# *Simulation-Based Approaches to Bayesian Inference*

**Frank Schorfheide**
Department of Economics
University of Pennsylvania

# Introduction

- For more complicated models it is often infeasible to derive moments of the posterior distribution analytically and to calculate the marginal data density.
- Instead: use algorithm that enables us to generate draws from posterior distribution.
- Based on draws, calculate numerical approximations to moments of interest.
- Important algorithms: Gibbs sampler and Metropolis-Hastings:
  - Gibbs sampler: for instance, for VARs
  - Metropolis-Hastings: for instance, DSGE models
- Markov-Chain Monte Carlo (MCMC) algorithms: construct a Markov Chain with ergodic distribution $p(\theta|Y)$.

# Numerical Integration

- The posterior expectation of a function $h(\theta)$ is equal to the integral

$$\mathbb{E}[h(\theta)|Y] = \int h(\theta)p(\theta|Y)d\theta \tag{1}$$

- **MC Integration**:

(i) Requirements: It is possible to generate draws from the posterior density $p(\theta|Y)$.

(ii) Steps:

1. For $s = 1, \ldots, n_{sim}$ draw $\theta^{(s)}$ from the posterior density $p(\theta|Y)$.
2. Approximate the integral by

$$\mathbb{E}[h(\theta)|Y] \approx \frac{1}{n_{sim}} \sum_{s=1}^{n_{sim}} h(\theta^{(s)}) \quad \square \tag{2}$$

- Use Law of Large Numbers for $h(\theta^{(s)})$ and Central Limit Theorem to assess the approximation error.

# Importance Sampling

- Suppose it is not possible to generate draws from $p(\theta|Y)$ on the computer, but one can calculate the posterior density up to a constant, that is

$$q(\theta|Y) = cp(\theta|Y) \tag{3}$$

and generate draws from a normalized density $g(\theta)$.

- Note that

$$\begin{aligned} I\!\!E[h(\theta)|Y] &= \int h(\theta)p(\theta|Y)d\theta \\ &= \int h(\theta)\frac{p(\theta|Y)}{g(\theta)}g(\theta)d\theta \\ &= \frac{1}{c}\int h(\theta)\frac{q(\theta|Y)}{g(\theta)}g(\theta)d\theta \end{aligned} \tag{4}$$

# Importance Sampling

- Moreover,

$$\int \frac{q(\theta|Y)}{g(\theta)} g(\theta) d\theta = c \int p(\theta|Y) d\theta) = c \tag{5}$$

- Define the importance ratios

$$w(\theta) = q(\theta|Y)/g(\theta) \tag{6}$$

- The above analysis suggests the following algorithm, known as importance sampling...

# Importance Sampling

- **MC Integration through Importance Sampling**:

  (i) Requirements: Posterior density $p(\theta|Y)$ can be evaluated up to a constant: $q(\theta|Y) = cp(\theta|Y)$. It is possible to generate draws from a normalized density $g(\theta)$. The draws need not be independent.

  (ii) Steps:

  1. For $s = 1, \ldots, n_{sim}$ draw $\theta^{(s)}$ from the density $g(\theta)$.
  2. Approximate the integral by

$$\mathbb{E}[h(\theta)|Y] \approx \frac{\frac{1}{n_{sim}} \sum_{s=1}^{n_{sim}} w(\theta^{(s)}) h(\theta^{(s)})}{\frac{1}{n_{sim}} \sum_{s=1}^{n_{sim}} w(\theta^{(s)})} \quad \Box \tag{7}$$

# Importance Sampling

- In principle the approximation error converges to zero for a large class of densities $g(\theta)$.
- The important question is: how fast?
- The algorithm will work well if $g(\theta)$ can be chosen such that
$$h(\theta)w(\theta)$$
stays roughly constant across draws $\theta^{(s)}$.
- The convergence will be very slow if the importance ratios are very small with high probability and very large with low probability.
- Consider CLT: If $\{y_t\}$ is strictly stationary and ergodic with $I\!E[y_1] = 0$, $I\!E[y_1^2] = \sigma^2 < \infty$, and $\bar{\sigma}_T = var(T^{-1/2} \sum y_t) \longrightarrow \bar{\sigma}^2 < \infty$, then
$$\frac{1}{\sqrt{T}\bar{\sigma}_T} \sum y_t \xrightarrow{p} \mathcal{N}(0,1). \quad \square$$

# Data Augmentation Algorithms

- The data augmentation algorithm is based on two simple identities.

- The *posterior identity* is of the form

$$p(\theta|Y) = \int_Z p(\theta|Y,Z)p(Z|Y)dZ \tag{8}$$

- and the *predictive identity* is

$$p(Z|Y) \int p(Z|\theta,Y)p(\theta|Y)d\theta \tag{9}$$

- When substituting the latter identity into the former we obtain

$$\begin{aligned} p(\theta|Y) &= \int_Z p(\theta|Y,Z)\left[\int_{\tilde{\theta}} p(Z|\tilde{\theta},Y)p(\tilde{\theta}|Y)d\tilde{\theta}\right]dZ \\ &= \int_{\tilde{\theta}}\left[\int_Z p(\theta|Y,Z)p(Z|Y,\tilde{\theta})dZ\right]p(\tilde{\theta}|Y)d\tilde{\theta} \\ &= \int_{\tilde{\theta}} K(\theta|\tilde{\theta},Y)p(\tilde{\theta}|Y)d\tilde{\theta} \qquad (10) \end{aligned}$$

# Data Augmentation Algorithms

- Thus, $p(\theta|Y)$ has to satisfy an integral equation. Let $g(\theta)$ be a normalized probability density. Define the mapping

$$M[g(\theta)] = \int K(\theta|\tilde{\theta}, Y) g(\tilde{\theta}) d\tilde{\theta} \tag{11}$$

  $M$ maps a density $g(\theta)$ into a density $g'(\theta)$. Let $g_{(s+1)}(\theta) = M[g_{(s)}(\theta)]$.

- Under suitable regularity conditions (see for instance Tanner, 1996)

  (i) The fixed point $g^*(\theta)$ of the mapping $M[g(\theta)] = g(\theta)$ is unique.

  (ii) The mapping $M$ is a contraction mapping and the sequence of densities $\{g_{(s)}(\theta)\}$ converges to the fixed point $g^*(\theta)$

$$\int |g_{(s)}(\theta) - g^*(\theta)| d\theta \longrightarrow 0$$

  as $s \longrightarrow \infty$. $\Box$

# Data Augmentation Algorithms

- Thus, if one applies the mapping $M$ to some initial density $g_{(0)}(\theta)$ one converges eventually to the solution of the integral equation.

- **Data Augmentation Algorithm**

  (i) Requirements: It is easily possible to generate draws from $p(Z|\theta, Y)$ and $p(\theta|Z, Y)$.

  (ii) The following two steps are repeated for $s = 1, \ldots, n_{sim}$.

  1. Generate a sample $Z_1, \ldots, Z_m$ from the current approximation to $p(Z|Y)$ as follows: For $i = 1, \ldots, m$ draw $\theta_{(i)}$ from $p_{(s)}(\theta|Y)$ and draw $Z_{(i)}$ from $p(Z|\theta_{(i)}, Y)$.
  2. Use the posterior identity to update from $p_{(s)}(\theta|Y)$ to $p_{(s+1)}(\theta|Y)$:

$$p_{(s+1)}(\theta|Y) = \frac{1}{m} \sum_{i=1}^{m} p(\theta|Z_{(i)}, Y)$$

     A draw of $\theta$ can be easily generated by drawing an $i = 1, \ldots, m$ with uniform probability $1/m$ and a $\theta$ from $p(\theta|Z_{(i)}, Y)$

# Data Augmentation Algorithms

- For large $m$ and large $n_{sim}$ the mixture $p_{(s+1)}(\theta|Y)$ with provide a close approximation of $p(\theta|Y)$. In many cases we are not so much interested in the density $p(\theta|Y)$ but rather in random draws from this density. Choosing $m = 1$ leads to *chained data augmentation*.

- **Chained Data Augmentation Algorithm**

  (i) Requirements: It is easily possible to generate draws from $p(Z|\theta, Y)$ and $p(\theta|Z, Y)$.

  (ii) The following two steps are repeated for $s = 1, \ldots, n_{sim}$.

    1. Draw $Z^{(s+1)}$ from the density $p(Z|\theta^{(s)}, Y)$.

    2. Draw $\theta^{(s)}$ from the density $p(\theta|Z^{(s+1)}, Y)$. $\square$

# Data Augmentation Algorithms

- The sequence $\{\theta^{(s)}\}_{s=1}^{n_{sim}}$ in the chained data augmentation forms a Markov Chain with transition Kernel

$$K(\theta|\tilde{\theta}, Y) = \int p(\theta|Y, Z)p(Z|Y, \tilde{\theta})dZ \tag{12}$$

- The stationary distribution of this Markov chain has to satisfy the integral equation

$$p(\theta|Y) = \int K(\theta|\tilde{\theta}, Y)p(\tilde{\theta}|Y)d\tilde{\theta} \tag{13}$$

- Hence for large $s$, $\{\theta^{(s)}\}_{s=1}^{n_{sim}}$ are draws from the posterior distribution $p(\theta|Y)$. Moreover, the $Z$'s obtained from the chained data augmentation algorithm are draws from $p(Z|Y)$. This Fact is exploited by the Gibbs Sampler.

# Data Augmentation Algorithms

- **Gibbs Sampler**

  (i) Requirements: Suppose the parameter vector $\theta$ can be partitioned into $\theta = [\theta_1', \ldots, \theta_m']'$.

  For each $i$ it is possible to generate draws of $\theta_i$ from the conditional distribution $p(\theta_i|\theta_{-i}, Y)$ where $\theta_{-i}$ denotes the vector $\theta$ without the partition $\theta_i$.

  (ii) The following steps are repeated for $s = 1, \ldots, n_{sim}$.

  – Draw $\theta_1^{(s+1)}$ from the density $p(\theta_1|\theta_2^{(s)}, \ldots, \theta_m^{(s)}, Y)$.

  – Draw $\theta_2^{(s+1)}$ from the density $p(\theta_2|\theta_1^{(s+1)}, \theta_3^{(s)}, \ldots, \theta_m^{(s)}, Y)$.

  – $\ldots$

  – Draw $\theta_m^{(s+1)}$ from the density $p(\theta_m|\theta_1^{(s+1)}, \ldots, \theta_{m-1}^{(s+1)}, Y)$. $\square$

# Data Augmentation Algorithms

- For large $s$ we obtain dependent draws from the posterior distribution of $\theta$. It is common practice to discard the initial draws.

- Approximate the mean and covariance matrix of $\theta$ by Monte Carlo averages:

$$\widehat{I\!E[\theta]} = \frac{1}{n_{sim} - n_0} \sum_{s=n_0+1}^{n_{sim}} \theta^{(s)}.$$

# Example: Gibbs Sampler

- Simple example: suppose our posterior is of the form

$$\left[\begin{array}{c}\theta_1 \\ \theta_2\end{array}\right] \sim \mathcal{N}\left( \left[\begin{array}{c}\tilde{\theta}_1 \\ \tilde{\theta}_2\end{array}\right], \left[\begin{array}{cc}\tilde{V}_{11} & \tilde{V}_{12} \\ & \tilde{V}_{22}\end{array}\right] \right).$$

- In practice the posterior mean $\tilde{\theta}$ and variance $\tilde{V}$ are functions of data and prior.

- Let's assume:

$$\left[\begin{array}{c}\theta_1 \\ \theta_2\end{array}\right] \sim \mathcal{N}\left( \left[\begin{array}{c}0.5 \\ -0.5\end{array}\right], \left[\begin{array}{cc}1 & 0.5 \\ & 1\end{array}\right] \right).$$

# Example: Gibbs Sampler

- In our example the conditional distributions are...

