SAMSI Astrostatistics Tutorial

More Markov chain Monte Carlo
&
Demo of Mathematica software

Phil Gregory

University of British Columbia

2006

Bayesian Logical Data Analysis for the Physical Sciences

outline

Contents:

Outline

1. Introduce MCMC and parallel tempering 1
2. Technical difficulties 2
3. Return to simple spectral line problem 3
4. Tests for convergence 4
5. Exoplanet examples 5
6. Model comparison (global likelihoods) 6

In a Bayesian analysis we need to perform a lot of integration

outline

Example 1: to find the marginal posterior probability density for the orbital period, *P*, we need to integrate the joint posterior over all the other parameters.

$$p(P \mid D, M_1, I) = \int dK\, dV\, d\chi\, de\, d\omega \;\; p(P, K, V, \chi, e, \omega \mid D, M_1, I)$$

Example 2: to find the posterior probability of a model we need to evaluate the global likelihood of the model $p(D \mid M_1, I)$. This requires an integral over all the model parameters.

$$p(D \mid M_1, I) = \int dP\, dK\, dV\, d\chi\, de\, d\omega \; p(P, K, V, \chi, e, \omega \mid M_1, I)$$

$$p(D \mid M_1, P, K, V, \chi, e, \omega, I)$$

Markov chain Monte Carlo (MCMC) algorithms provide a powerful means for efficiently computing integrals in many dimensions.

MCMC to the rescue

In straight Monte Carlo integration independent samples are randomly drawn from the volume of the parameter space. The price that is paid for independent samples is that too much time is wasted sampling regions where posterior probability density is very small.

Suppose in a one-parameter problem the fraction of the time spent sampling regions of high probability is 10^{-1}. Then in an M-parameter problem, this fraction could easily fall to 10^{-M}.

MCMC algorithms avoid the requirement for completely independent samples, by constructing a kind of random walk in the model parameter space such that the number of samples in a particular region of this space is proportional to the posterior density for that region.

The random walk is accomplished using a Markov chain, whereby the new sample, $X_{\{t+1\}}$, depends on previous sample X_t according to an entity called the transition probability or transition kernel, $p(X_{\{t+1\}} | X_t)$. The transition kernel is assumed to be time independent.

The remarkable property of $p(X_{\{t+1\}} | X_t)$ is that after an initial burn-in period (which is discarded) it generates samples of X with a probability density equal to the desired posterior $p(X|D,I)$.

A simple Metropolis-Hastings MCMC algorithm

outline

P(X|D,M,I) = target posterior probability distribution
(X represents the set of model parameters)

1. Choose X_0 an initial location in the parameter space. Set $t = 0$.

2. Repeat {

– Obtain a new sample Y from a proposal distribution $q(Y \mid X_t)$ that is easy to evaluate. $q(Y \mid X_t)$ can have almost any form.

I use a Gaussian proposal distribution. i.e., Normal distribution $N(X_t, \sigma)$

– Sample a Uniform (0, 1) random variable U.

–If $U \leq \frac{p(Y \mid D, I)}{p(X_t \mid D, I)} \times \frac{q(X_t \mid Y)}{q(Y \mid X_t)}$ then set $X_{t+1} = Y$

otherwise set $X_{t+1} = X_t$

– Increment t }

This factor =1
for a symmetric proposal
distribution like a Gaussian

Conditions for convergence

Remarkably, for a wide range of proposal distributions $q(Y|X)$, the Metropolis-Hastings algorithm generates samples of X with a probability density which converges on the desired target posterior $p(X|D, I)$, called the *stationary distribution* of the Markov chain.

For the distribution of to converge to a stationary distribution, the Markov chain must have three properties (Roberts, 1996).

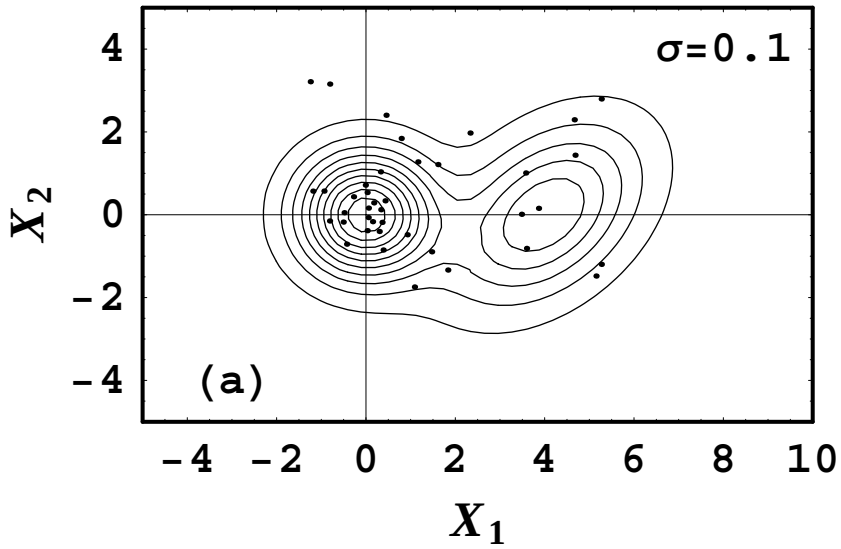
1. It must be *irreducible*. That is, from all starting points, the Markov chain must be able (eventually) to jump to all states in the target distribution with positive probability.

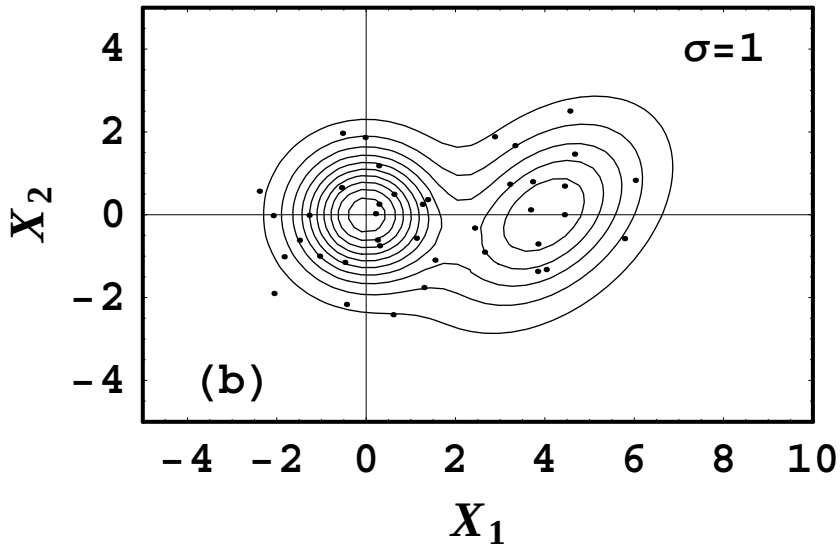
2. It must be *aperiodic*. This stops the chain from oscillating between different states in a regular periodic movement.

