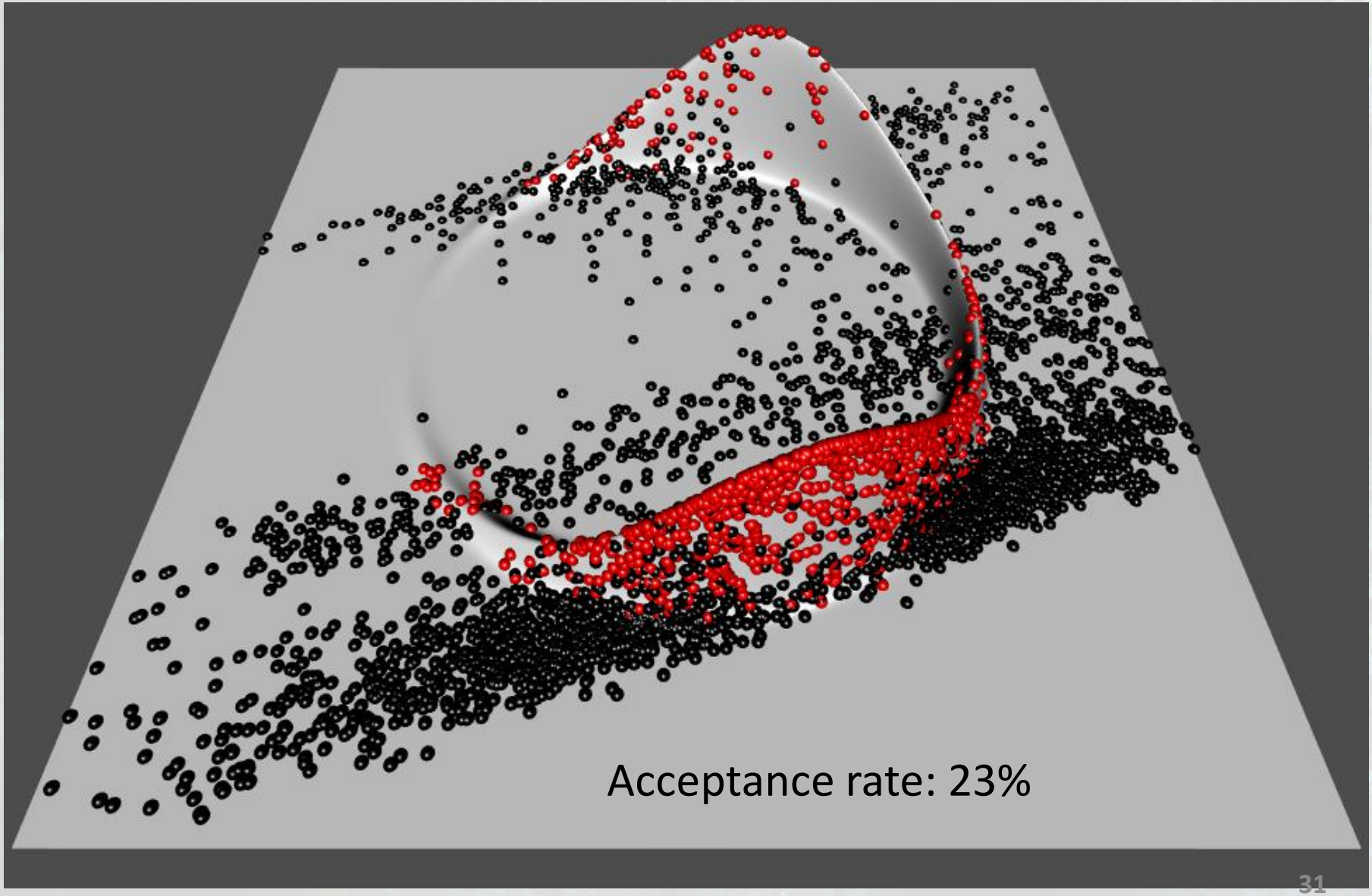
Red points are accepted samples
Black points are rejected samples