- Distribution of $\theta_1|\theta_2$:

$$\mathcal{N}\left(\mu_1 + \sigma_{12}\sigma_{22}^{-1}(\theta_2 - \mu_2)\ ,\ \sigma_{11} - \sigma_{12}\sigma_{22}^{-1}\sigma_{21}\right)$$

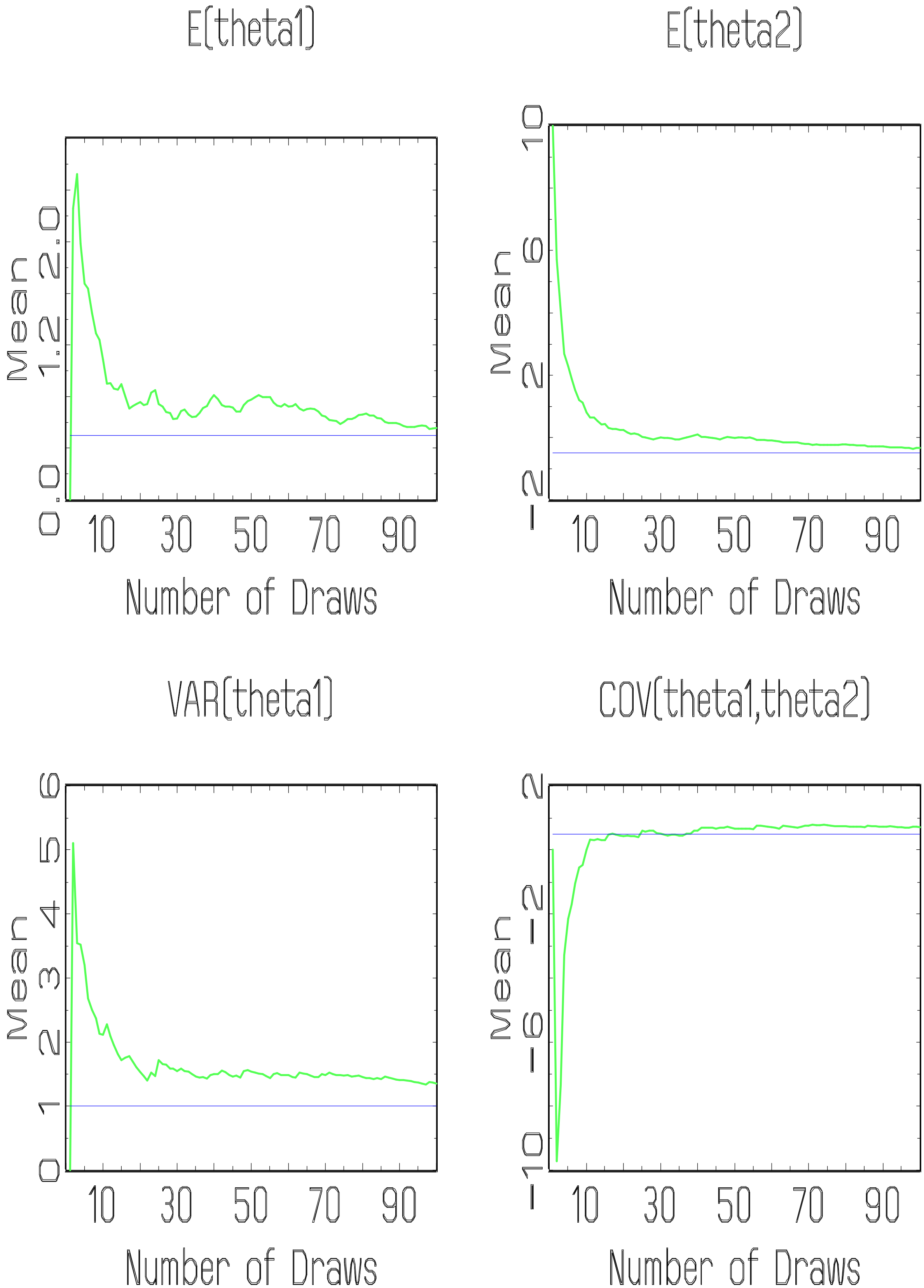
- Distribution of $\theta_2|\theta_1$:

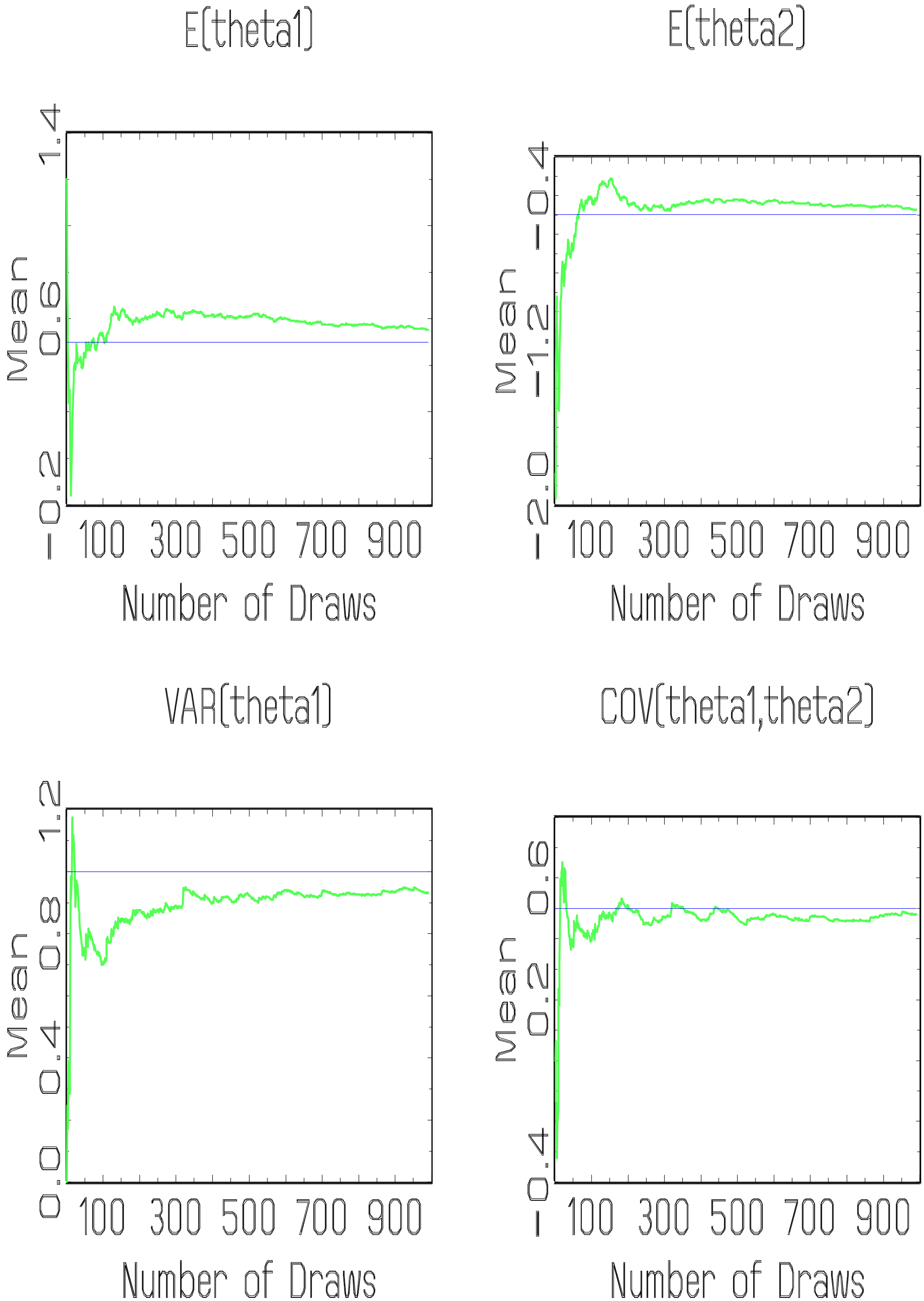
$$\mathcal{N}\left(\mu_2 + \sigma_{21}\sigma_{11}^{-1}(\theta_1 - \mu_1)\ ,\ \sigma_{22} - \sigma_{21}\sigma_{11}^{-1}\sigma_{12}\right)$$

- We consider cumulative means of $\theta$ draws... $(\theta_2^{(0)} = 10)$

- and autocorrelation function of $\theta$ draws.

(Figures)


E(theta1)
E(theta2)
VAR(theta1)
COV(theta1,theta2)
Mean
Number of Draws
10 30 50 70 90
0.0 1.2 2.0
-2 2 6 10
0 1 2 3 4 5 6
-10 -6 -2 2


E(theta1)
E(theta2)
VAR(theta1)
COV(theta1,theta2)
Mean
Number of Draws
100 300 500 700 900

# Metropolis-Hastings Algorithm

- Consider an $m$-state Markov process $x_t$.
- The possible states are denoted by $S = \{s_1, \dots, s_m\}$.
- Transition probabilities:

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1m} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{pmatrix} \tag{14}$$

  where $p_{ij}$ is the probability of moving from state $i$ to state $j$.
- Let $w(t) = [w_1(t), \dots, w_m(t)]$ be a $1 \times m$ vector of probabilities of $x_t$ being in state $i$ in period $t$.
- The corresponding probabilities for period $t+1$ are $w(t+1) = w(t)P$.

# Metropolis-Hastings Algorithm

- $\pi$ is equilibrium distribution if $\pi = \pi P$.

- A Markov chain is reversible if probability of $i \mapsto j$ is the same as $j \mapsto i$:

$$\pi_i p_{ij} = \pi_j p_{ji}$$

- A chain that is reversible has an equilibrium distribution $\pi$ because

$$(\pi P)_j = \sum_{i=1}^{m} \pi_i p_{ij} = \sum_{i=1}^{m} \pi_j p_{ji} = \pi_j \sum_{i=1}^{m} p_{ji} = \pi_j \tag{15}$$

- To sample from the equilibrium distribution, one can start the chain from any $w(0)$ until it settles down to the equilibrium distribution.

# Metropolis-Hastings Algorithm

- Idea of Metropolis algorithm: construct transition matrix $P$ from a transition matrix $Q$ such that $P$ has desired equilibrium distribution $\pi$.

- Why: we don't know how to draw from the posterior $p(\theta|Y^T)$ (corresponds to $\pi$ in our example), but we know how to draw from a normal distribution (corresponds to using $Q$).

- Time $t$ iteration: suppose we are in state $s_i$ Based on $Q$ draw a proposed state $s_j$.

  - With probability $\alpha_{ij}$ proposed state is accepted. Move from $s_i$ to $s_j$.

  - With probability $1 - \alpha_{ij}$ stay in state $s_i$.

- Choice of $\alpha_{ij}$?

# Metropolis-Hastings Algorithm

- Choose $\alpha_{ij} = \min\,[1, \pi_j/\pi_i]$.
- The resulting chain is reversible and has equilibrium distribution $\pi$:

$$
\begin{aligned}
\pi_i p_{ij} &= \pi_i \min[1, \pi_j/\pi_i] q_{ij} \\
&= \min[\pi_i, \pi_j] q_{ij} \\
&= \min[\pi_i, \pi_j] q_{ji} \\
&= \pi_j p_{ji} \qquad (16)
\end{aligned}
$$

- Weak regularity conditions on $Q$ can ensure that the equilibrium distribution $\pi$ is unique and that the chain is convergent.
- These ideas can be generalized to the continuous case.

# Metropolis-Hastings Algorithm

- Special Case: Random Walk Metropolis Algorithm
  - Initialization: Choose a $\theta^{(0)}$ to initialize the chain.
  - Step $s_1$: Draw candidate parameter vector $\psi$ from a jumping distribution $J_s(\psi|\theta^{(s-1)})$.