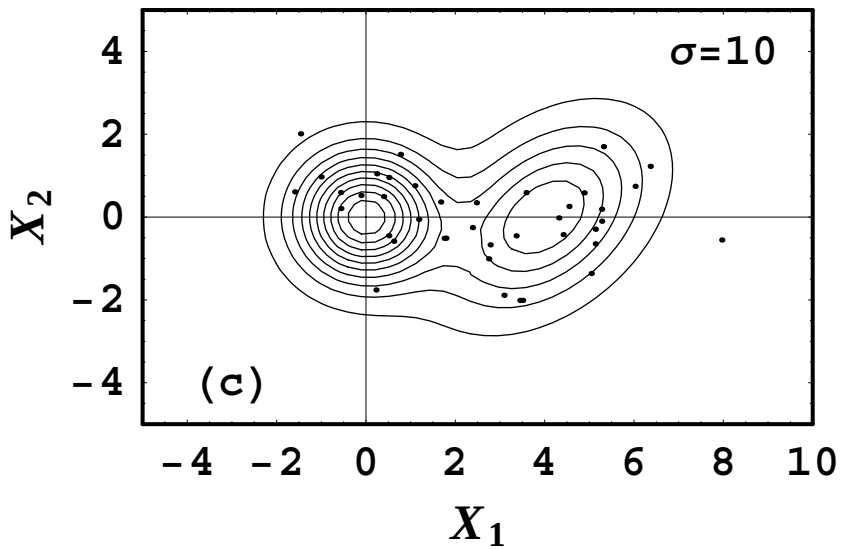
3. It must be *reversible*.

$$p\left(\vec{X} \mid D, I\right) p\left(\vec{X} \mid \vec{X}'\right) = p\left(\vec{X}' \mid D, I\right) p\left(\vec{X}' \mid \vec{X}\right)$$

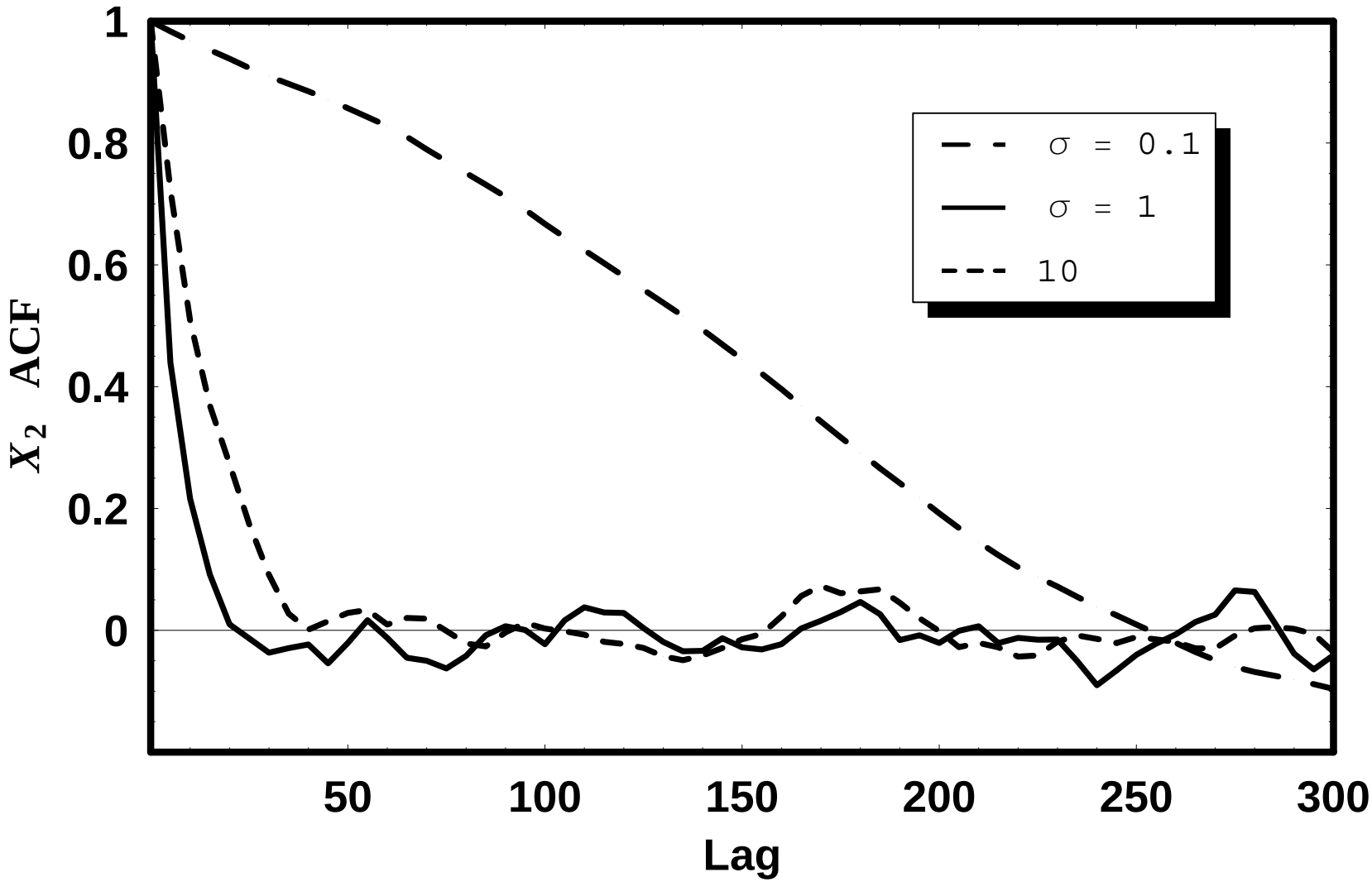
A comparison of the samples from three Markov Chain Monte Carlo runs using Gaussian proposal distributions with differing values of the standard deviation.







In this example the posterior probability distribution consists of two 2 dimensional Gaussians



A comparison of the autocorrelation functions for three Markov Chain Monte Carlo runs using Gaussian proposal distributions with differing values of the standard deviation.

Parallel Tempering

outline

The simple Metropolis-Hastings MCMC algorithm can run into difficulties if the probability distribution is multi-modal with widely separated peaks. It can fail to fully explore all peaks which contain significant probability, especially if some of the peaks are very narrow.

One solution is to run multiple Metropolis-Hastings simulations in parallel, employing probability distributions of the kind

$$\pi(X|D, M, \beta, I) = p(X|M, I)\ p(D|X, M, I)^{\beta} \quad (0 < \beta \leqslant 1)$$

β = 1 corresponds to our desired target distribution. The others correspond to progressively flatter probability distributions.

I learned about parallel tempering from this book.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, Springer Series in Statistics, Springer NY.

Parallel Tempering

At intervals, a pair of adjacent simulations are chosen at random and a proposal made to swap their parameter states. The update can be accepted/rejected with a Metropolis-Hastings criterion. At time *t*, simulation β_i is in state $X_{t,i}$ and simulation β_{i+1} is in state $X_{t,i+1}$. If the swap is accepted by the test given below then these states are interchanged. Accept the swap with probability

$$r = \min\left\{1, \frac{\pi(X_{t,i+1}|D,\beta_i,I)\ \pi(X_{t,i}|D,\beta_{i+1},I)}{\pi(X_{t,i}|D,\beta_i,I)\ \pi(X_{t,i+1}|D,\beta_{i+1},I)}\right\}$$

The swap allows for an exchange of information across the ladder of simulations.

In the low β simulations, radically different configurations can arise, whereas at higher β, a configuration is given the chance to refine itself.

Final results are based on samples from the β = 1 simulation. Samples from the other simulations can be used to evaluate the Bayes Factor in model selection problems.