ADMB's standard MCMC

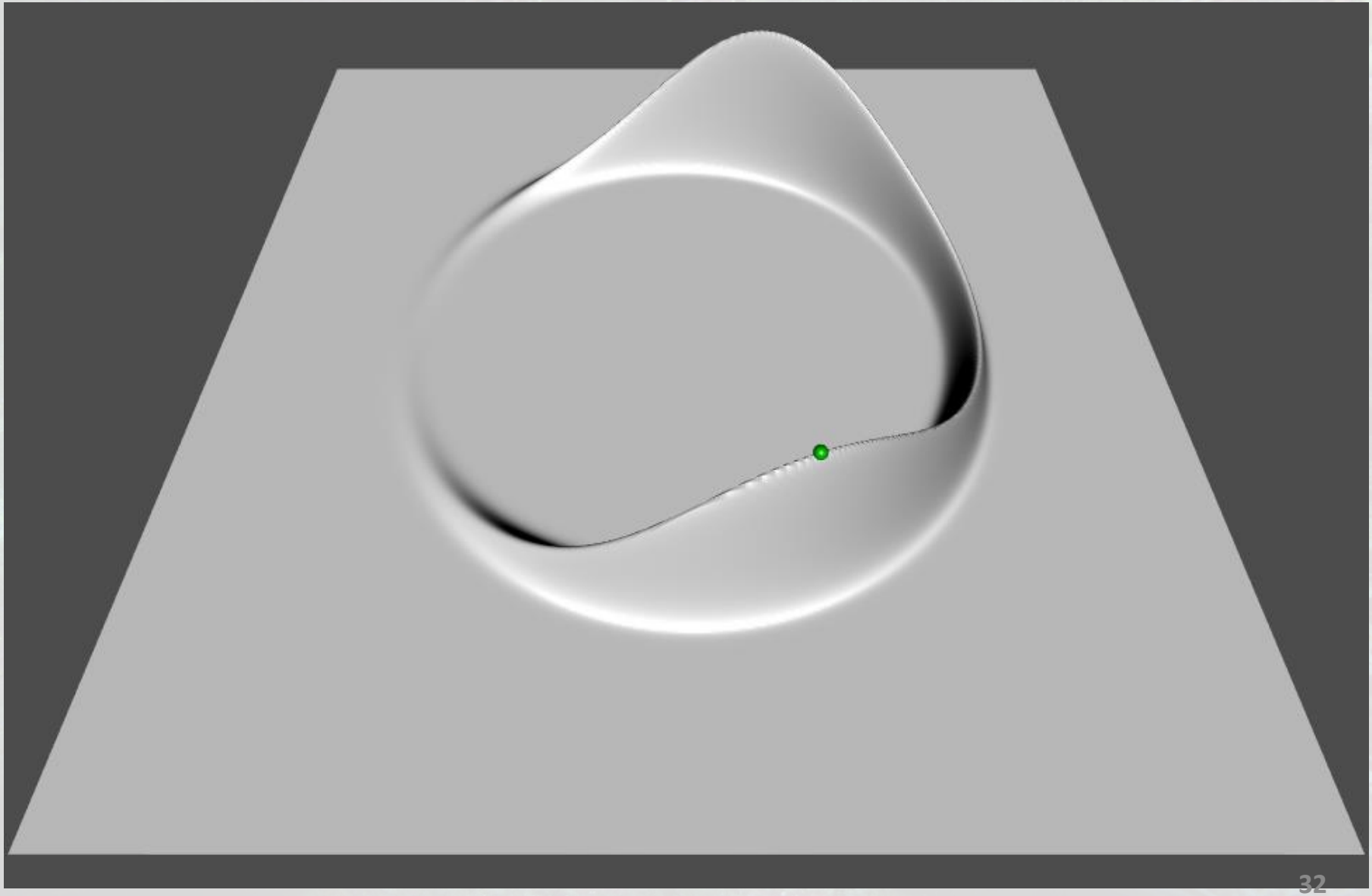


Red points are accepted samples
Black points are rejected samples