    Example: draw $\psi$ from the distribution

    $$\psi \sim \mathcal{N}(\theta^{(s-1)}, c^2 I)$$

    where $I$ is the identity matrix and $c$ is a scalar tuning parameter for the algorithm.
  - Step $s_2$:

    $$\theta^{(s)} = \begin{cases} \psi & \text{with probability } \min\left\{1, \frac{p(\psi|Y)}{p(\theta^{(s-1)}|Y)}\right\} \\ \theta^{(s-1)} & \text{otherwise} \end{cases}$$

    We refer to $\theta^{(s)} = \theta^{(s-1)}$ as "rejection" of the proposed step.

  Execute steps $s_1$ and $s_2$ for $s = 1, \ldots, n_{sim}$.

# Metropolis-Hastings Algorithm

- We saw: the sequence of draws $\{\theta^{(s)}\}$ is serially correlated.

- It typically satisfies a weak law of large numbers, that is,

$$\hat{I\!E}[\theta|Y^T] = \frac{1}{n_s}\sum_{s=1}^{n_s} \theta^{(s)} \stackrel{p}{\longrightarrow} \int \theta p(\theta|Y^T)d\theta. \tag{17}$$

- Under some regularity conditions it also satisfies a central limit theorem, meaning: Newey-West standard errors could be used to get numerical standard errors.

- Of course we can transform $\theta^{(s)}$ and evaluate:

$$\hat{I\!E}[g(\theta)|Y^T] = \frac{1}{n_s}\sum_{s=1}^{n_s} g(\theta^{(s)}) \stackrel{p?}{\longrightarrow} \int g(\theta)p(\theta|Y^T)d\theta. \tag{18}$$

  Use for: standard deviations, impulse response functions, etc...

- Univariate confidence intervals (connected): (i) sort draws, (ii) search for shortest interval that contains a pre-specified fraction, say 95 percent, of the draws.

# Convergence?

- Issue: speed of convergence of Markov Chain and Monte Carlo averages.

- Informal assessment of convergence: plot

$$\frac{1}{n_s}\sum_{s=1}^{n_s} g(\theta^{(s)}) \tag{19}$$

  as a function of $n_s$.

- Start Markov-Chain at over-dispersed (i.e., extreme) values of $\theta$ and check whether different runs of the chain settle to the same distribution.

- Huge literature, here some references: Gelman, Carlin, Stern and Rubin (1995), Tanner (1996), Geweke (1999).

# Example: Metropolis-Hastings Algorithm

- Simple example: suppose our posterior is a mixture of normals of the form

$$\left[\begin{array}{c}\theta_1\\ \theta_2\end{array}\right] \sim \left\{\begin{array}{ll} \mathcal{N}\left(\left[\begin{array}{c}\tilde{\theta}_{11}\\ \tilde{\theta}_{12}\end{array}\right], \left[\begin{array}{cc}\tilde{V}_{11} & \tilde{V}_{12}\\ & \tilde{V}_{22}\end{array}\right]\right) & \text{with probability } \frac{1}{2}\\ \mathcal{N}\left(\left[\begin{array}{c}\tilde{\theta}_{21}\\ \tilde{\theta}_{22}\end{array}\right], \left[\begin{array}{cc}\tilde{V}_{11} & \tilde{V}_{12}\\ & \tilde{V}_{22}\end{array}\right]\right) & \text{with probability } \frac{1}{2}\end{array}\right.$$

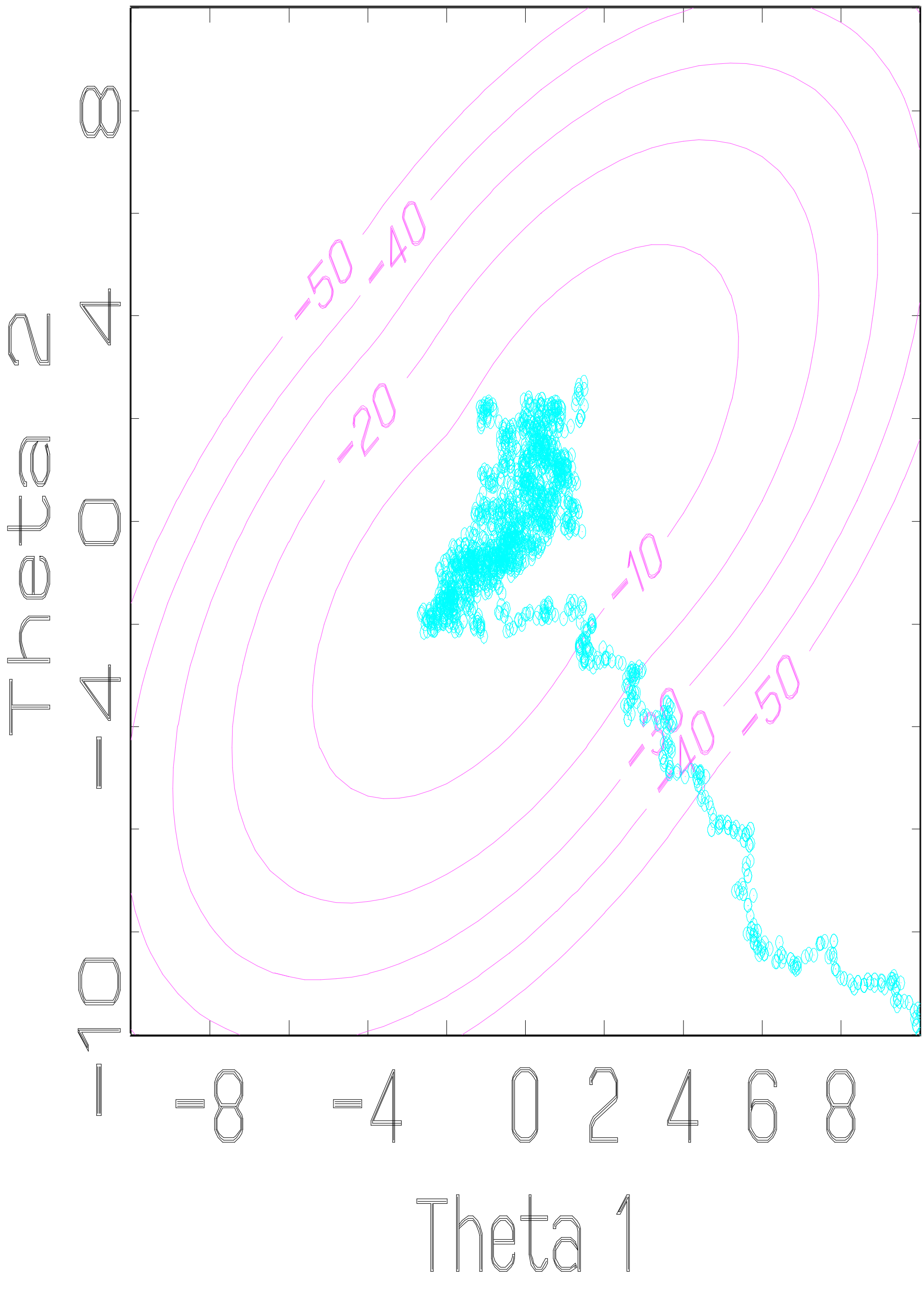
# Example: Metropolis-Hastings Algorithm

- Simple example: suppose our posterior is a mixture of normals of the form

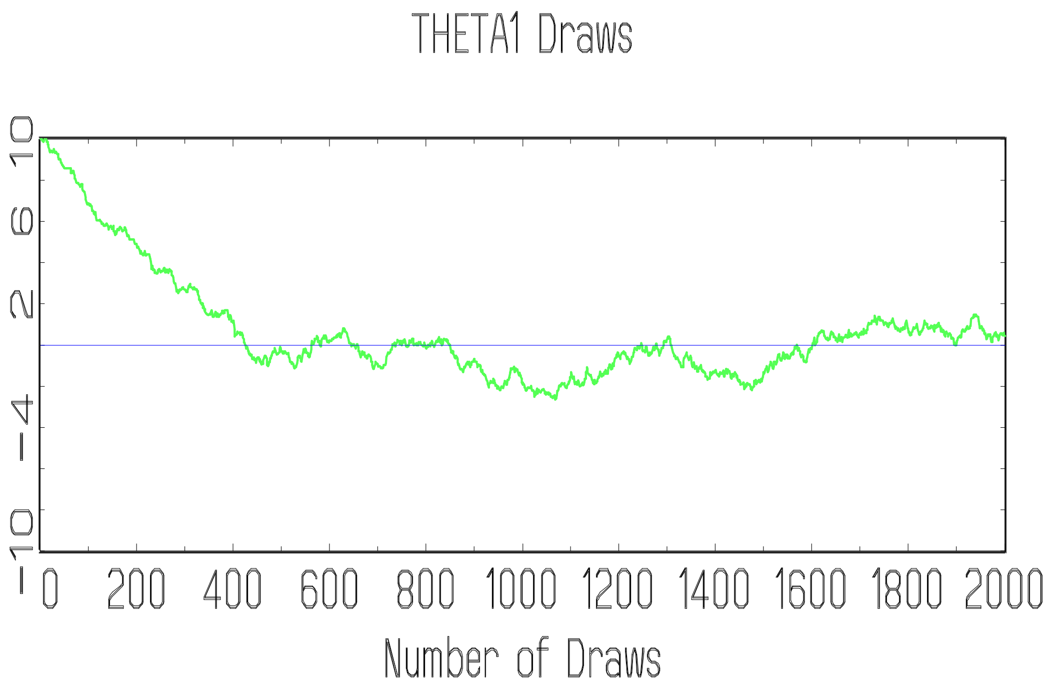
$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \sim \begin{cases} \mathcal{N}\left( \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \\ \mathcal{N}\left( \begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \end{cases}$$

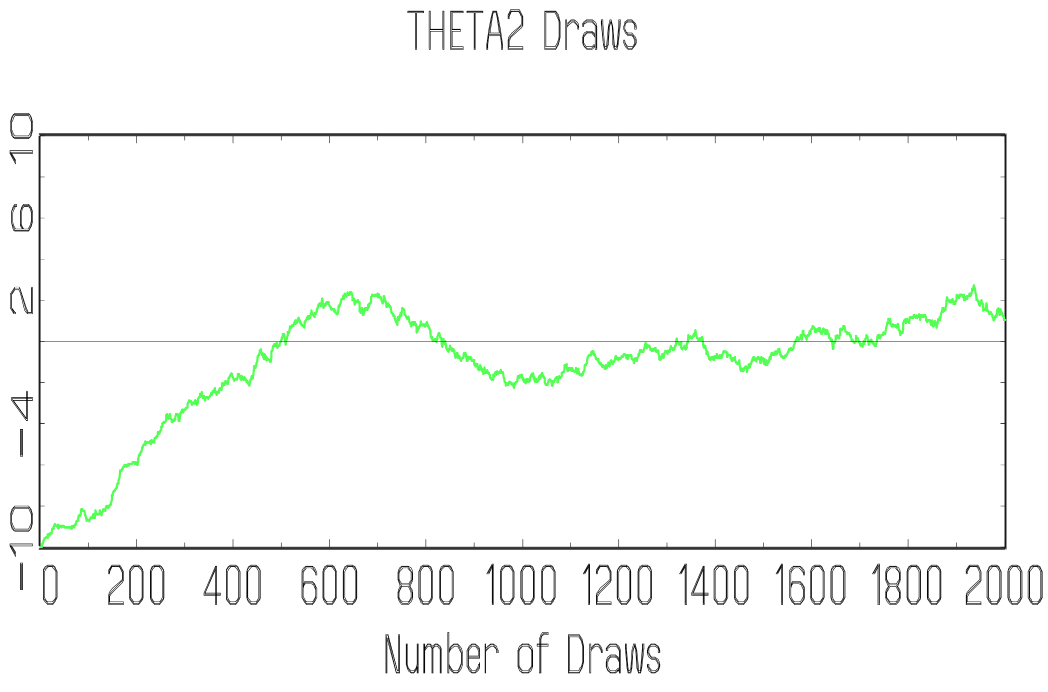
- Configuration of Algorithm
  - Starting values: $\theta_1^{(1)} = 10$, $\theta_2^{(1)} = -10$
  - Proposal density: $\mathcal{N}(\theta^{(s-1)}, 0.1^2 * \mathcal{I})$
  - Number of draws: 2000