Technical Difficulties

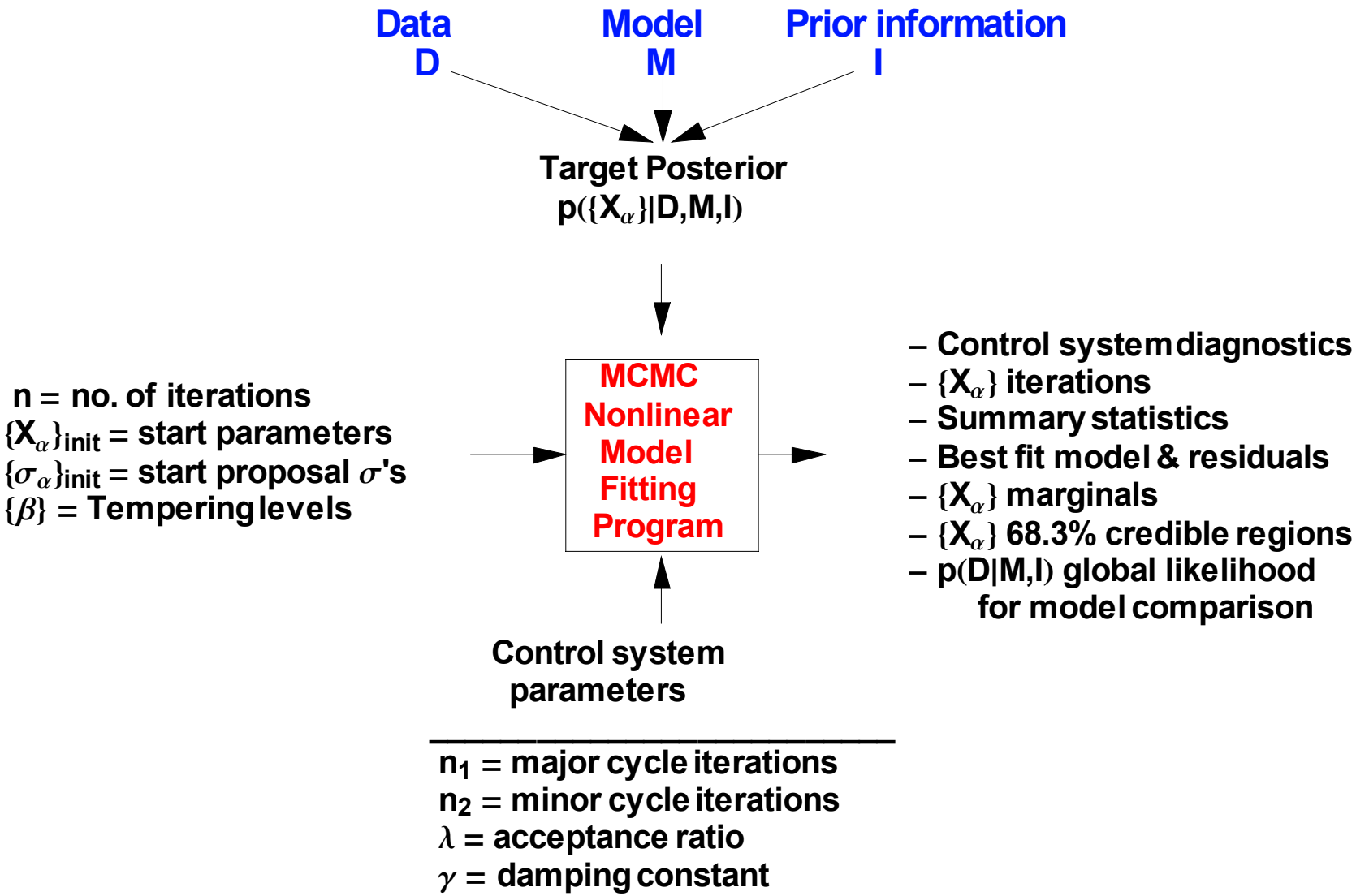
1. Deciding on the burn-in period.

2. Choosing a good choice for the characteristic width of each proposal distribution, one for each model parameter.

   For Gaussian proposal distributions this means picking a set of proposal σ's. This can be very time consuming for a large number of different parameters.

3. Deciding how many iterations are sufficient.

4. Deciding on a good choice of tempering levels (β values).

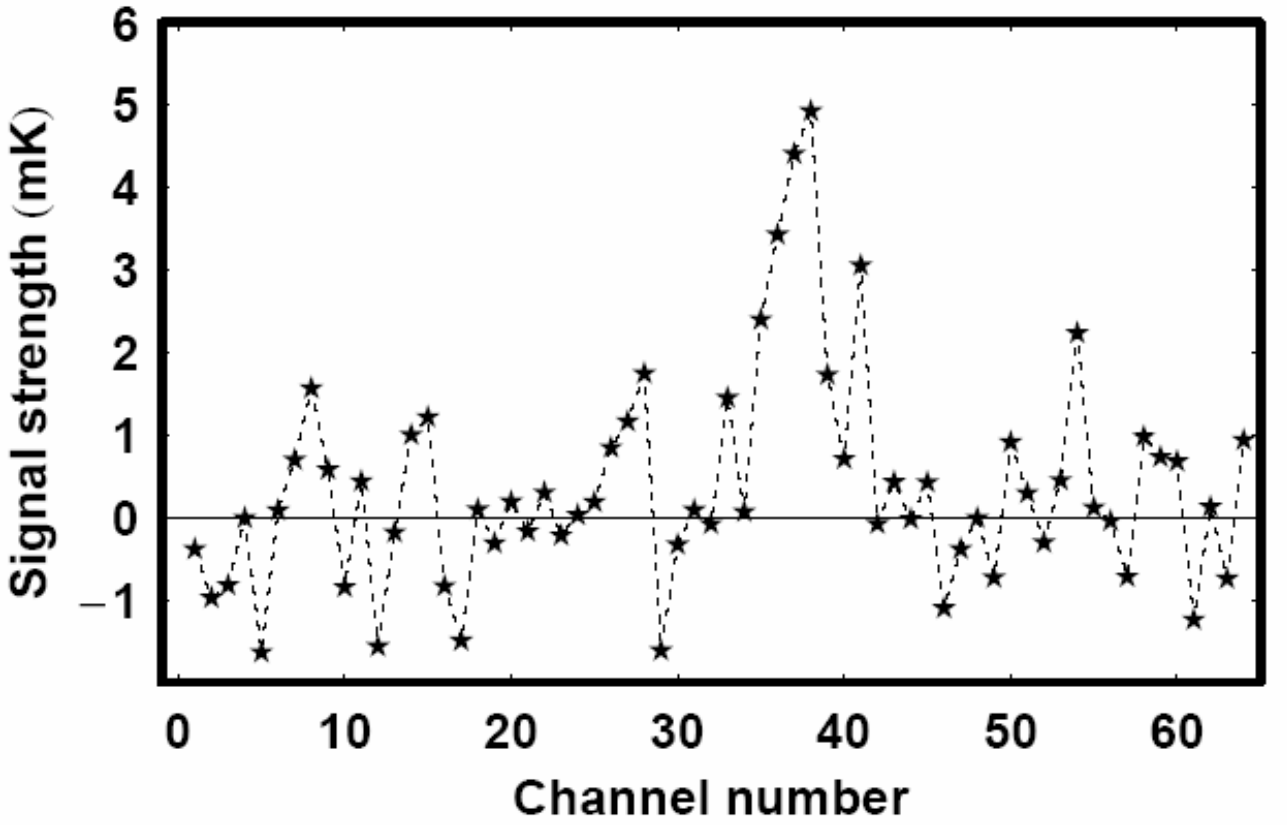


Schematic of a Bayesian Markov Chain Monte Carlo program for nonlinear model fitting. The program incorporates a control system that automates the selection of Gaussian proposal distributon σ's.

Return to simple spectral line problem

Now assume 4 unknowns line center frequency, T, line width, and an extra noise term with unknown standard deviation

Parameters f, T, lw, s



All channels have Gaussian noise characterized by $\sigma = 1$ mK. The noise in separate channels is independent. The line center frequency $\nu_0 = 37$.

