

NPL REPORT MS 20

**MCM2MCMC USER MANUAL: SOFTWARE IMPLEMENTING A
MARKOV CHAIN MONTE CARLO ALGORITHM FOR UNCERTAINTY
EVALUATION BASED ON THE MONTE CARLO METHOD OF
SUPPLEMENT 1 TO THE GUM**

A B FORBES, K JAGAN AND I M SMITH

APRIL 2018

MCM2MCMC User Manual: Software implementing a Markov chain Monte Carlo algorithm for uncertainty evaluation based on the Monte Carlo method of Supplement 1 to the GUM

A B Forbes, K Jagan and I M Smith
Data science Group

April 2018

ABSTRACT

This report constitutes a *user manual* for software developed at the National Physical Laboratory to convert a sample from a Bayesian posterior distribution corresponding to a particular choice of prior distribution derived using the Monte Carlo method, to a Bayesian posterior corresponding to a preferred prior distribution. A Metropolis-Hastings Markov chain Monte Carlo algorithm is used to achieve this conversion.

NPL Report MS 20

© NPL Management Limited, 2018

ISSN 1754–2960

National Physical Laboratory,
Hampton Road, Teddington, Middlesex, United Kingdom TW11 0LW

Extracts from this report may be reproduced provided the source is acknowledged and the
extract is not taken out of context

We gratefully acknowledge the financial support of the UK Department for Business,
Energy & Industrial Strategy

Approved on behalf of NPLML by Louise Wright,
Science Area Leader for Modelling

Contents

1	Introduction	1
1.1	Scope	1
1.2	Software user licence agreement	1
2	Measurement response model	1
3	Software implementation	3
3.1	Metropolis-Hastings independence chain algorithm	3
3.1.1	Determine selection indices	4
3.1.2	Apply selection indices	5
3.1.3	Determine convergence indices	6
3.1.4	Calculate sample summary information	7
3.2	MHIC applied to uncertainty evaluation	8
4	Numerical examples	9
4.1	Gauge block examples	9
4.1.1	Example 1: Likelihood defined by a Gaussian distribution	11
4.1.2	Example 2: Likelihood defined by a Student's t-distribution	14
4.1.3	Example 3: Likelihood specified by a symmetric Beta distribution	16
4.2	Exponential examples	18
4.2.1	Example 1: Univariate problem	18
4.2.2	Example 2: Bivariate problem	20

1 Introduction

1.1 Scope

This document describes MATLAB software implementing the second stage of a two-stage algorithm for uncertainty evaluation described in [6, 7]. The first stage uses the Monte Carlo method (MCM) [2, 3, 4] to produce a sample from a probability distribution that can be regarded as a Bayesian posterior distribution corresponding to a particular choice of prior distribution imposed by the computational approach. The second stage uses a Metropolis-Hastings Markov chain Monte Carlo (MCMC) algorithm to convert the MCM sample to a sample from the Bayesian posterior distribution corresponding to a preferred prior distribution. Section 2 describes the model underlying the algorithm, Section 3 describes the individual software components and numerical examples are given in Section 4.

1.2 Software user licence agreement

The software is provided with a software user licence agreement and the use of the software is subject to the terms laid out in that agreement. By running the software, the user accepts the terms of the agreement.

2 Measurement response model

The software is concerned with the case in which the responses $\boldsymbol{\eta} = (\eta_1, \dots, \eta_k)^\top$ of an instrument are modelled as functions of influence quantities $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)^\top$ and quantities of interest $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)^\top$.

The instrument response η_i is modelled as

$$\eta_i = \phi_i(\boldsymbol{\alpha}, \boldsymbol{\beta}), \quad i = 1, \dots, k,$$

so that

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_k \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ \vdots \\ \phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) \end{bmatrix} = \boldsymbol{\phi}(\boldsymbol{\alpha}, \boldsymbol{\beta}).$$

It is assumed that the response model can be re-expressed as

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} \psi_1(\boldsymbol{\eta}, \boldsymbol{\beta}) \\ \vdots \\ \psi_k(\boldsymbol{\eta}, \boldsymbol{\beta}) \end{bmatrix} = \boldsymbol{\psi}(\boldsymbol{\eta}, \boldsymbol{\beta}), \quad (1)$$

expressing α as an explicit function of η and β .