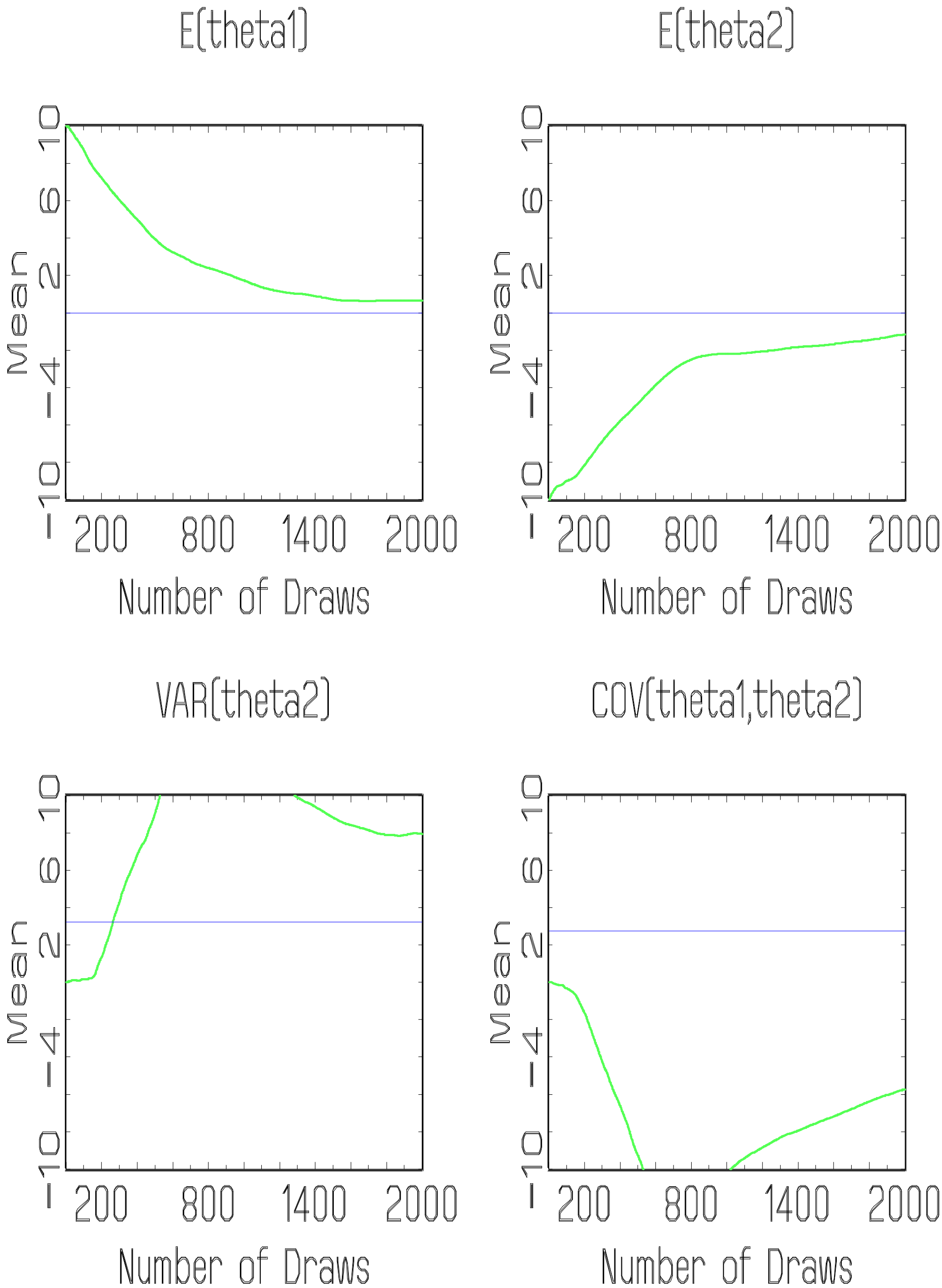
- Small steps... Rejection rate: 8.45 %.



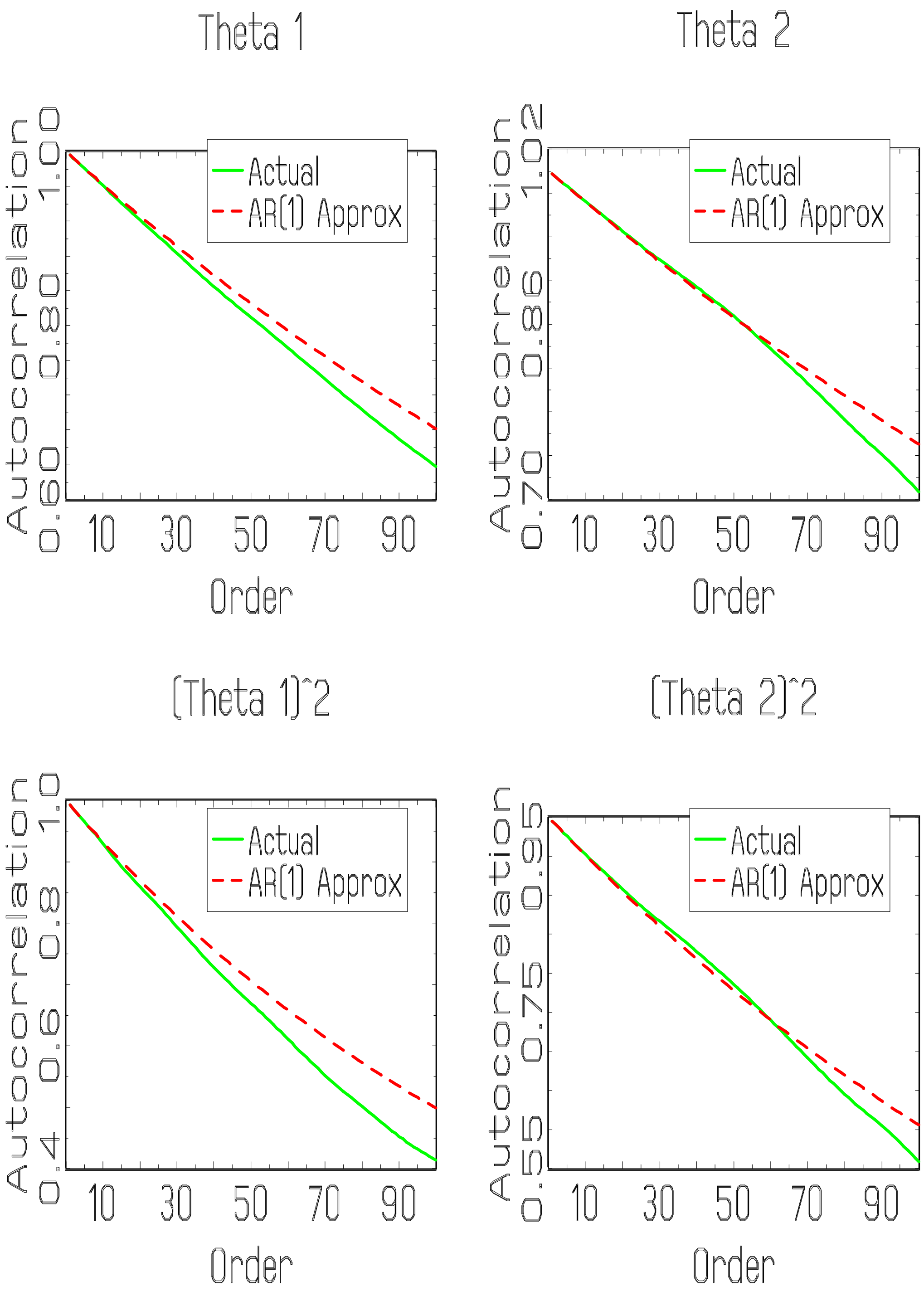
Theta 1
Theta 2
-8
-4
0
2
4
6
8
-10
-4
0
4
8
-50
-40
-20
-10
-30
-40
-50


THETA1 Draws
10
6
2
-4
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Number of Draws


THETA2 Draws
10
6
2
-4
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Number of Draws



E(theta1)
E(theta2)
VAR(theta2)
COV(theta1,theta2)
Mean
10
6
2
−4
−10
200
800
1400
2000
Number of Draws


Theta 1
Theta 2
(Theta 1)^2
(Theta 2)^2
Actual
AR(1) Approx
Autocorrelation
Order
10 30 50 70 90
1.00 0.80 0.60
1.02 0.86 0.70
1.0 0.8 0.6 0.4
0.95 0.75 0.55

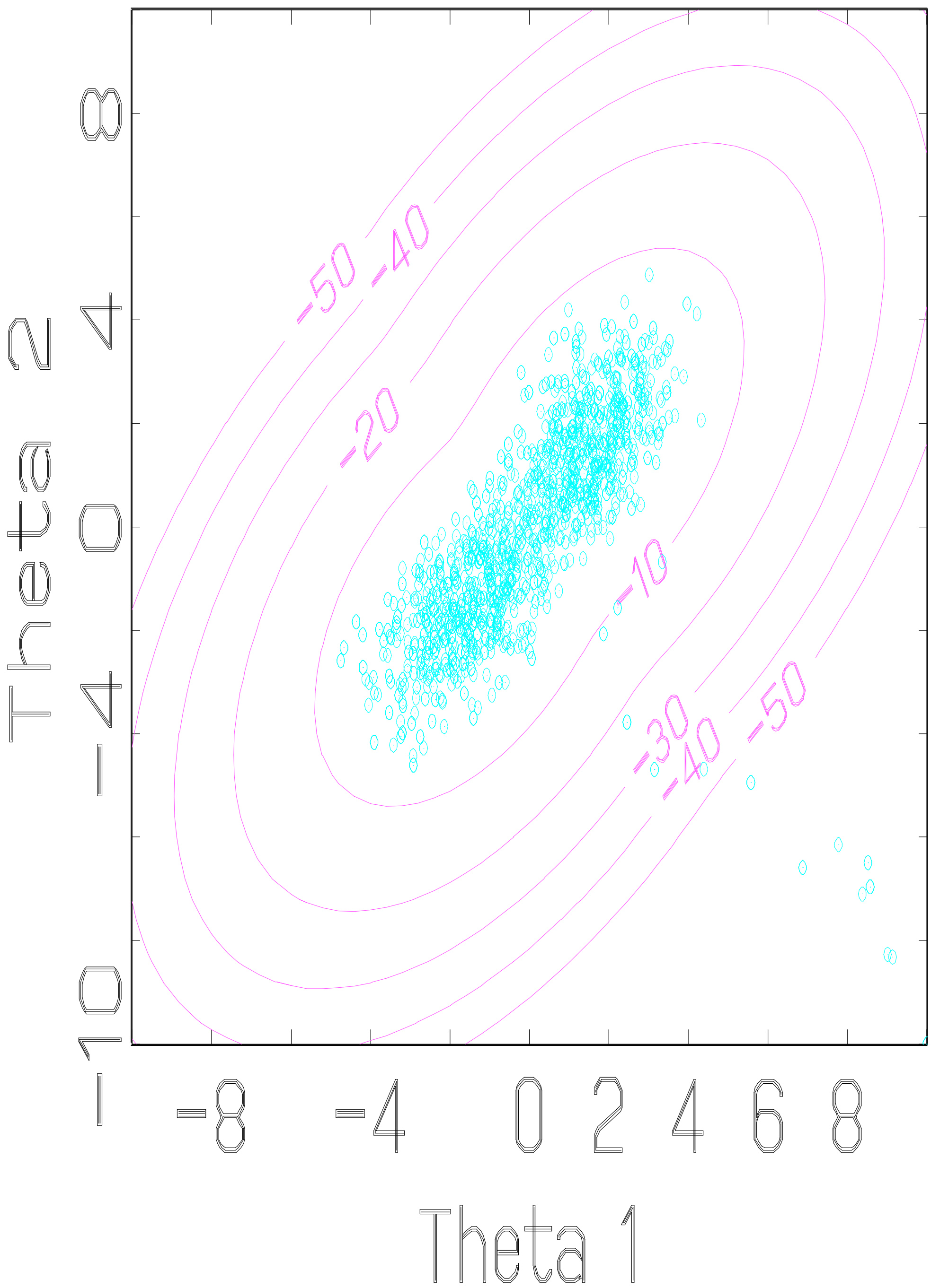
# Example: Metropolis-Hastings Algorithm