Tests for convergence

1. Examine plots of the MCMC iterations for each parameter.

2. Divide the post burn-in iterations into two halves and plot one on top of the other using different colors.

3. Compute the Gelman-rubin statistic for each parameter from the repeated runs.

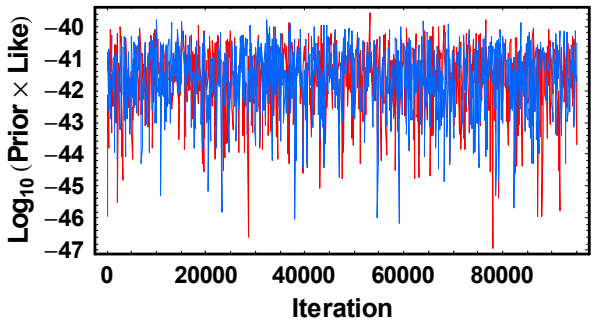
Tests for convergence

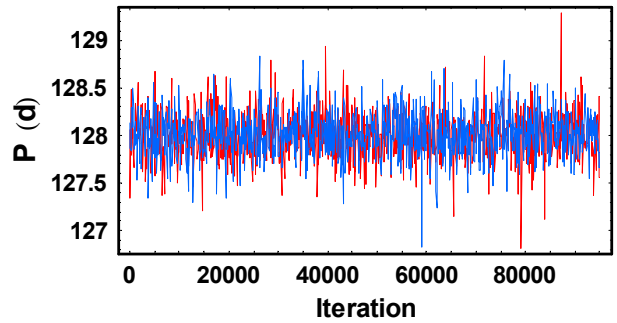
Compare iterations from multiple chains

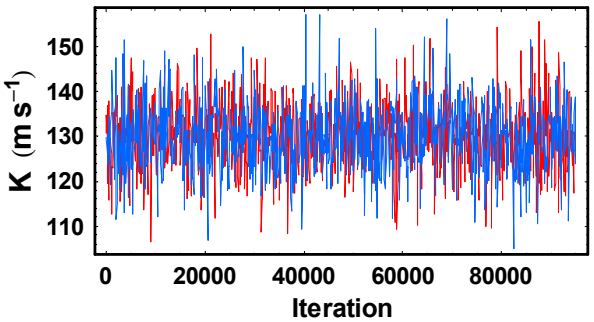
For each parameter compute Gelman-Rubin statistic

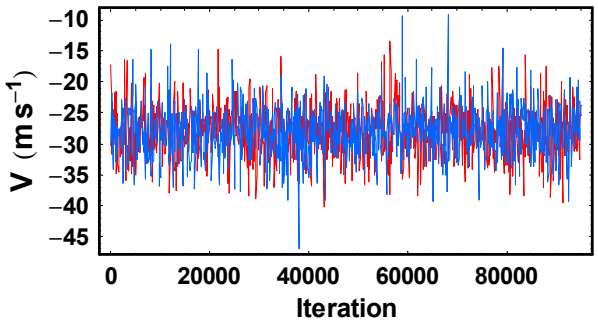
GR should be close to 1.0

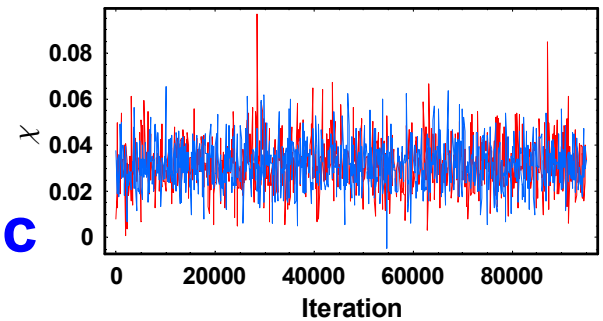
Measured GR values < 1.02

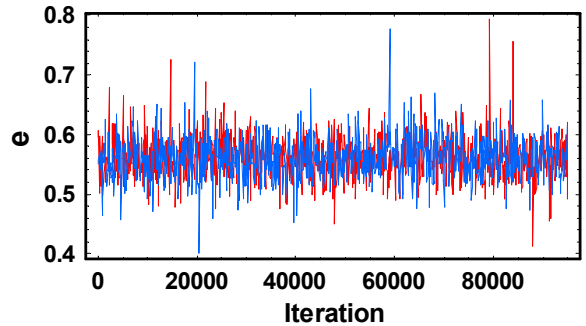


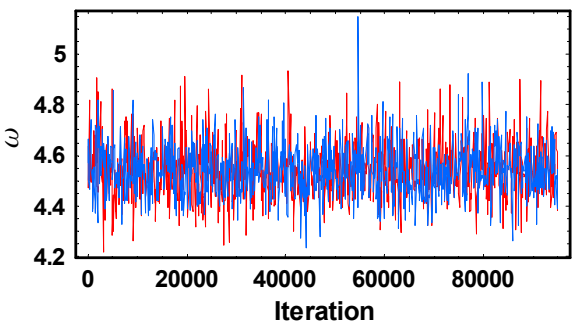


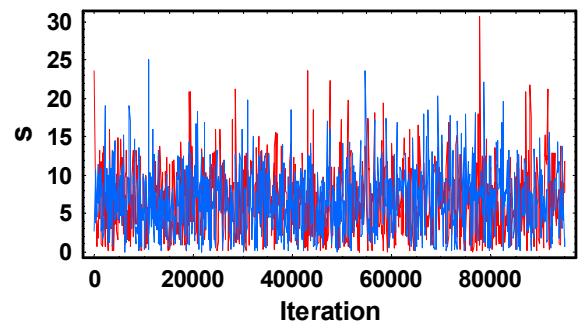












Gelman-Rubin Statistic

Let θ represent one of the model parameters.

Let θ_j^i represent the i^{th} iteration of the j^{th} chain.

Extract the last η post burn-in iterations for each simulation.

$$\text{Mean within chain variance } W = \frac{1}{m(\eta-1)} \sum_{j=1}^{m} \sum_{i=1}^{\eta} \left(\theta_j^i - \bar{\theta}_j\right)^2$$

$$\text{Between chain variance } B = \frac{\eta}{m-1} \sum_{j=1}^{m} (\bar{\theta}_j - \bar{\theta})^2$$

$$\text{Estimated variance } \hat{V}(\theta) = \left(1 - \frac{1}{\eta}\right) W + \frac{1}{\eta} B$$

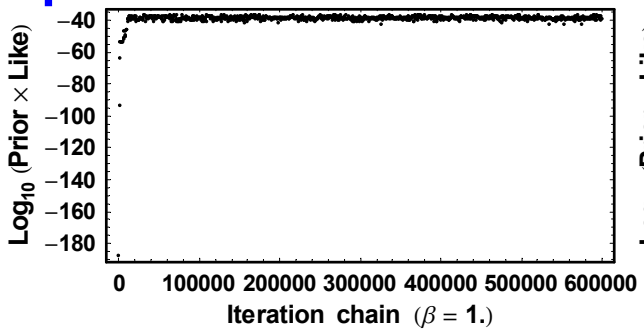
$$\text{Gelman} - \text{Rubin statistic } = \sqrt{\frac{\hat{V}(\theta)}{W}}$$

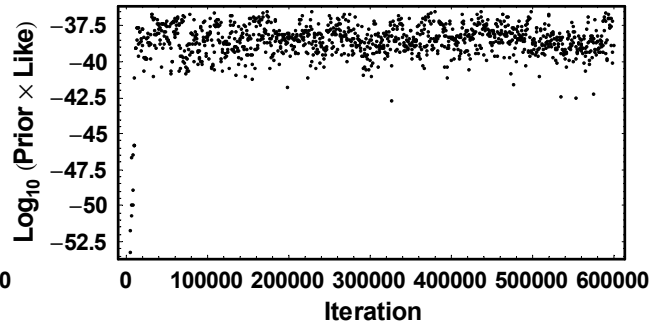
The Gelman – Rubin statistic should be close to 1.0 (e.g. < 1.05) for all paramaters for convergence

Ref: Gelman, A. and D.B. Rubin (1992) 'Inference from iterative simulations using multiple sequences (with discussion)', Statistical Science 7, pp. 457 – 511.