A joint probability density $p(\eta, \beta)$ for η and β may be propagated through model (1) to $p(\alpha)$ for α using a Monte Carlo method. If $\{(\mathbf{y}_q, \mathbf{b}_q), q = 1, \dots, M\}$ is a sample drawn from $p(\eta, \beta)$, then $\{\mathbf{a}_q = \psi(\mathbf{y}_q, \mathbf{b}_q), q = 1, \dots, M\}$ is a sample from $p(\alpha)$.

Supplement 1 (GUMS1, [2]) and Supplement 2 (GUMS2, [3]) to the *Guide to the expression of uncertainty in measurement* (GUM, [1]) employ the Monte Carlo method in the following context. It is assumed that prior knowledge about the influence quantities is encoded in a prior density $p(\beta)$. Measurement of the response of the instrument produces indications ζ with associated likelihood $p(\zeta|\eta) \equiv p(\zeta|\alpha, \beta)$, e.g.,

$$\zeta|\eta \sim N(\eta, \sigma^2 I).$$

On the basis of ζ , η is assigned a density $p(\eta|\zeta) \propto p(\zeta|\eta)$, i.e., the Bayesian posterior distribution corresponding to a non-informative prior $p(\eta) \propto 1$ for η . For this example, the form of the density $p(\eta|\zeta)$ is

$$p(\eta|\zeta) \propto \frac{1}{\sigma} \exp \left\{ -\frac{1}{2\sigma^2} (\zeta - \eta)^\top (\zeta - \eta) \right\}.$$

The joint density $p(\eta, \beta|\zeta)$ is assigned to be

$$p(\eta, \beta|\zeta) = p(\eta|\zeta)p(\beta),$$

and independent samples $\{\mathbf{y}_q, q = 1, \dots, M\}$ and $\{\mathbf{b}_q, q = 1, \dots, M\}$ are drawn from $p(\eta|\zeta)$ and $p(\beta)$, respectively. It can be shown [5, 8] that $\{(\mathbf{a}_q, \mathbf{b}_q), q = 1, \dots, M\}$, with $\mathbf{a}_q = \psi(\mathbf{y}_q, \mathbf{b}_q)$, are draws from the Bayesian posterior distribution

$$p_{\text{MCM}}(\alpha, \beta|\zeta) \propto p(\zeta|\alpha, \beta)|J(\alpha, \beta)|p(\beta),$$

where $|J(\alpha, \beta)|$ is the absolute value of the determinant of the Jacobian matrix $J = J(\alpha, \beta)$ with

$$J_{ij} = \frac{\partial \phi_i}{\partial \alpha_j}(\alpha, \beta).$$

Thus, the MCM approach provides a sample from a Bayesian posterior distribution corresponding to the prior

$$p(\alpha, \beta) \propto |J(\alpha, \beta)|p(\beta).$$

If the preferred prior is $p(\alpha, \beta) = p_{00}(\alpha)p(\beta)$, leading to posterior distribution

$$p(\alpha, \beta|\zeta) \propto p(\zeta|\alpha, \beta)p_{00}(\alpha)p(\beta),$$

then

$$\frac{p(\alpha, \beta|\zeta)}{p_{\text{MCM}}(\alpha, \beta|\zeta)} \propto \frac{p_{00}(\alpha)}{|J(\alpha, \beta)|}, \quad (2)$$

involving only $p_{00}(\alpha)$ and $|J(\alpha, \beta)|$. For the case of a univariate response model,

$$\eta = \phi(\alpha, \beta),$$

α and J are scalar quantities, where

$$J = \frac{\partial \phi}{\partial \alpha}(\alpha, \beta).$$

The relationship (2) leads to a particularly simple implementation of the Metropolis-Hastings independence chain algorithm [9, 10] in which draws from the approximating distribution $p_{\text{MCM}}(\alpha, \beta|\zeta)$ are converted to draws from $p(\alpha, \beta|\zeta)$.