- Simple example: suppose our posterior is a mixture of normals of the form

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \sim \begin{cases} \mathcal{N}\left( \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \\ \mathcal{N}\left( \begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \end{cases}$$

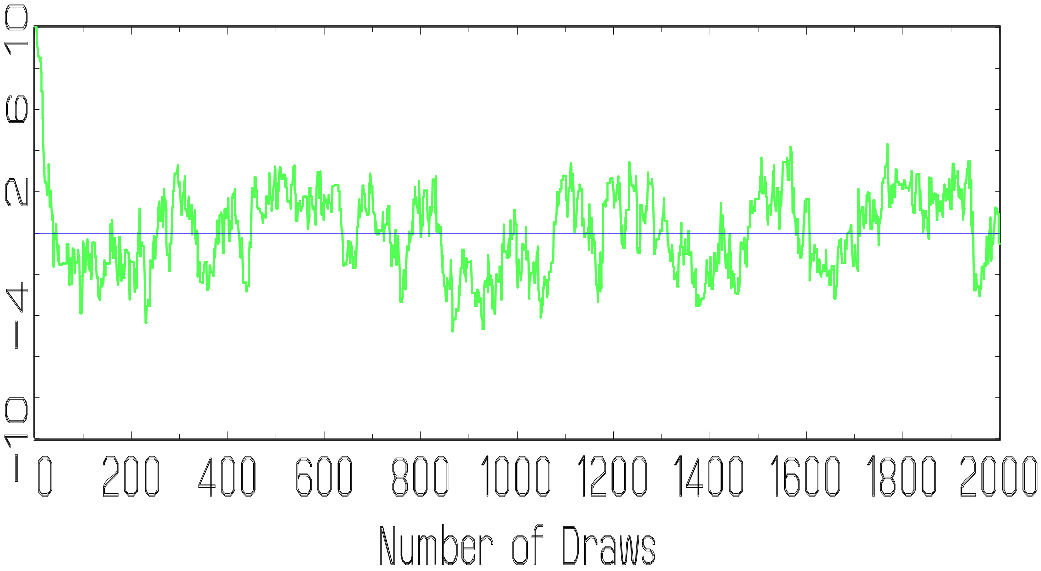
- Configuration of Algorithm
  - Starting values: $\theta_1^{(1)} = 10$, $\theta_2^{(1)} = -10$
  - Proposal density: $\mathcal{N}(\theta^{(s-1)}, 1^2 * \mathcal{I})$
  - Number of draws: 2000

- Mid-size steps... Rejection rate: 45.25 %.

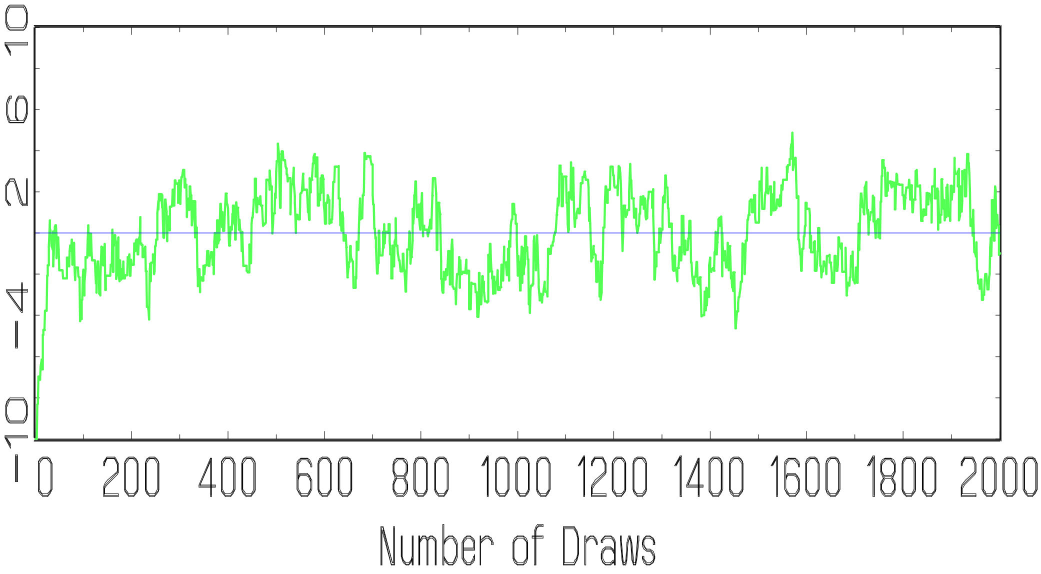


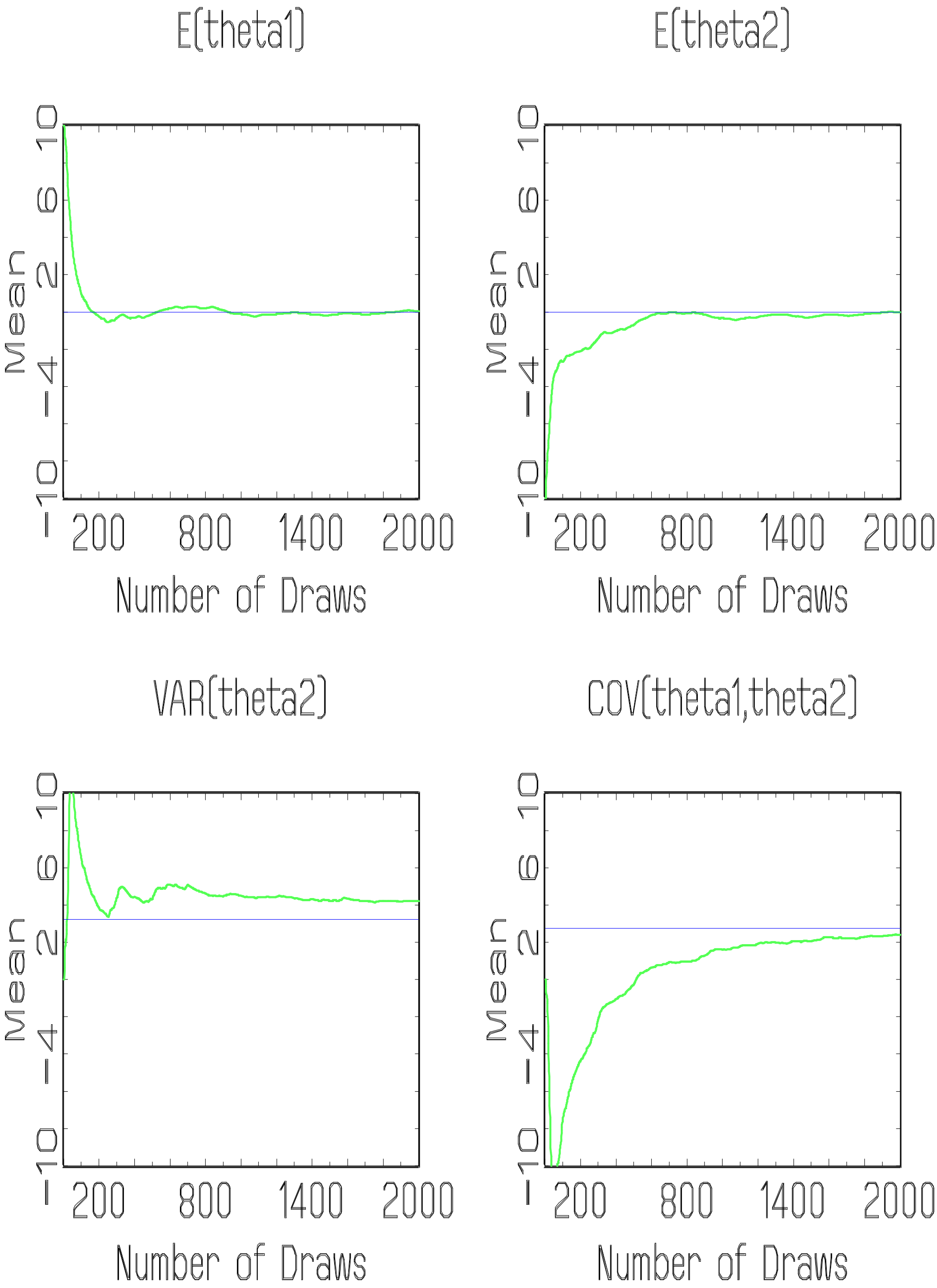
Theta 2
Theta 1
-10
-4
0
4
8
-8
-4
0
2
4
6
8
-50
-40
-20
-10
-30
-40
-50

THETA1 Draws




THETA2 Draws





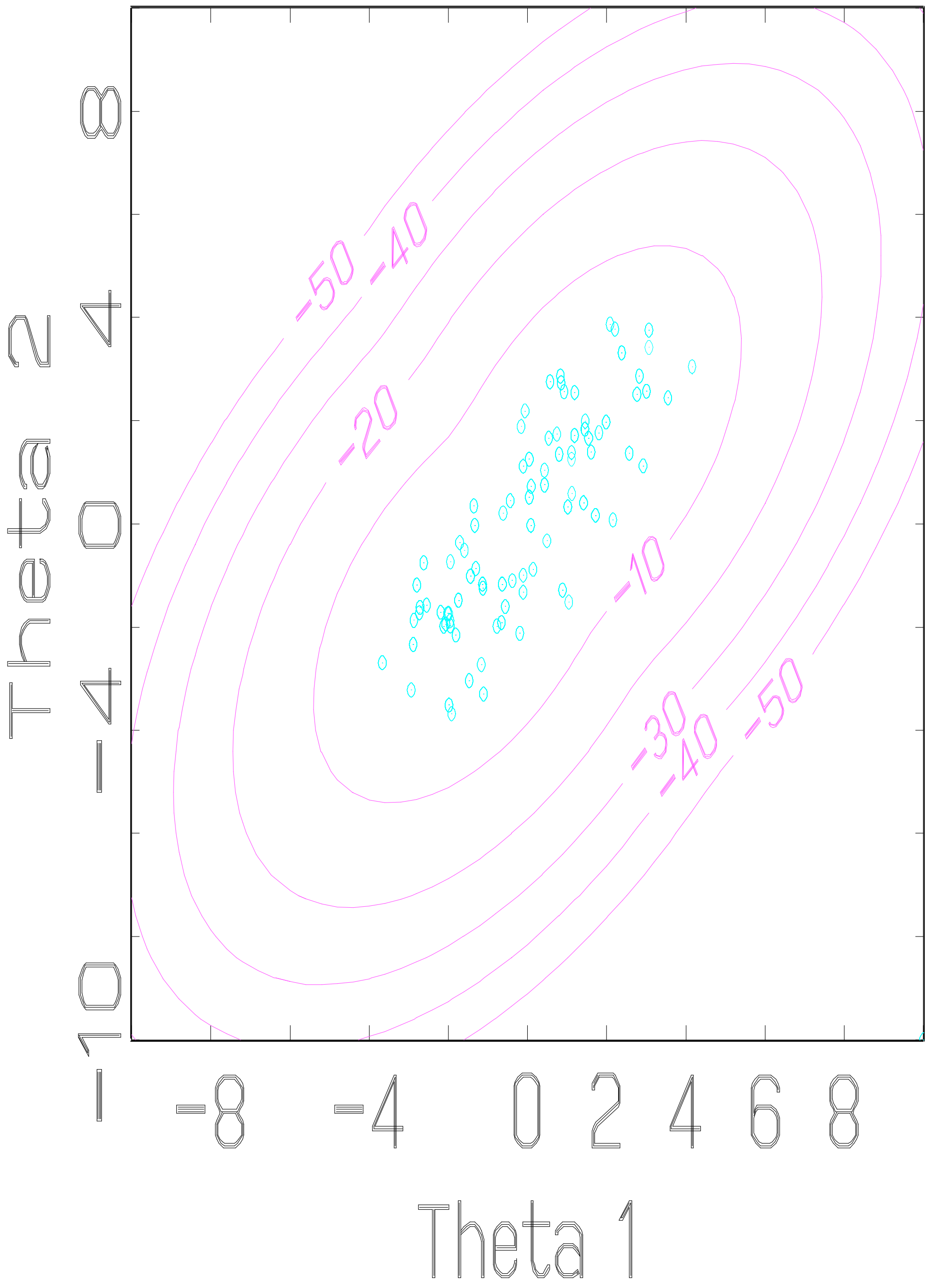
E(theta1)
E(theta2)
VAR(theta2)
COV(theta1,theta2)
Mean
10
6
2
−4
−10
200
800
1400
2000
Number of Draws