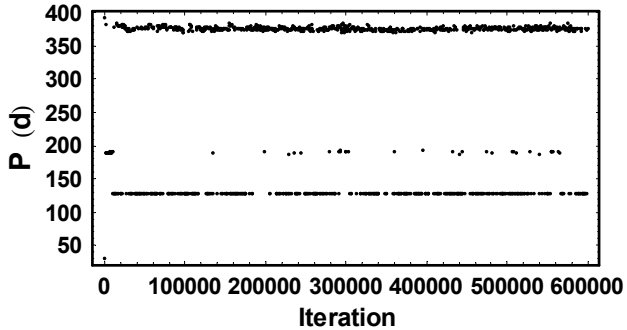
HD 73526 MCMC orbital parameter iterations

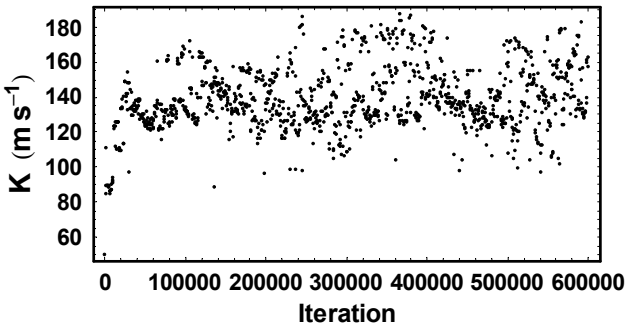
outline

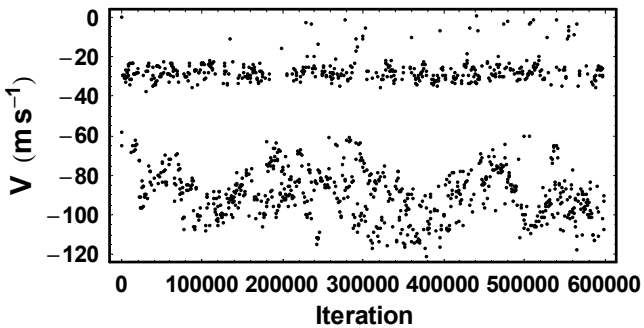


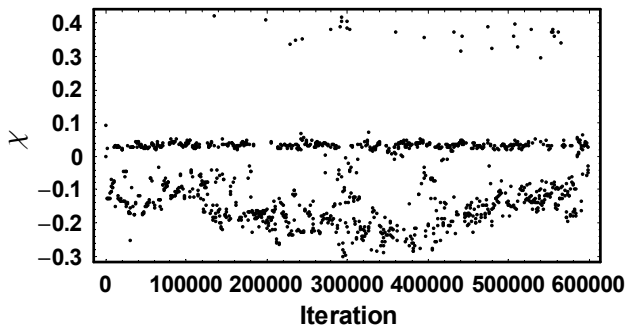


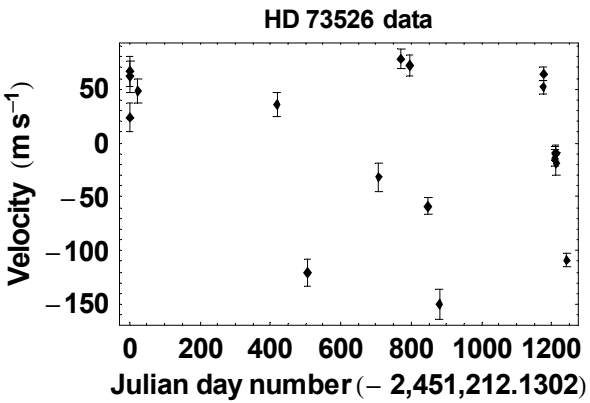
Evidence for 3 periods

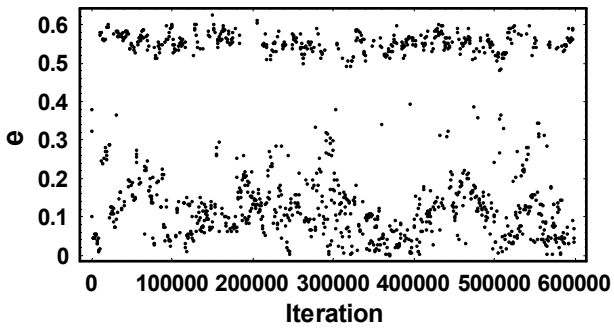


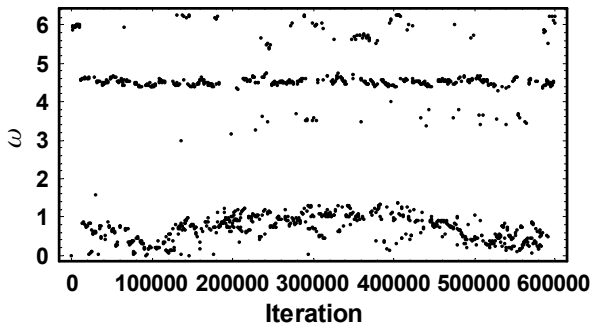








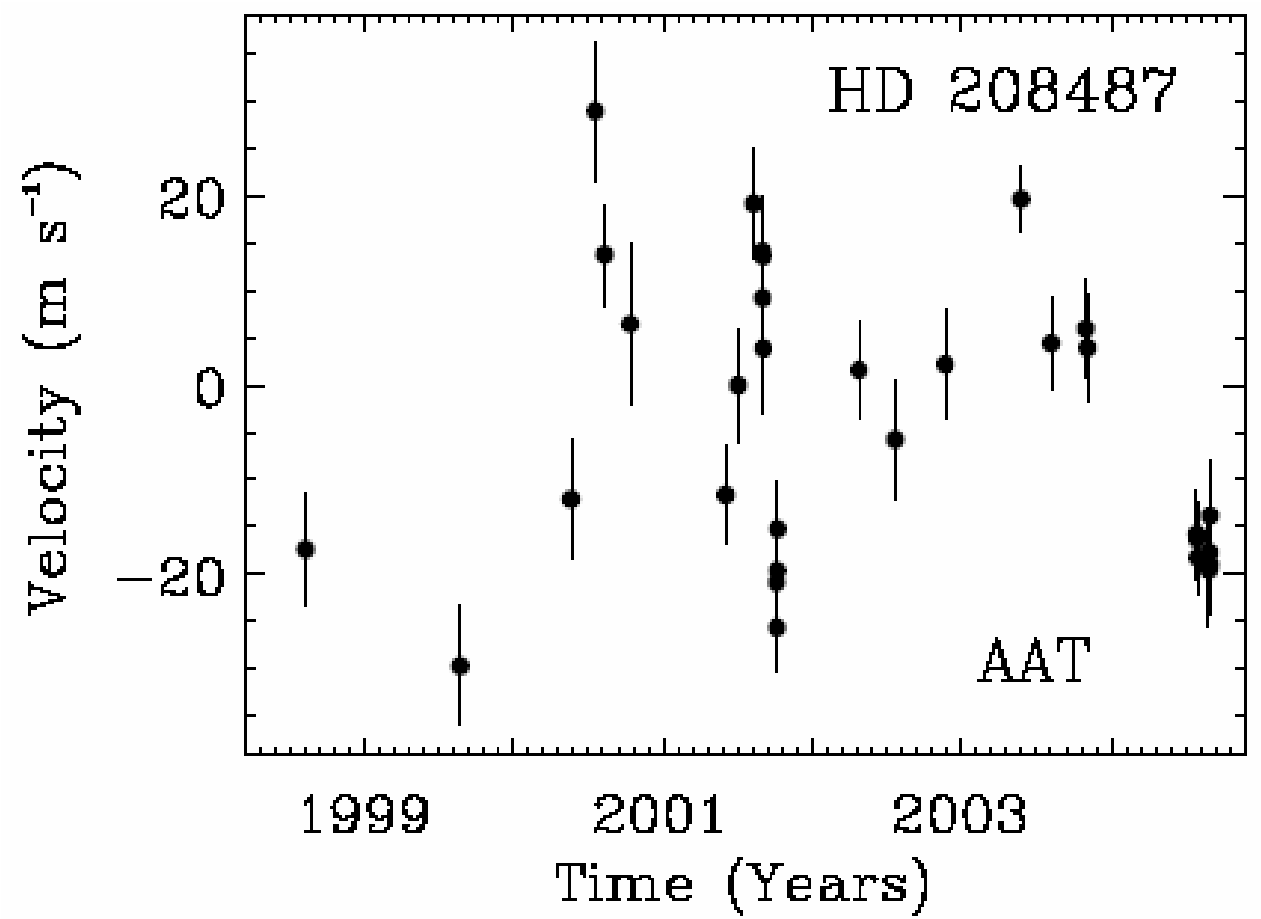




THREE LOW-MASS PLANETS FROM THE ANGLO-AUSTRALIAN PLANET SEARCH[1]

C. G. Tinney,[2] R. Paul Butler,[3] Geoffrey W. Marcy,[4,5] Hugh R. A. Jones,[6,7] Alan J. Penny,[8,9]