3 Software implementation

3.1 Metropolis-Hastings independence chain algorithm

The Metropolis-Hastings independence chain (MHIC) algorithm determines a sample from a target distribution $p(\alpha)$ given a sample from an approximating distribution $p_0(\alpha)$. The algorithm assumes that $\{\mathbf{a}_{0,q,r}, q = 1, \dots, M, r = 1, \dots, N\}$ have been sampled independently from $p_0(\alpha)$ and that densities $p_0(\mathbf{a}_{0,q,r})$ and $p(\mathbf{a}_{0,q,r})$ have been evaluated, each up to a normalising constant. The algorithm requires that each chain starts at a feasible point, i.e., $p(\mathbf{a}_{0,1,r}) > 0$ for all r . Since the sample $\mathbf{a}_{0,q,r}$ has been drawn, necessarily $p_0(\mathbf{a}_{0,q,r}) > 0$. With this information, N chains of length M are determined, $\{\mathbf{a}_{q,r}, q = 1, \dots, M, r = 1, \dots, N\}$, accepting or rejecting the proposed $\mathbf{a}_{0,q,r}$, according to the Metropolis-Hastings scheme (see, e.g., [10]).

The MHIC algorithm can be applied in situations in which samples are required from a distribution $p(a)$ which has an approximating distribution $p_0(a)$ that is easy to sample from. The general procedure is outlined below.

Given a draw a_{q-1} , a proposed draw a^* for the next member of the sequence is sampled at random from the approximating distribution $p_0(a)$. Then a_q is set to a^* with acceptance probability

$$p_q = \min\{1, r_q\}, \quad r_q = \frac{p(a^*)p_0(a_{q-1})}{p(a_{q-1})p_0(a^*)}.$$

A number of iterations are allowed for the Markov chain to reach the target distribution. This number is known as the “burn-in period”. A burn-in period of length M_0 is assigned, with the intention that, for $q > M_0$, a_q is assumed to be a sample from the posterior distribution.

For the current problem, the MCM sample of α and β is considered to be a sample from the approximating distribution and the target distribution is the joint posterior distribution for α and β with a preferred prior for α .

The MHIC algorithm is implemented in two software components:

- `mhicI.m`: Use $p_0(\mathbf{a}_{0,q,r})$ and $p(\mathbf{a}_{0,q,r})$ to evaluate indices that determine which samples are accepted and which are rejected.
- `mhicia2a.m`: Use the indices to convert the sample $\mathbf{a}_{0,q,r}$ to a sample $\mathbf{a}_{q,r}$ from the posterior distribution, $q > M_0$.

Two further software components are also provided:

- `mcmcci.m`: Assess the convergence of an MCMC algorithm from multiple chains.
- `mcmcsun.m`: Provide summary information about the posterior distribution on the basis of the sample $\mathbf{a}_{q,r}$.

The first three software components apply to a single (scalar) quantity, whereas the last component applies to a general (vector) quantity.

3.1.1 Determine selection indices

The software component `mhicI.m` has calling syntax

```
[IS, IA] = mhicI(P, P0)
```

Given p and p_0 evaluated at an $M \times N$ array of samples $\mathbf{a}_{0,q,r}$, this component calculates the selection indices I_S used to convert the samples to Markov chains, and the acceptance indices I_A .

Name	Size	Description
Inputs		
P	$M \times N$	Array P storing the target density: $P_{q,r} = p(\mathbf{a}_{0,q,r})$. Constraint: $P_{1,r} > 0$ for all r .
P0	$M \times N$	Array P_0 storing the approximating density $P_{0,q,r} = p_0(\mathbf{a}_{0,q,r})$ Constraint: $P_{0,q,r} > 0$ for all q and for all r .
Outputs		
IS	$M \times N$	Array I_S storing selection indices: $I_{S,q,r} = q$ if $\mathbf{a}_{0,q,r}$ is accepted as a draw from $p(\boldsymbol{\alpha})$, and is $q - 1$ otherwise.
IA	$M \times N$	Array I_A storing acceptance indices: $I_{A,q,r} = 1$ if $\mathbf{a}_{0,q,r}$ is accepted as a draw from $p(\boldsymbol{\alpha})$, and is zero otherwise.

3.1.2 Apply selection indices

The software component `mhicia2a.m` has calling syntax

```
[A] = mhicia2a(A0, IS)
```

This component uses the selection indices I_S to convert the samples contained in A_0 to Markov chains.

Name	Size	Description
		Inputs
A0	$M \times N$	Array A_0 storing samples from the proposal distribution.
IS	$M \times N$	Array I_S storing selection indices returned by <code>mhicI.m</code> .
		Output
A	$M \times N$	Array A storing N chains of length M .