# Example: Metropolis-Hastings Algorithm

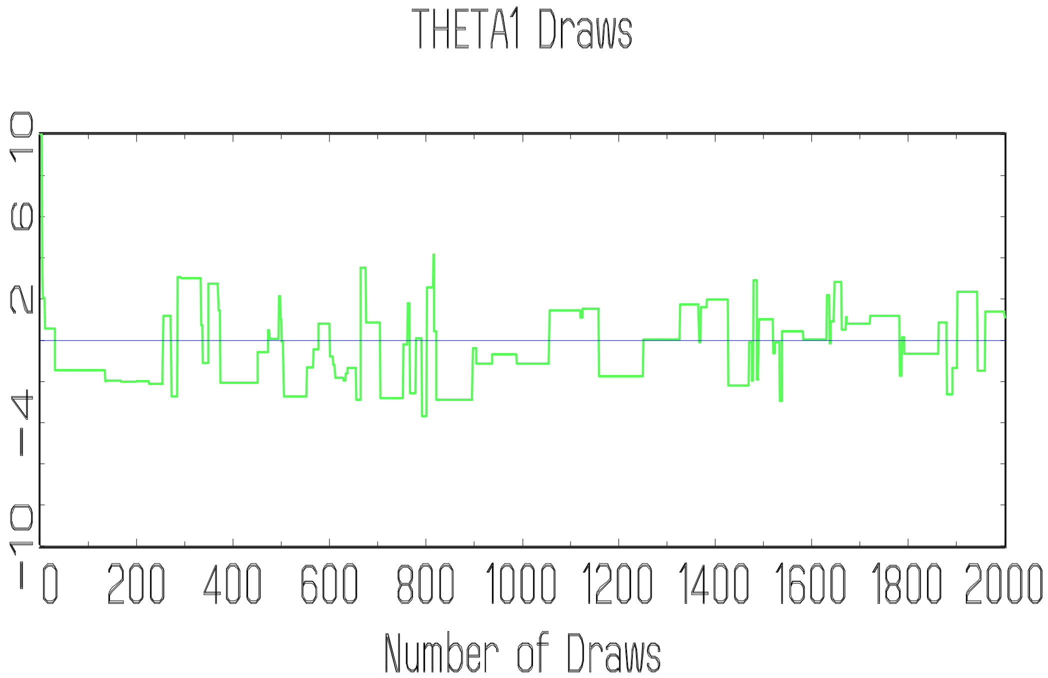
- Simple example: suppose our posterior is a mixture of normals of the form

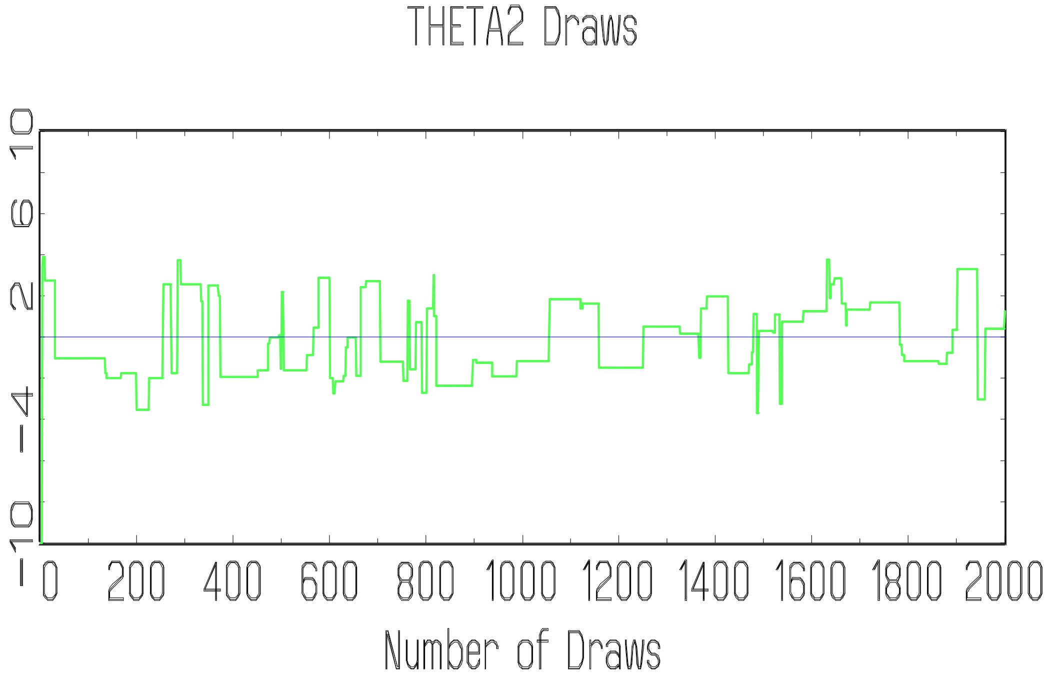
$$\left[\begin{array}{c}\theta_1\\ \theta_2\end{array}\right] \sim \left\{\begin{array}{ll}\mathcal{N}\left(\left[\begin{array}{c}1.5\\ 1.5\end{array}\right],\left[\begin{array}{cc}1 & 0.5\\ & 1\end{array}\right]\right) & \text{with probability } \frac{1}{2}\\ \mathcal{N}\left(\left[\begin{array}{c}-1.5\\ -1.5\end{array}\right],\left[\begin{array}{cc}1 & 0.5\\ & 1\end{array}\right]\right) & \text{with probability } \frac{1}{2}\end{array}\right.$$

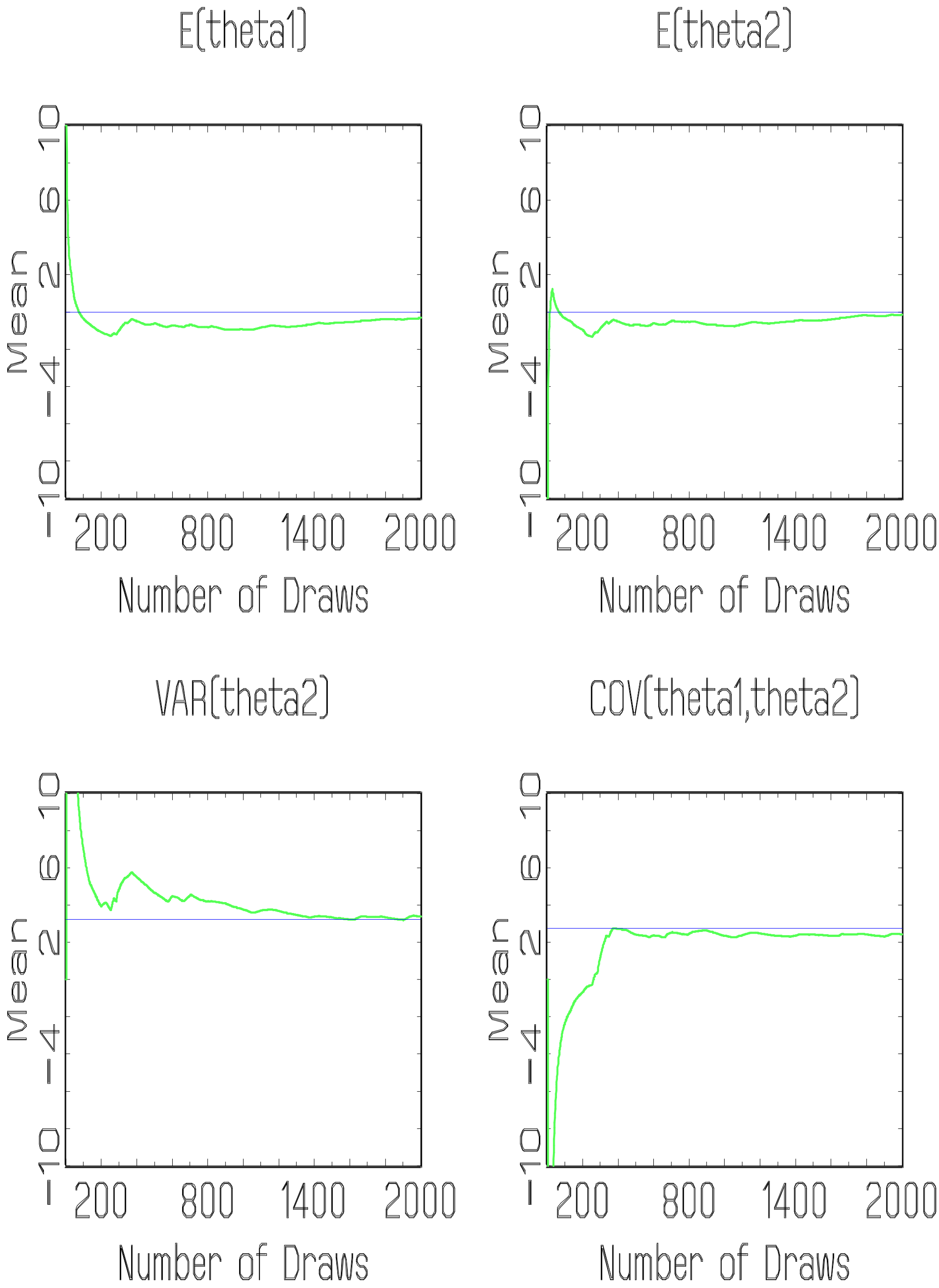
- Configuration of Algorithm
  - Starting values: $\theta_1^{(1)} = 10$, $\theta_2^{(1)} = -10$
  - Proposal density: $\mathcal{N}(\theta^{(s-1)}, 8^2 * \mathcal{I})$
  - Number of draws: 2000
- Large steps... Rejection rate: 95.50 %.