Chris McCarthy,[3,5] Brad D. Carter,[10] and Debra A. Fischer[4,5]

Data



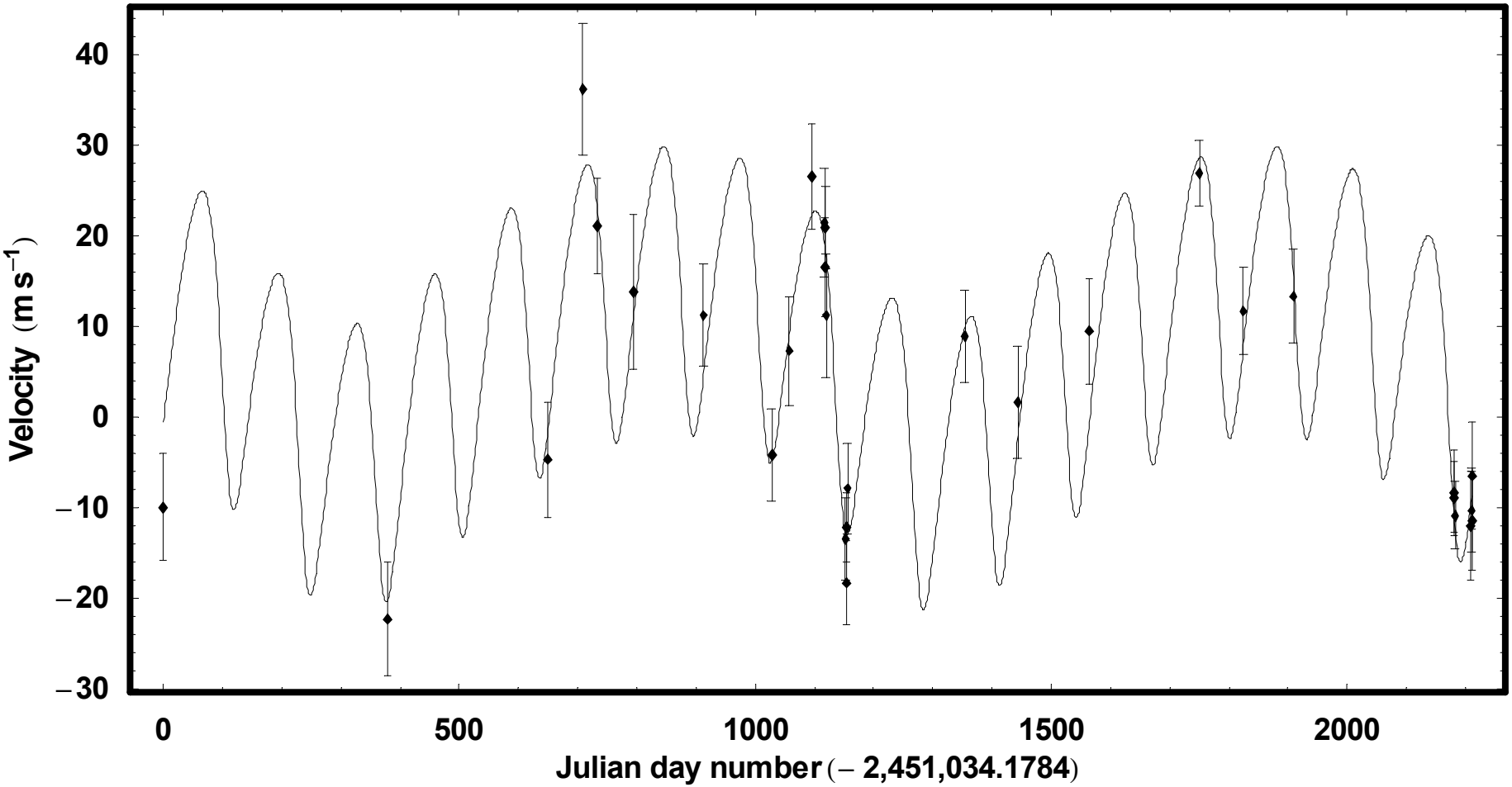
The Astrophysical Journal, 623:1171–1179, 2005 April 20

Evidence for a second planet in HD 208487

outline

Gregory, P. C. (2005), AIP Conference Proceeding 803, p. 139, 2005

Best fit $P_1 = 129.517$; $K_1 = 15.9778$; $V = 7.92451$; $\chi_1 = 0.184585$; $e_1 = 0.224581$; $\omega_1 = 2.20511$;
$P_2 = 998.085$; $K_2 = 9.81886$; $\chi_2 = 0.723248$; $e_2 = 0.189448$; $\omega_2 = 2.74554$