3.1.3 Determine convergence indices

The software component `mcmcci.m` has calling syntax

$$[\text{Rhat}, \text{Neff}, \text{Abar}, \text{stdA}] = \text{mcmcci}(\text{A}, \text{M0})$$

This component may be applied to the outputs of any MCMC scheme involving multiple chains. It evaluates convergence indices that indicate if the chains have converged to the target distribution based on the scheme described in [10, Section 11.4].

Name	Size	Description
Inputs		
A	$M \times N$	Array A storing N chains of length M . Constraint: $N > 1$.
M0		Integer M_0 specifying the burn-in period. Constraint: $M > M_0 \geq 0$. <i>Note:</i> the convergence indices are calculated using $A_{q,r}$, $q \geq M_0 + 1$.
Output		
Rhat		Convergence index $\hat{R}$. In theory, $\hat{R} \geq 1$ and the closer the value is to 1, the more confidence that convergence has been achieved. The output value is $\max\{\hat{R}, 1\}$.
Neff		Effective number n_{eff} of independent draws. In theory, $n_{\text{eff}} \leq (M - M_0)N$ and the closer n_{eff} is to the limit $(M - M_0)N$, the less autocorrelation in the chains. The output value is $\min\{(M - M_0)N, n_{\text{eff}}\}$.
Abar	$M \times 1$	Vector storing the means $\bar{a}_q$ of the rows $A_{q,r}$, $r = 1, \dots, N$, of the array A .
stdA	$M \times 1$	Vector storing the standard deviations s_q of the rows $A_{q,r}$, $r = 1, \dots, N$, of the array A .

3.1.4 Calculate sample summary information

The software component `mcmcsun.m` has calling syntax

```
[abar, s, aQ, V, AA] = mcmcsun(A, M0, Q)
```

This component may be applied to the outputs of any MCMC scheme (or indeed any sampling scheme). It provides estimates of quantiles derived from the samples associated with the quantities. Here, A contains the samples for a vector quantity and is of dimension $M \times N \times L$. For the case where the vector quantity comprises α and β , $L = k + m$.

Given $0 \leq q \leq 100$, the associated quantile Q is such that

$$\Pr(\alpha \leq Q) = q/100.$$

If $q = 0$, the component calculates the minimum of the sample; if $q = 100$, the component calculates the maximum of the sample.

Name	Size	Description
Inputs		
A	$M \times N \times L$	Array storing N chains of length M for L quantities. Constraint: $N > 1$.
M0		Integer M_0 specifying the burn-in period. Constraint: $M > M_0 \geq 0$. <i>Note:</i> the summary information for the j th quantity is calculated using $A_{q,r,j}$, $q \geq M_0 + 1$.
Q	$n_Q \times 1$	Quantile specifications q_i , with $0 \leq q_i \leq 100$.
Output		
abar	$L \times 1$	Vector storing the means of the samples.
s	$L \times 1$	Vector storing the standard deviations of the samples.
aQ	$n_Q \times L$	Array storing the estimated quantiles.
V	$L \times L$	Array storing the variance matrix of the samples.
AA	$n \times L$	Array storing the samples for each quantity as a column vector, with $n = (M - M_0)N$.

3.2 MHIC applied to uncertainty evaluation

The software component `mcm2mcmc.m` has calling syntax

```
[A, Rhat, Neff, IS, IA] = mcm2mcmc(A0, D, M0, P00)
```

This component generates Markov chains associated with the target distribution $p(\alpha, \beta | \zeta)$ given a sample from the distribution $p_{\text{MCM}}(\alpha, \beta | \zeta)$. The samples are generated as follows. For $q = 1, \dots, M$, and $r = 1, \dots, N$, sample $\mathbf{y}_{q,r}$ according to $p(\eta | \zeta)$ and $\mathbf{b}_{q,r}$ according to $p(\beta)$ and evaluate

$$\mathbf{a}_{0,q,r} = \psi(\mathbf{y}_{q,r}, \mathbf{b}_{q,r}), \quad D_{q,r} = |J(\mathbf{a}_{0,q,r}, \mathbf{b}_{q,r})|, \quad P_{00,q,r} = p_{00}(\mathbf{a}_{q,r}).$$

For univariate models (as covered by GUMS1), $D(q, r)$ stores

$$D_{q,r} = \left| \frac{\partial \phi}{\partial \alpha}(a_{0,q,r}, \mathbf{b}_{q,r}) \right|.$$

The samples corresponding to quantities α and β are stored in $L = k + m$ arrays of dimension $M \times N$.