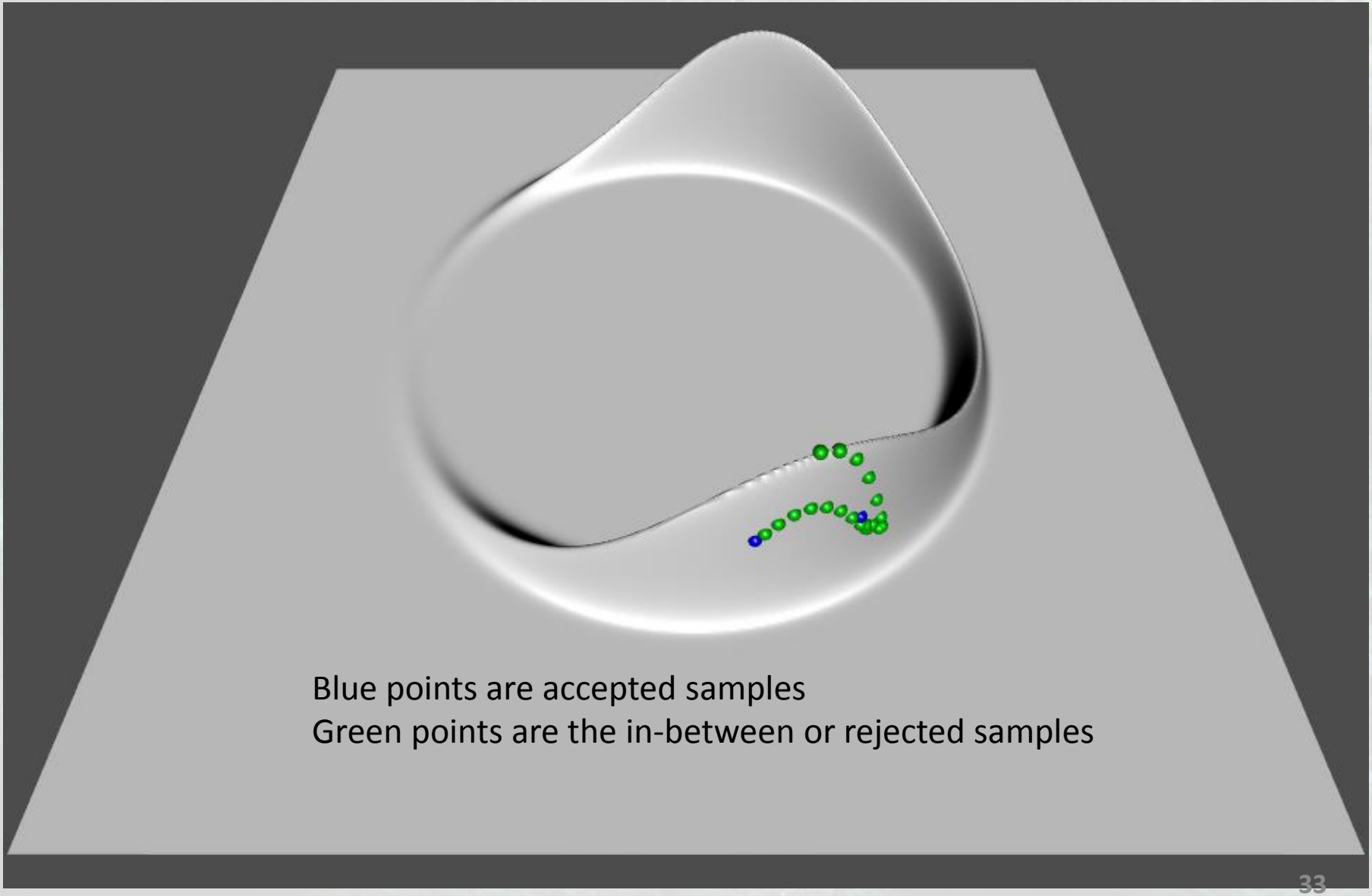
ADMB's standard MCMC



ADMB's "hybrid" MCMC

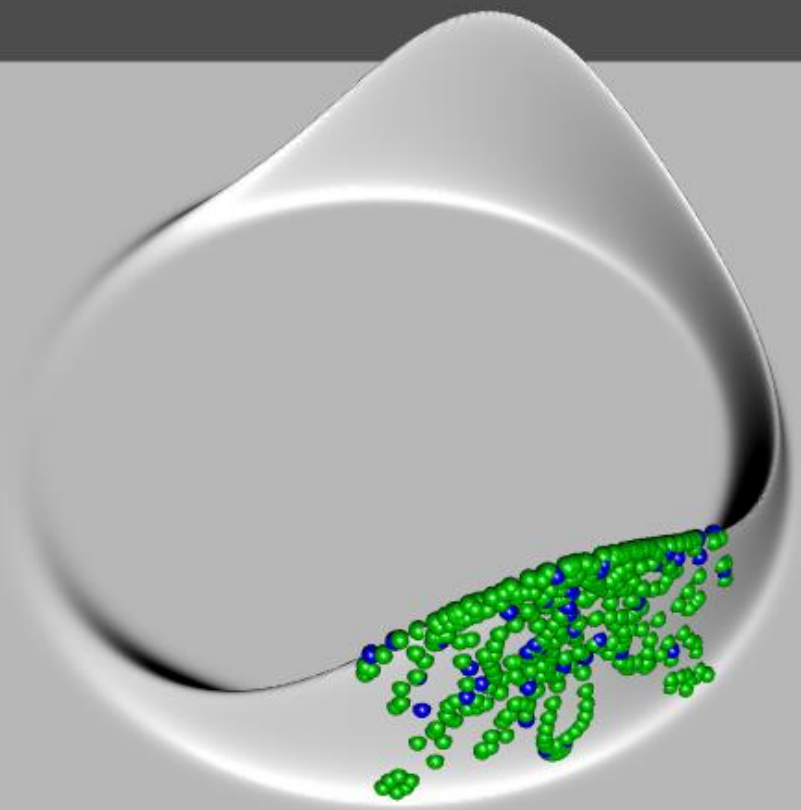


ADMB's "hybrid" MCMC