Theta 2
Theta 1
-10
-4
0
4
8
-8
-4
0
2
4
6
8
-50
-40
-20
-10
-30
-40
-50


THETA1 Draws
10
6
2
-4
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Number of Draws


THETA2 Draws
10
6
2
-4
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Number of Draws

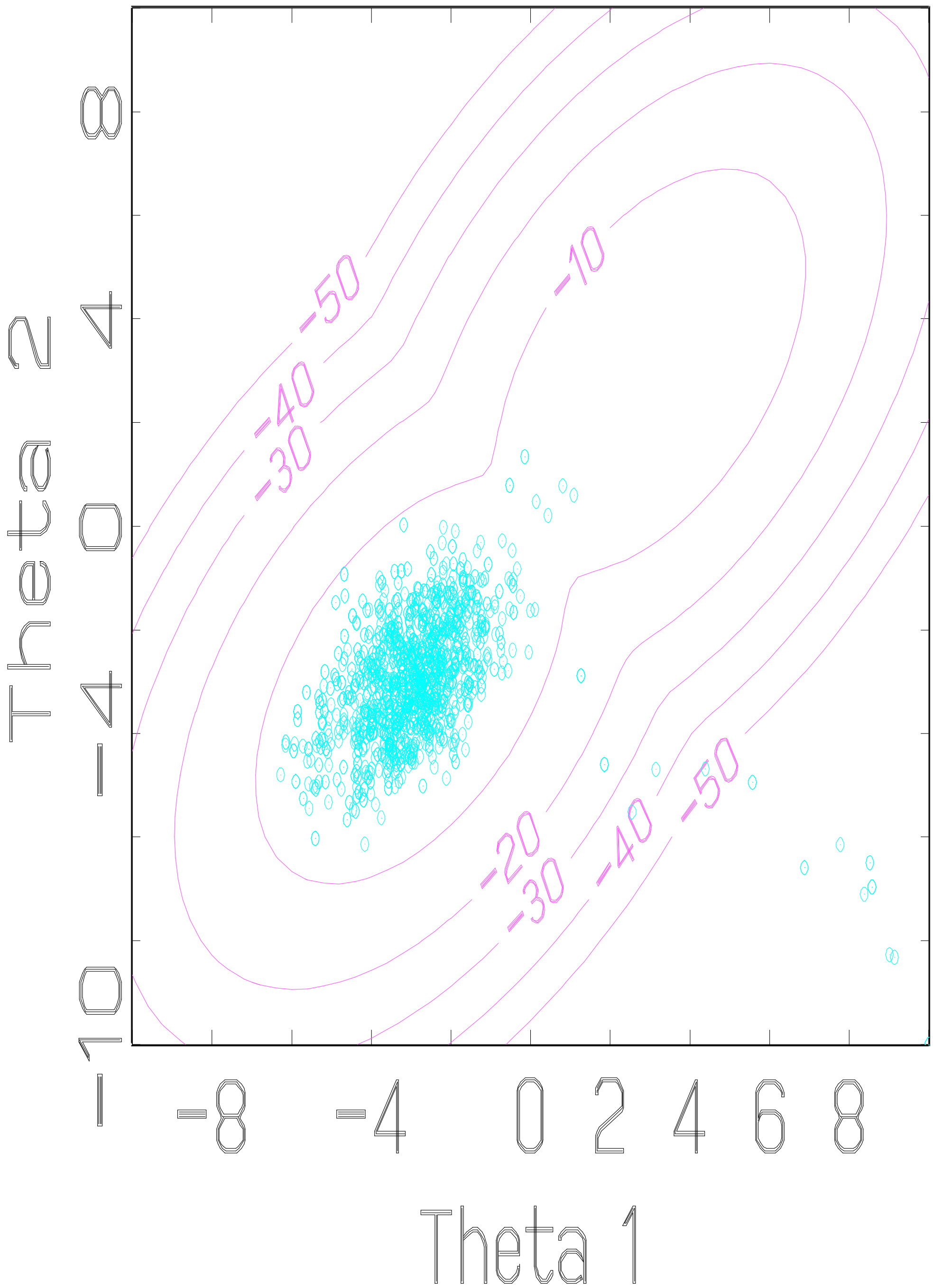

E(theta1)
E(theta2)
VAR(theta2)
COV(theta1,theta2)
Mean
−10 −4 2 6 10
Number of Draws
200 800 1400 2000

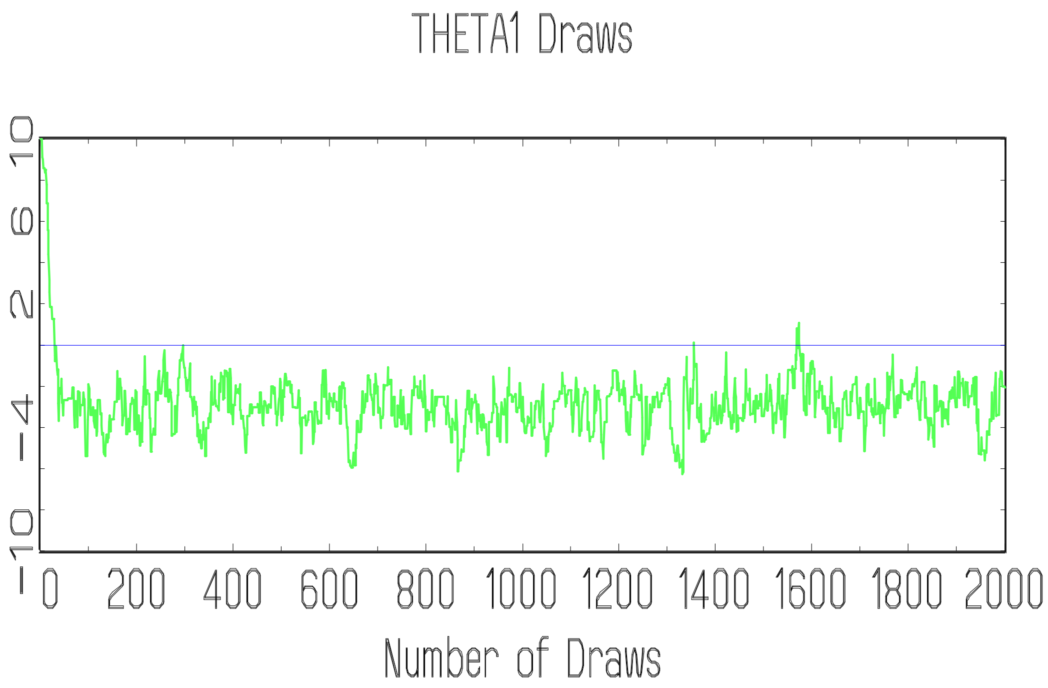
# Example: Metropolis-Hastings Algorithm

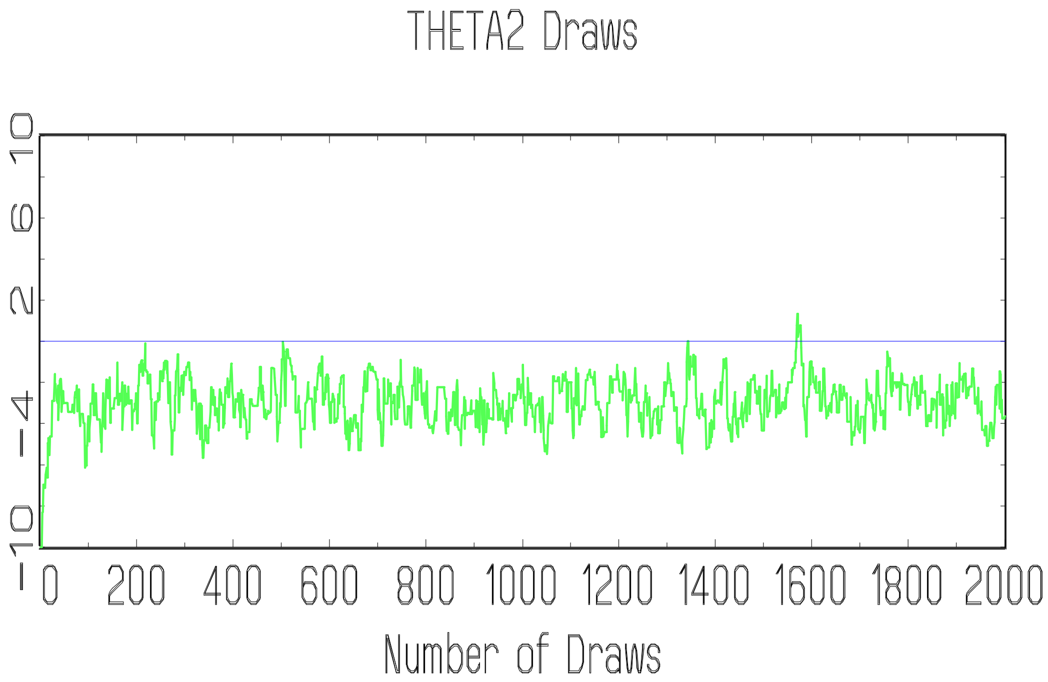
- Move modes of mixture further apart...

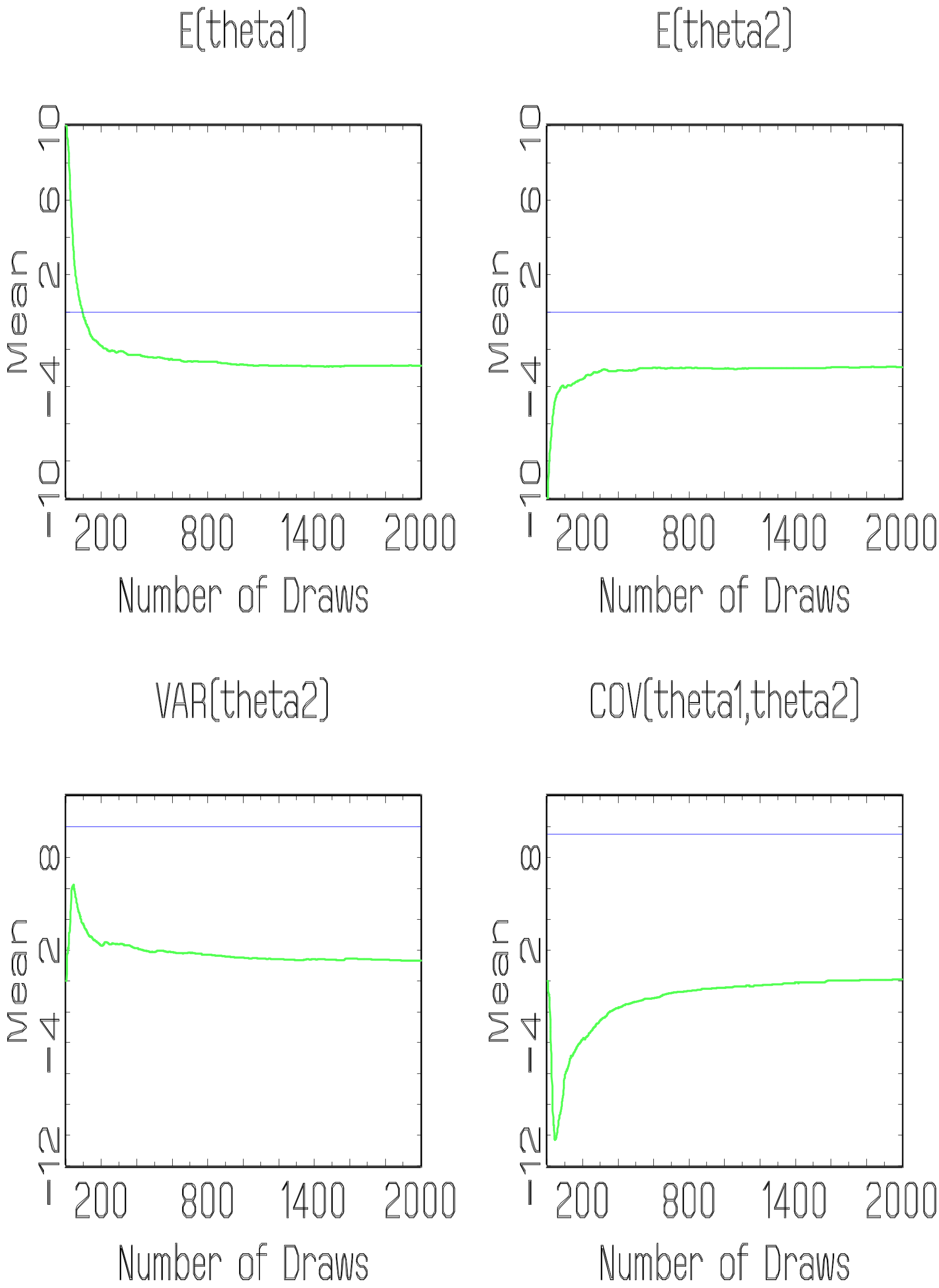
$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \sim \begin{cases} \mathcal{N}\left( \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \\ \mathcal{N}\left( \begin{bmatrix} -3 \\ -3 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \end{cases}$$

- Configuration of Algorithm
  - Starting values: $\theta_1^{(1)} = 10$, $\theta_2^{(1)} = -10$
  - Proposal density: $\mathcal{N}(\theta^{(s-1)}, 1^2 * \mathcal{I})$
  - Number of draws: 2000
- Rejection rate: 48.35 %.