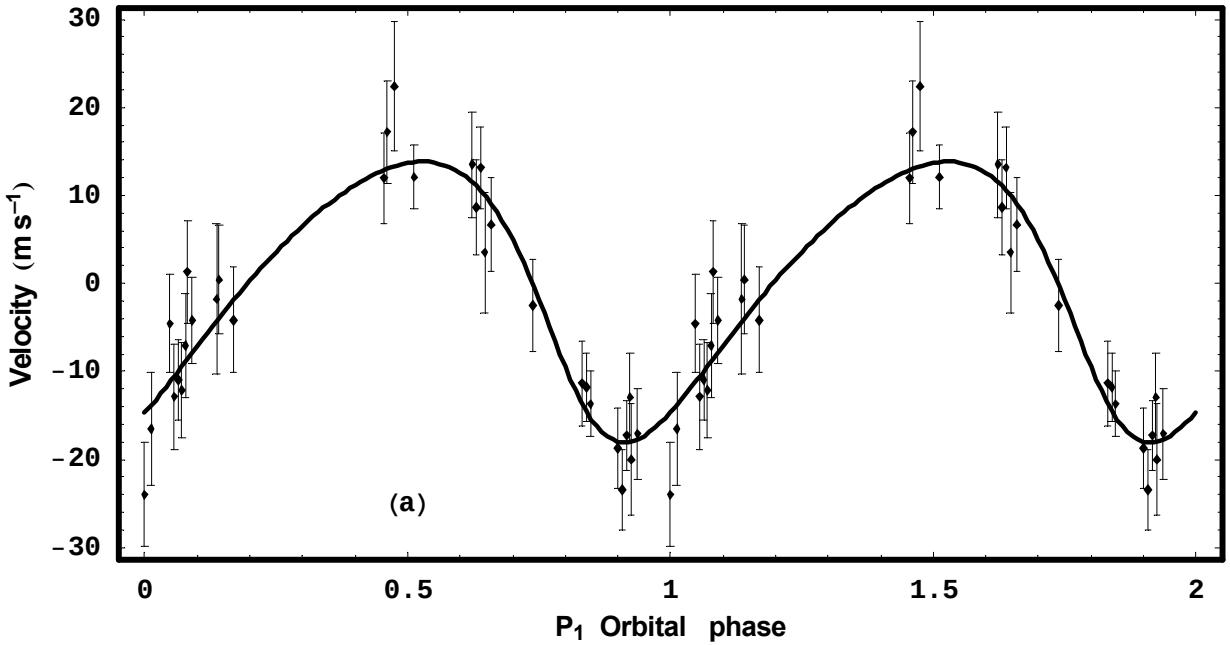
RMS residual = 4.2 m s^{-1}

$\chi^2_\upsilon = 0.83$

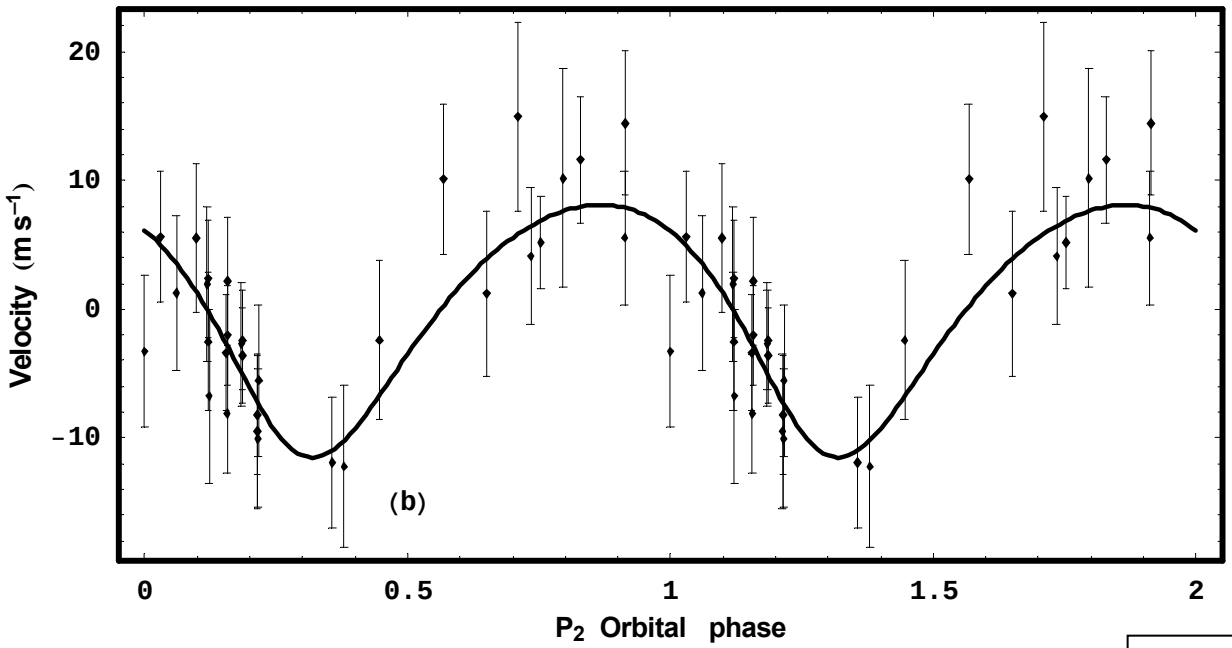
priors

HD 208487

P_1 = 129.52 d
e_1 = 0.22



P_2 = 998 d
= 2.73 yr
e_2 = 0.19



Model compare

Model selection

To answer the model selection question, we compute the odds ratio (abbreviated simply by the *odds*) of model M_1 to model M_2 .

Expand numerator and denominator with Bayes' theorem

$$O_{12} = \frac{p(M_1 \mid D, I)}{p(M_2 \mid D, I)} = \frac{\dfrac{p(M_1 \mid I)\, p(D \mid M_1, I)}{p(D \mid I)}}{\dfrac{p(M_2 \mid I)\, p(D \mid M_2, I)}{p(D \mid I)}} = \frac{p(M_1 \mid I)}{p(M_2 \mid I)} \frac{p(D \mid M_1, I)}{p(D \mid M_2, I)}$$

posterior probability ratio

prior probability ratio

Bayes factor

$p(D|M_1, I)$, the called the global likelihood of M_1 .

$$p(D \mid M_1, I) = \int_T p(D, T \mid M_1, I)\, dT$$

Expanded with product rule

$$= \int_T p(T \mid M_1, I)\; p(D \mid M_1, T, I)\, dT$$

The global likelihood of a model is equal to the weighted average likelihood for its parameters.

Model selection

One way to compute the global likelihood of the mode, $\log[p(D \mid M1, I)]$.

$$\log[p(D \mid M1, I)] = \int \langle \ln p(D \mid M_E, P, K, V, \chi, e, \omega, I) \rangle_\beta \, d\beta$$

See my book for derivation

$\langle \ln p(D \mid M_E, P, K, V, \chi, e, \omega, I) \rangle_\beta$

= expectation value of $\ln [p(D \mid M_E, P, K, V, \chi, e, \omega, I)]$ for given β

$$= \frac{1}{n} \sum_{t=1}^{n} \ln[p(D \mid M1, P_{t,\beta}, K_{t,\beta}, V_{t,\beta}, \chi_{t,\beta}, e_{t,\beta}, \omega_{t,\beta}, I)]$$

where n = number of MCMC iterations.

One way to compute the global likelihood of the mode, $\log[p(D \mid M1, I)]$.

$$\log[p(D \mid M1, I)] = \int \langle \ln p(D \mid M_E, P, K, V, \chi, e, \omega, I) \rangle_\beta \, d\beta$$