`mcm2mcmc.m` calls modules `mhicI.m`, `mhicia2a.m` and `mcmcci.m`.

Name	Size	Description
		Inputs
A0	$M \times N \times L$	Array A_0 storing the $M \times N$ arrays of samples drawn from $p_{\text{MCM}}(\alpha, \beta \zeta)$ for the L quantities.
D	$M \times N$	Array D storing the absolute value of the determinant of the Jacobian matrix: $D_{q,r} = J(\mathbf{a}_{0,q,r}, \mathbf{b}_{q,r}) $.
M0		Integer M_0 specifying the burn-in period. Constraint: $M > M_0 \geq 0$.
P00	$M \times N$	Optional array P_{00} storing the prior density $P_{00,q,r} = p_{00}(\mathbf{a}_{0,q,r})$. If $p_{00}(\alpha) \propto 1$ is the non-informative prior, P_{00} does not need to be provided as an input argument to <code>mcm2mcmc.m</code> .
		Output
A	$M \times N \times L$	Array A storing the N Markov chains of length M for the L quantities.
Rhat	$L \times 1$	Convergence index $\hat{R}$ for each of the quantities; see <code>mcmcci.m</code> .
Neff	$L \times 1$	Effective number n_{eff} of independent draws for each of the quantities; see <code>mcmcci.m</code> .
IS	$M \times N$	Array I_S of selection indices; see <code>mhicI.m</code> .
IA	$M \times N$	Array I_A of acceptance indices; see <code>mhicI.m</code> .

4 Numerical examples

4.1 Gauge block examples

The gauge block examples involve the observation and measurement models

$$\eta = \phi(\alpha, \beta_1, \beta_2) = \alpha [1 + \beta_2(\beta_1 - b_1)], \quad \alpha = \psi(\eta, \beta_1, \beta_2) = \frac{\eta}{1 + \beta_2(\beta_1 - b_1)},$$

where η is the length (mm) of the gauge block at temperature $\beta_1(^{\circ}\text{C})$, α is the length (mm) of the gauge block at the reference temperature $b_1 = 20^{\circ}\text{C}$, and β_2 is the coefficient of thermal expansion ($^{\circ}\text{C}^{-1}$).

Rectangular priors are assigned to β_1 and β_2 :

$$\beta_1 \sim \text{R}(18, 22), \quad \beta_2 \sim \text{R}(0.09, 0.11).$$

A non-informative prior is assumed for α , $p_{00}(\alpha) \propto 1$.

Three examples are considered:

1. Likelihood specified by a Gaussian distribution (Section 4.1.1)
2. Likelihood defined by a Student's t distribution (Section 4.1.2)
3. Likelihood specified by a symmetric Beta distribution (Section 4.1.3)

These examples illustrate that the algorithm is applicable to a range of likelihood functions and is not restricted to the Gaussian case alone. In particular, the Student's t likelihood indicates that ζ is derived from repeated measurements, and the symmetric Beta likelihood indicates that ζ has a distribution that is symmetric but is finite in extent.

Figure 1 shows the shapes of the likelihoods $p(\zeta|\eta)$ for each of the three cases.