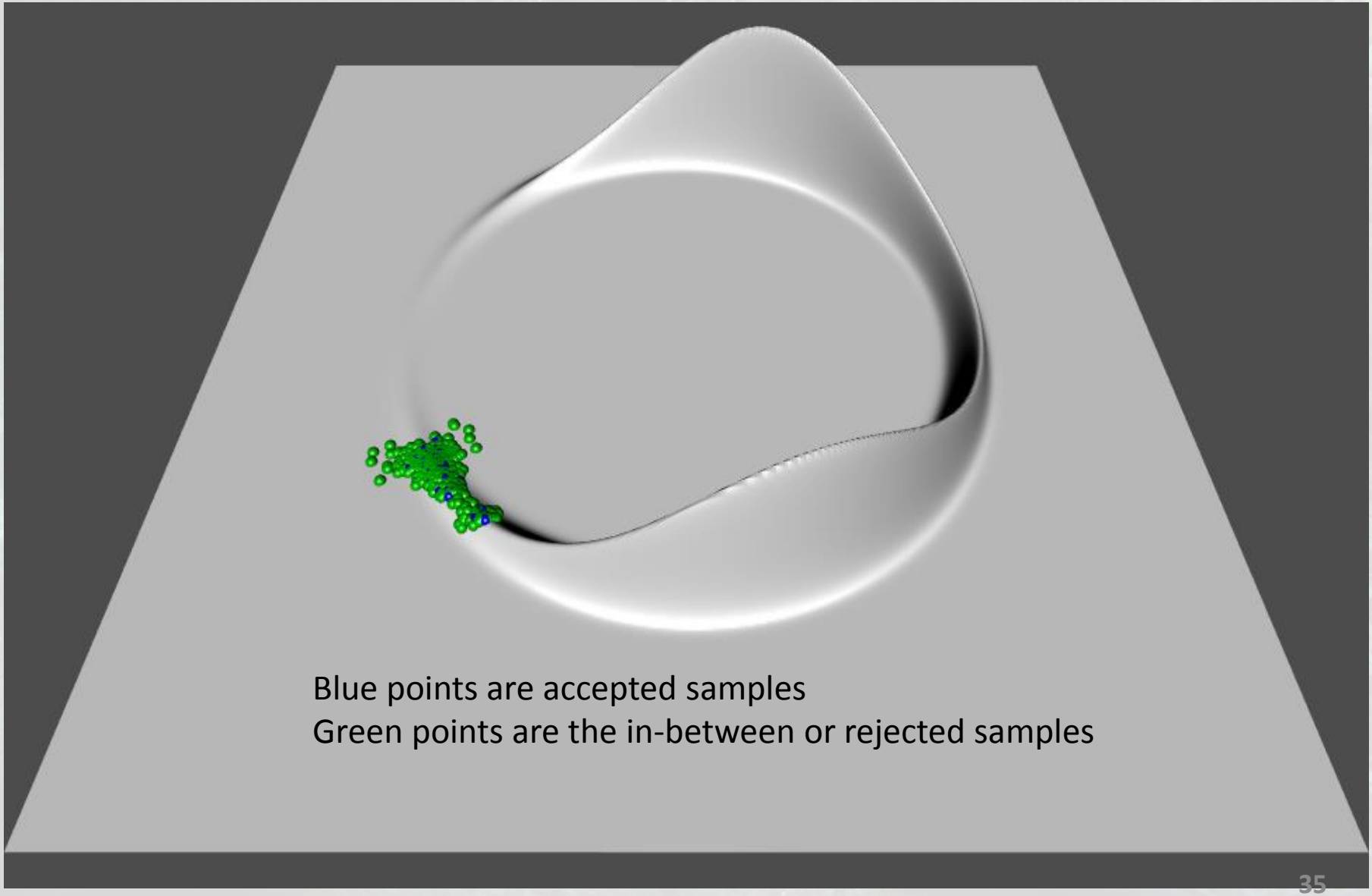
Blue points are accepted samples
Green points are the in-between or rejected samples