See my book for derivation

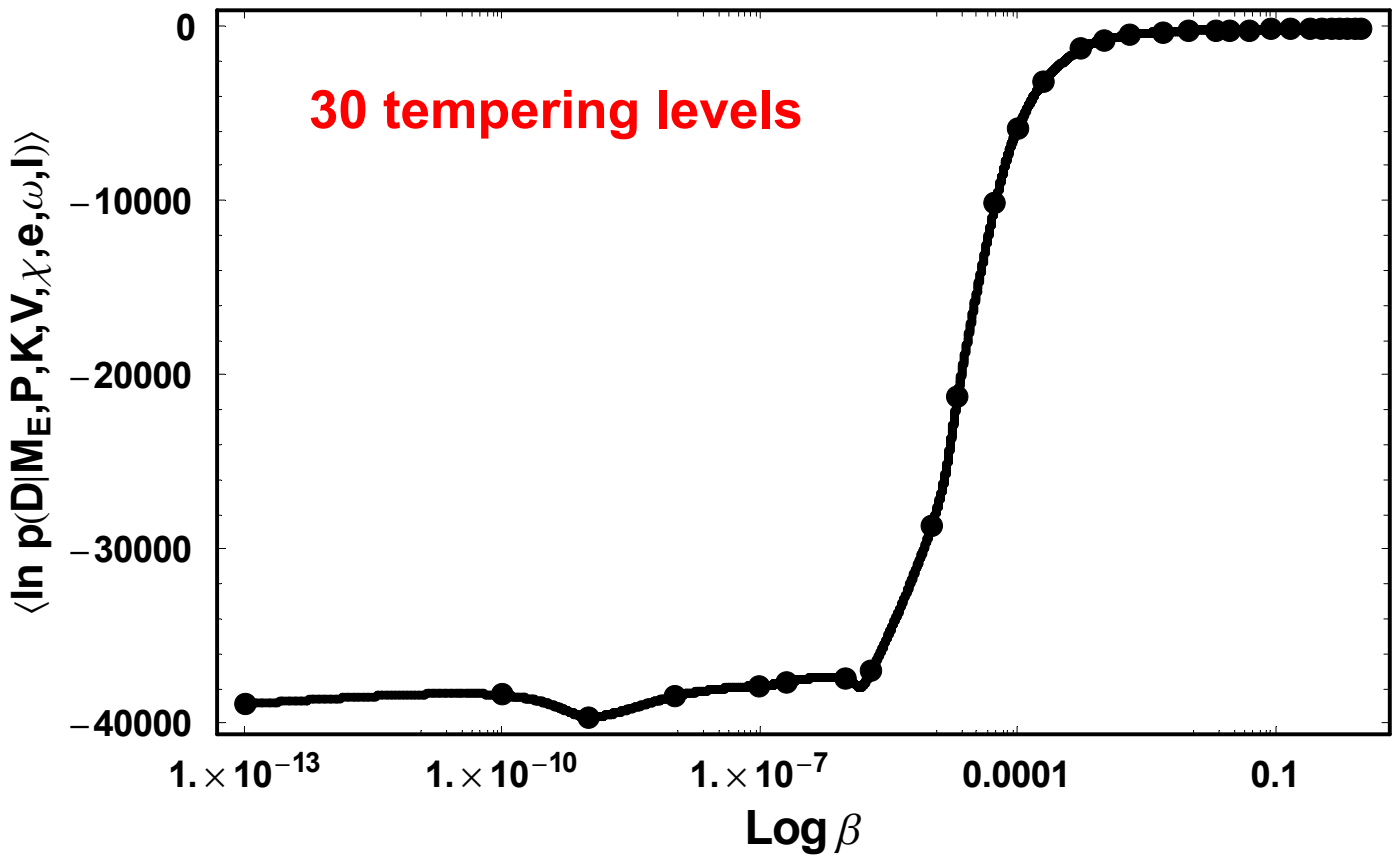
outline

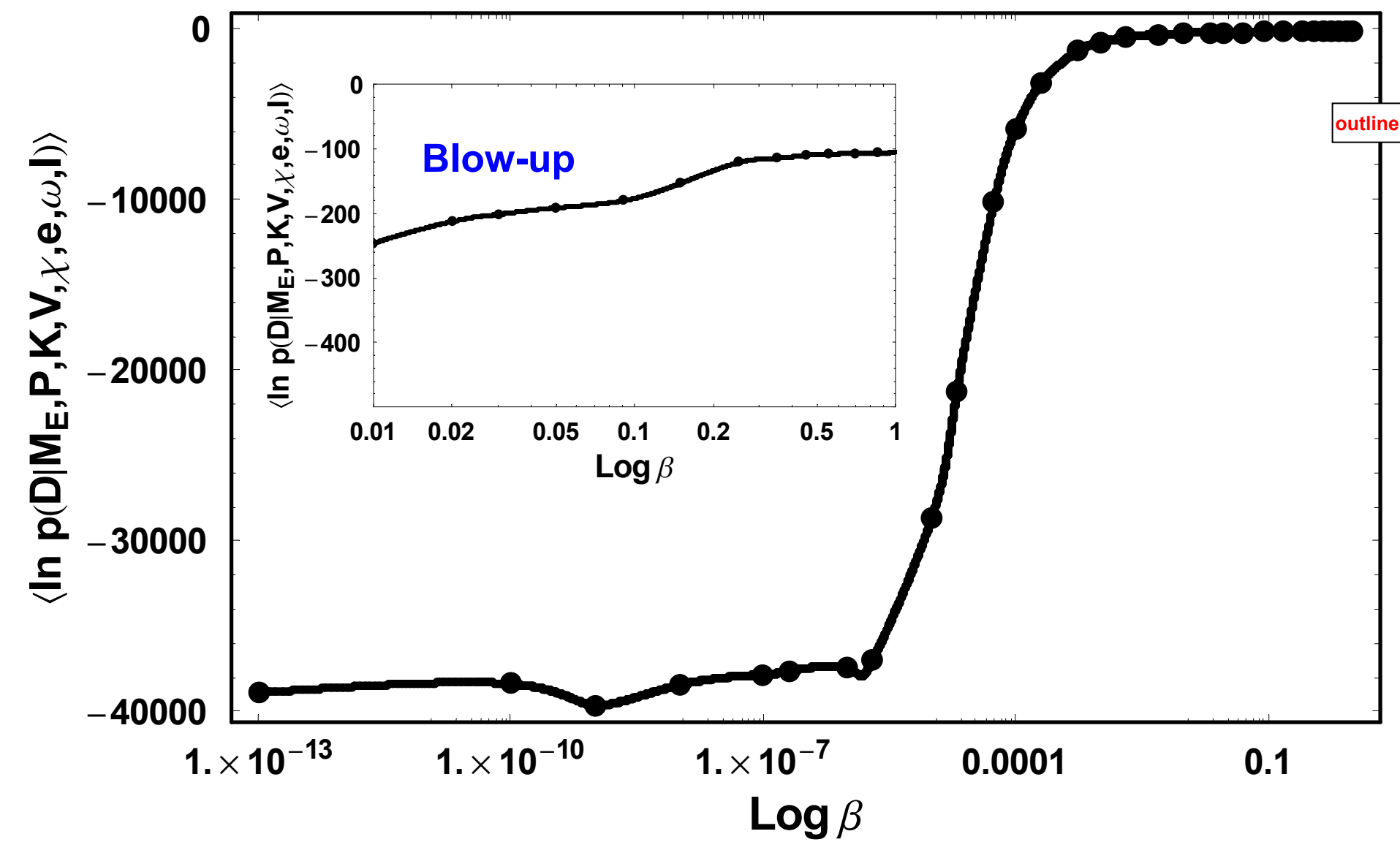
$\langle \ln p(D \mid M_E, P, K, V, \chi, e, \omega, I) \rangle_\beta$

= expectation value of $\ln [p(D \mid M_E, P, K, V, \chi, e, \omega, I)]$ for given β

$$= \frac{1}{n} \sum_{t=1}^{n} \ln[p(D \mid M1, P_{t,\beta}, K_{t,\beta}, V_{t,\beta}, \chi_{t,\beta}, e_{t,\beta}, \omega_{t,\beta}, I)]$$

where n = number of MCMC iterations.





$p(M_{1J}|D,I) = 1.4 \times 10^{-55}$ (from tempering levels)

$p(M_{1J}|D,I) = 2.5 \times 10^{-55}$ (from restricted Monte Carlo integration)

$p(M_{0s}|D,I) = 1.5 \times 10^{-59}$ (from numerical integration)