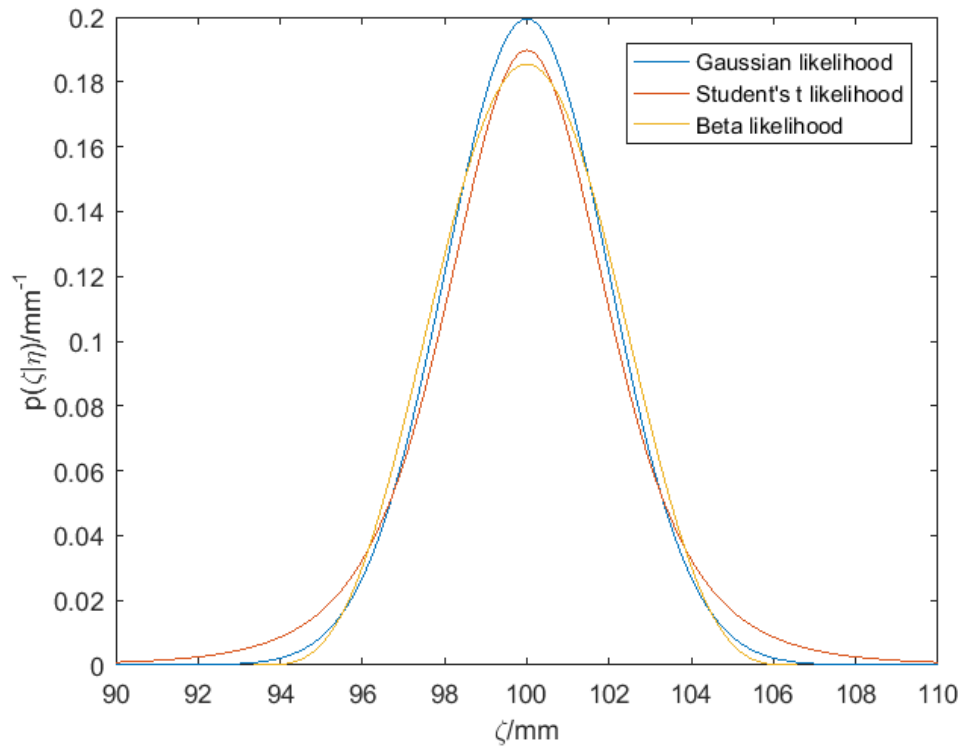


Figure 1: Sampling distributions for $\zeta|\eta$.

4.1.1 Example 1: Likelihood defined by a Gaussian distribution

The likelihood of observing ζ , given η , is specified by

$$\zeta|\eta \sim N(\eta, \sigma^2).$$

An indication $\zeta = 100$ is observed, with σ taken to be 2, so that $\eta|\zeta$ is assigned the distribution

$$\eta|\zeta \sim N(100, 2^2).$$

The MCM approach samples

$$b_{1,q,r} \in R(18, 22), \quad b_{2,q,r} \in R(0.09, 0.11) \quad \text{and} \quad y_{q,r} \in N(100, 2^2),$$

and evaluates

$$a_{0,q,r} = \frac{y_{q,r}}{1 + b_{2,q,r}(b_{1,q,r} - b_1)}.$$

In order to implement the MCMC algorithm, it is also required to evaluate

$$D(q, r) = \left| \frac{\partial \phi}{\partial \alpha}(a_{0,q,r}, b_{1,q,r}, b_{2,q,r}) \right| = |1 + b_{2,q,r}(b_{1,q,r} - b_1)|.$$

The array A_0 is of dimension $M \times N \times 3$, comprising N chains of length M for each of the three quantities α , β_1 and β_2 .

The calculations are performed in `r_gauge_block_example_1.m` and the results are published in `r_gauge_block_example_1.html`. Convergence indices along with the summary statistics based on the posterior distribution are displayed below.

Figure 2 shows the posterior distributions for α , β_1 and β_2 estimated from the MCM and MCMC samples.

Percentage acceptance: 93

Convergence indices

```
Parameter 1: 1.000139
Parameter 2: 1.000116
Parameter 3: 1.000065
```

Effective number of independent draws

```
Parameter 1: 78276
Parameter 2: 81243
Parameter 3: 88533
```

Summary information for posterior distribution

Mean

	MCM	MCMC
Parameter 1:	101.333	102.742
Parameter 2:	20.0046	19.8691
Parameter 3:	0.0999784	0.0999807

Standard deviation

	MCM	MCMC
Parameter 1:	12.0897	12.2173
Parameter 2:	1.15601	1.15481
Parameter 3:	0.00577961	0.00578233

0 percentiles

	MCM	MCMC
Parameter 1:	76.4781	77.8602
Parameter 2:	18.0001	18.0001
Parameter 3:	0.0900003	0.0900003

2.5 percentiles

	MCM	MCMC
Parameter 1:	83.2705	83.5541
Parameter 2:	18.102	18.084
Parameter 3:	0.0905127	0.0905162

50 percentiles

	MCM	MCMC
Parameter 1:	99.9574	102.055
Parameter 2:	20.0042	19.7962
Parameter 3:	0.0999586	0.0999513

97.5 percentiles

	MCM	MCMC
Parameter 1:	124.3	124.815
Parameter 2:	21.9014	21.8821
Parameter 3:	0.109519	0.109517