ADMB's "hybrid" MCMC



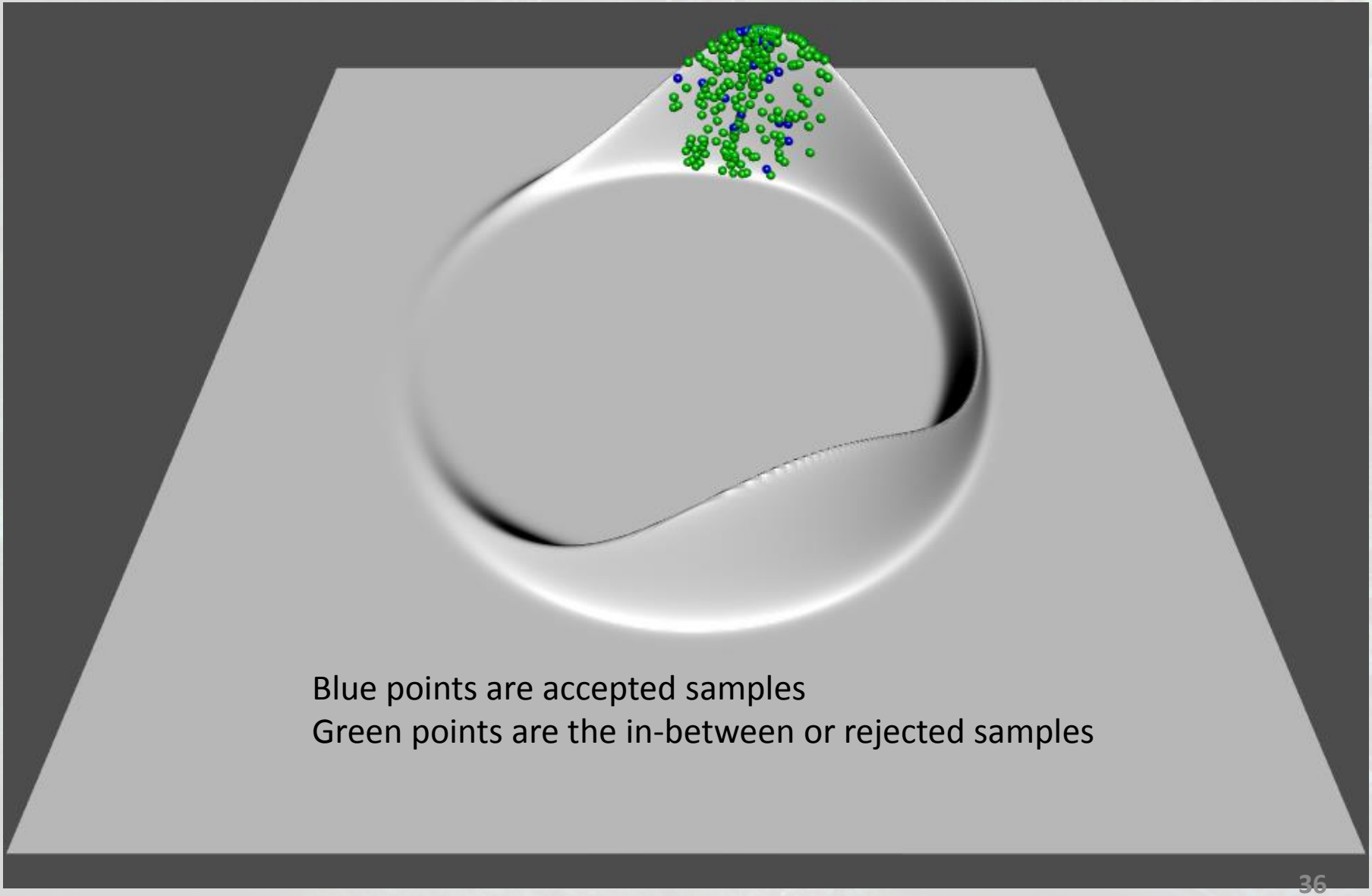
Blue points are accepted samples
Green points are the in-between or rejected samples