Theta 1
Theta 2
-8
-4
0
2
4
6
8
-10
-4
0
4
8
-10
-20
-30
-40
-50
-30
-40
-50


THETA1 Draws
10
6
2
-4
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Number of Draws


THETA2 Draws
10
6
2
-4
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Number of Draws

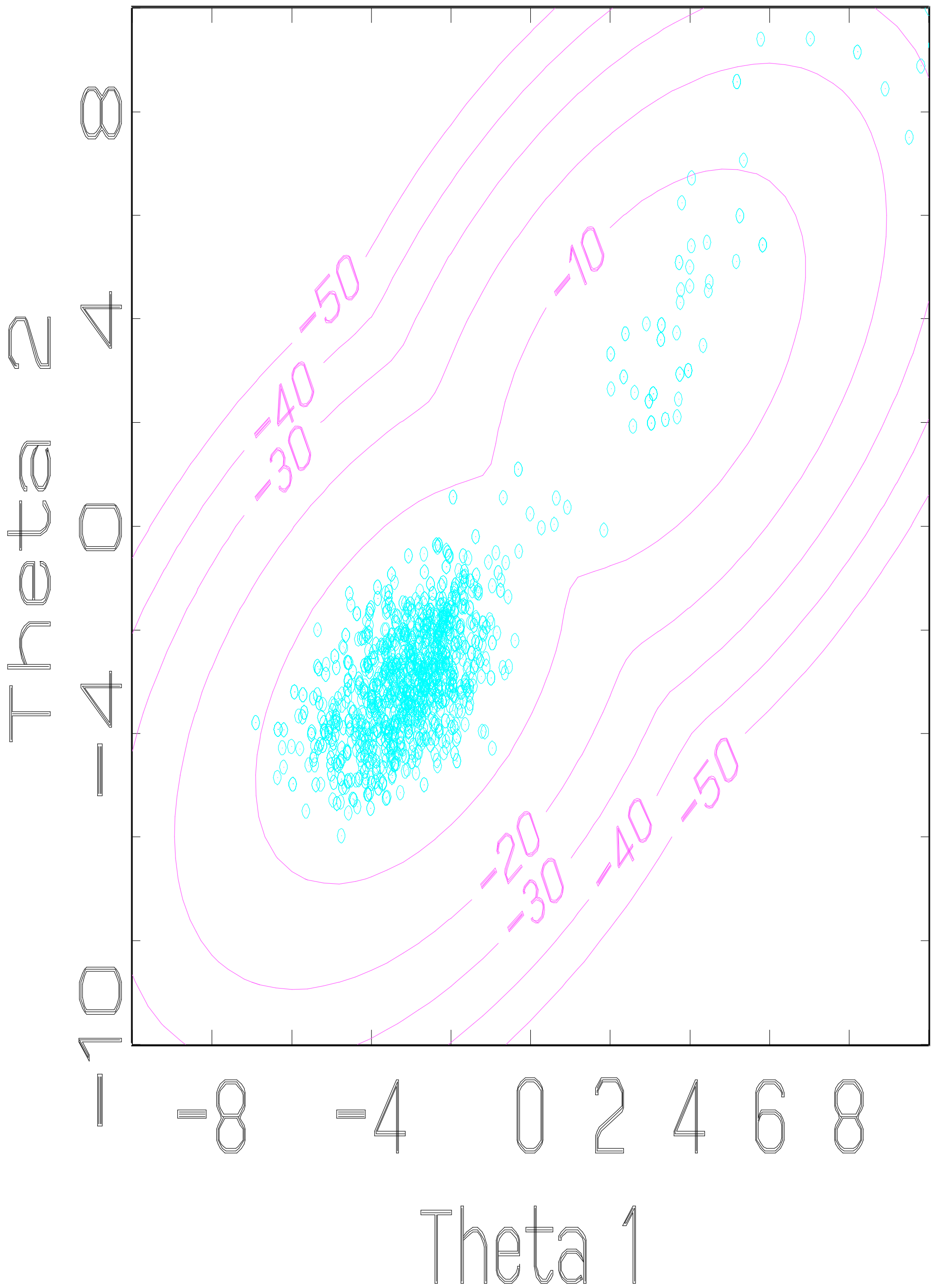

E(theta1)
E(theta2)
VAR(theta2)
COV(theta1,theta2)
Mean
Number of Draws
200
800
1400
2000
10
6
2
−4
−10
8
−12

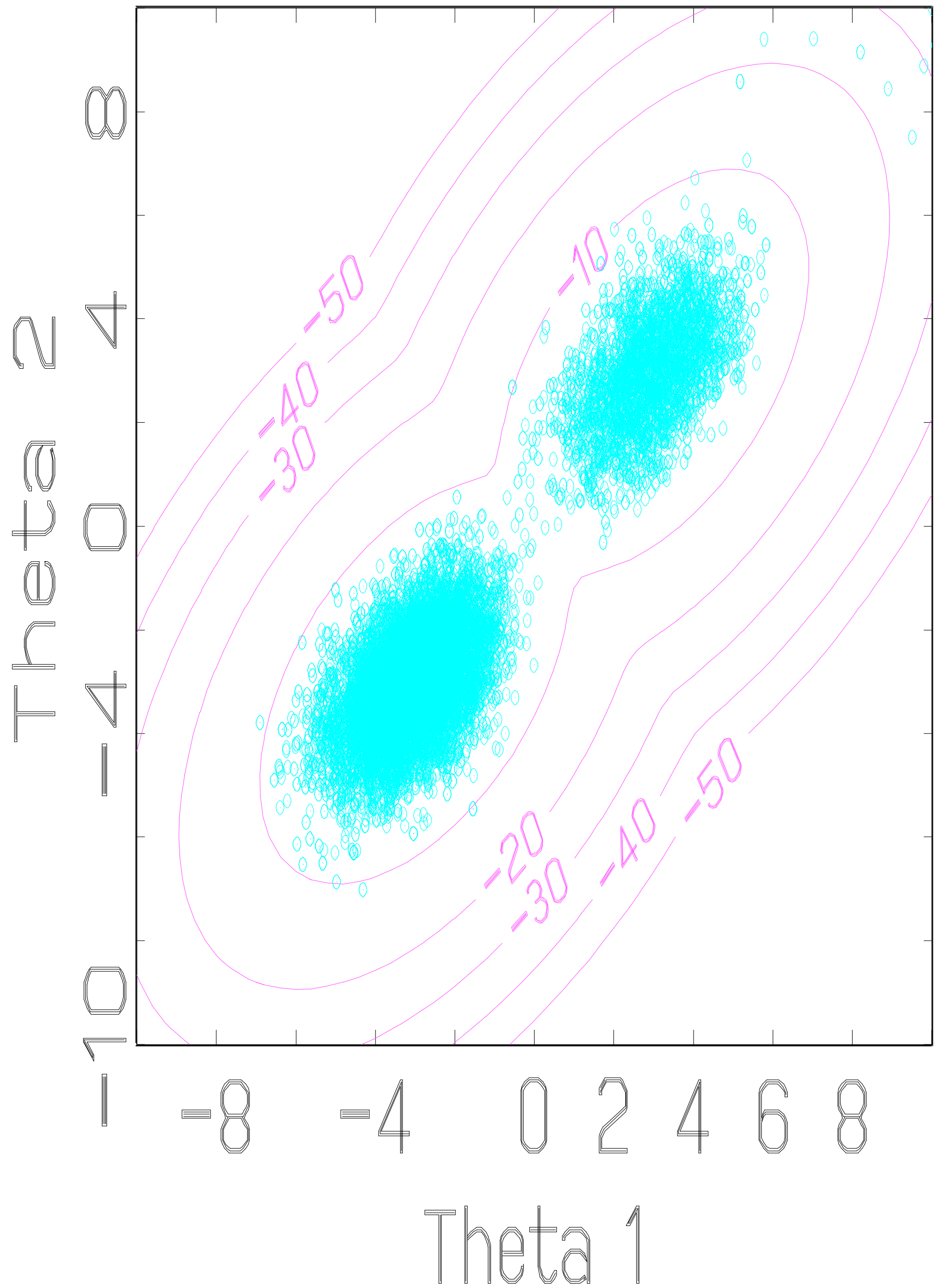
# Example: Metropolis-Hastings Algorithm

- Simple example: increase number of draws...

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \sim \begin{cases} \mathcal{N}\left( \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \\ \mathcal{N}\left( \begin{bmatrix} -3 \\ -3 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ & 1 \end{bmatrix} \right) & \text{with probability } \frac{1}{2} \end{cases}$$

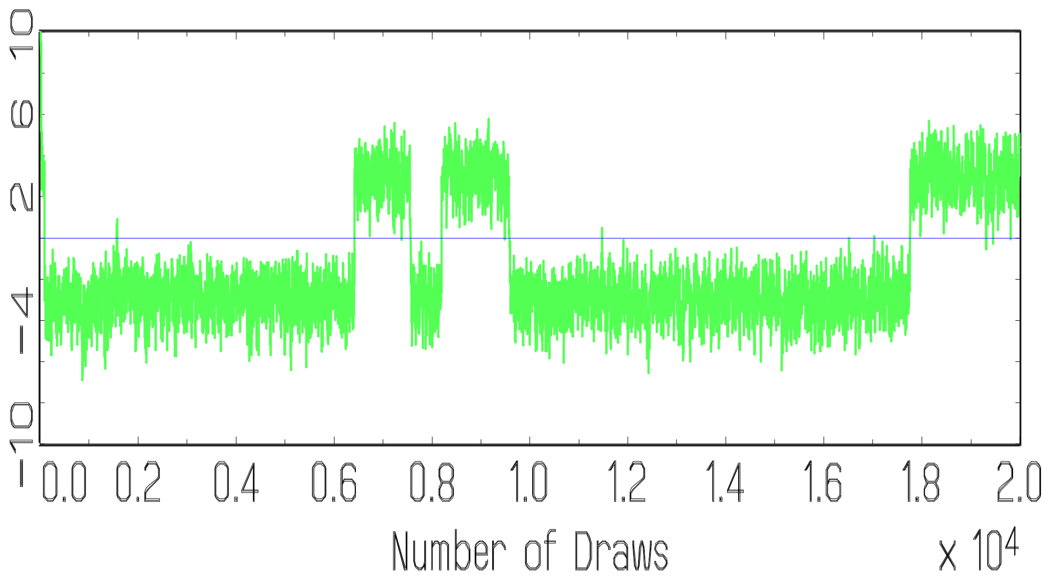
- Configuration of Algorithm
  - Starting values: $\theta_1^{(1)} = 10$, $\theta_2^{(1)} = 10$
  - Proposal density: $\mathcal{N}(\theta^{(s-1)}, 1^2 * \mathcal{I})$
  - Number of draws: 20000
- Rejection rate: 48.82 %.


Theta 1
Theta 2
-8
-4
0
2
4
6
8
-10
-4
0
4
8
-10
-20
-30
-40
-50



-50
-40
-30
-10
-20
-30
-40
-50
Theta 2
-10
-4
0
4
8
-8
-4
0
2
4
6
8
Theta 1

THETA1 Draws




THETA2 Draws