ADMB's "hybrid" MCMC



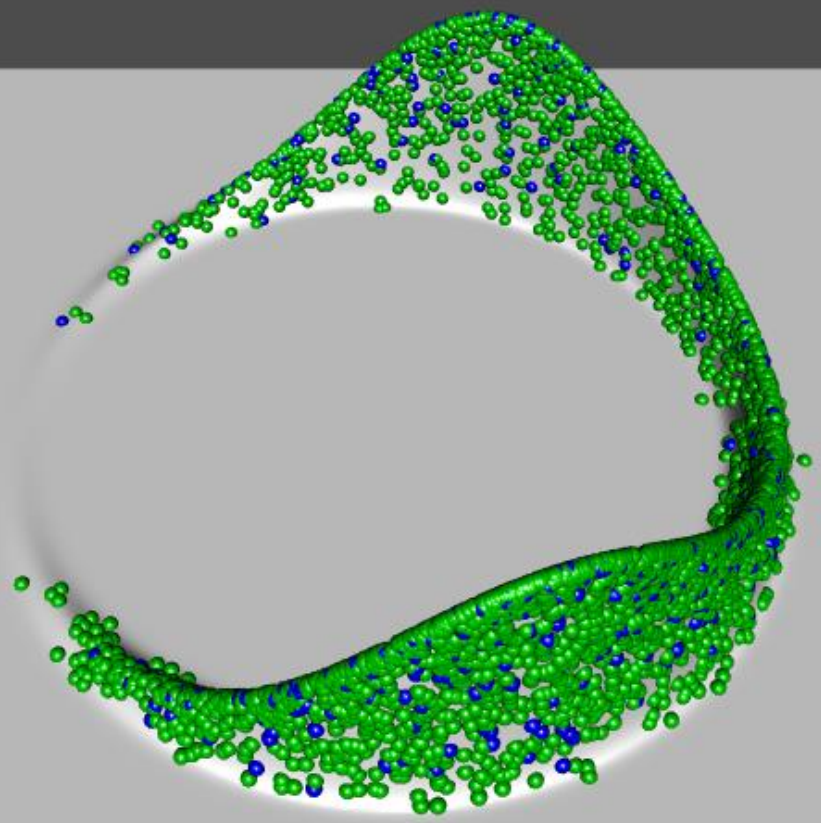
Blue points are accepted samples
Green points are the in-between or rejected samples

ADMB's "hybrid" MCMC



Blue points are accepted samples
Green points are the in-between or rejected samples

ADMB's "hybrid" MCMC



Acceptance rate: 96%
(but requires a bunch of extra steps along the way)

ADMB's "hybrid" MCMC

- Shows promise
- Worth studying to understand better
- Would be nice of additional inputs didn't require manual tuning
- With naïve settings in real-world applications, improved sampling didn't offset slower speed
- Might lower the bar for which models work well with mcmc