100 percentiles

	MCM	MCMC
Parameter 1:	134.506	134.506
Parameter 2:	22	22
Parameter 3:	0.11	0.11

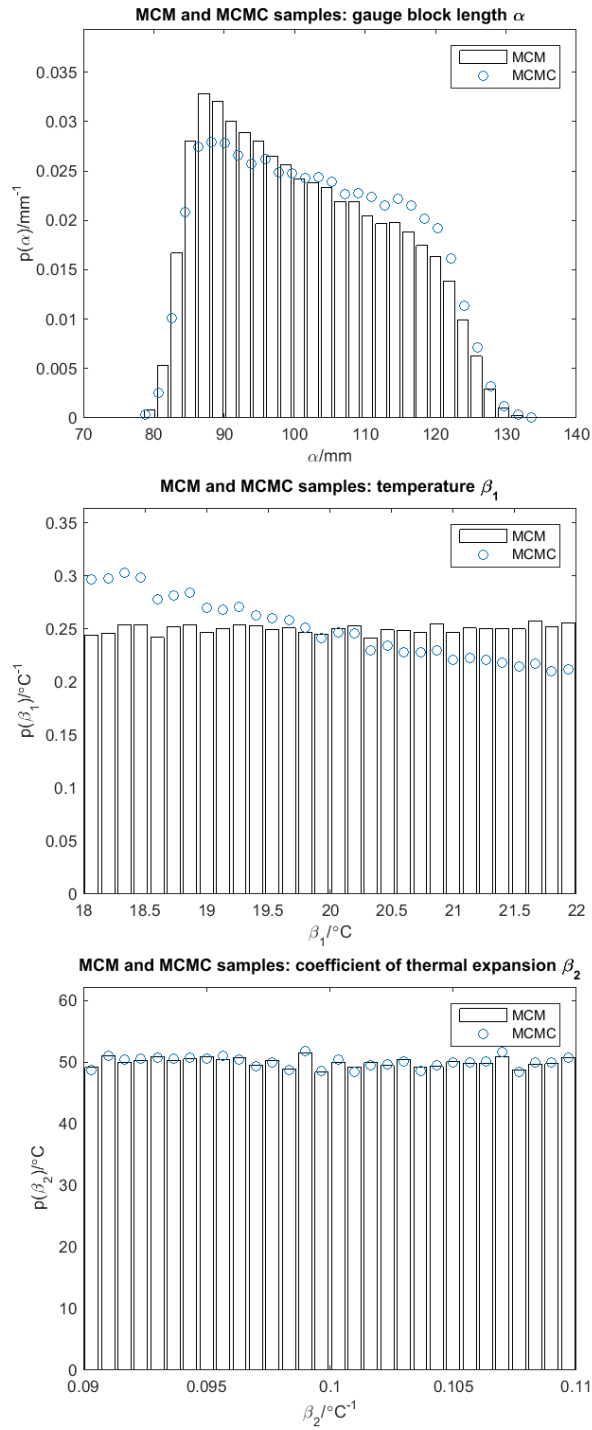


Figure 2: Posterior distributions estimated from MCM and MCMC samples for gauge block example 1.

4.1.2 Example 2: Likelihood defined by a Student's t-distribution

The likelihood of observing ζ , given η , is specified by

$$\zeta|\eta \sim t_\nu(\eta, s^2).$$

An indication $\zeta = 100$ is observed, and s and ν are taken to be 2 and 5, respectively, so that $\eta|\zeta$ is assigned the distribution

$$\eta|\zeta \sim t_5(100, 2^2).$$

The MCM approach samples

$$b_{1,q,r} \in \mathcal{R}(18, 22), \quad b_{2,q,r} \in \mathcal{R}(0.09, 0.11) \quad \text{and} \quad y_{q,r} \in t_5(100, 2^2),$$

and evaluates

$$a_{0,q,r} = \frac{y_{q,r}}{1 + b_{2,q,r}(b_{1,q,r} - b_1)}.$$

In order to implement the MCMC algorithm, it is also required to evaluate

$$D(q, r) = |1 + b_{2,q,r}(b_{1,q,r} - b_1)|.$$

The array A_0 is of dimension $M \times N \times 3$, comprising N chains of length M for each of the three quantities α , β_1 and β_2 .

The calculations are performed in `r_gauge_block_example_2.m` and the results are published in `r_gauge_block_example_2.html`. Convergence indices along with the summary statistics based on the posterior distribution are displayed below.

Figure 3 shows the posterior distributions for α , β_1 and β_2 estimated from the MCM and MCMC samples.

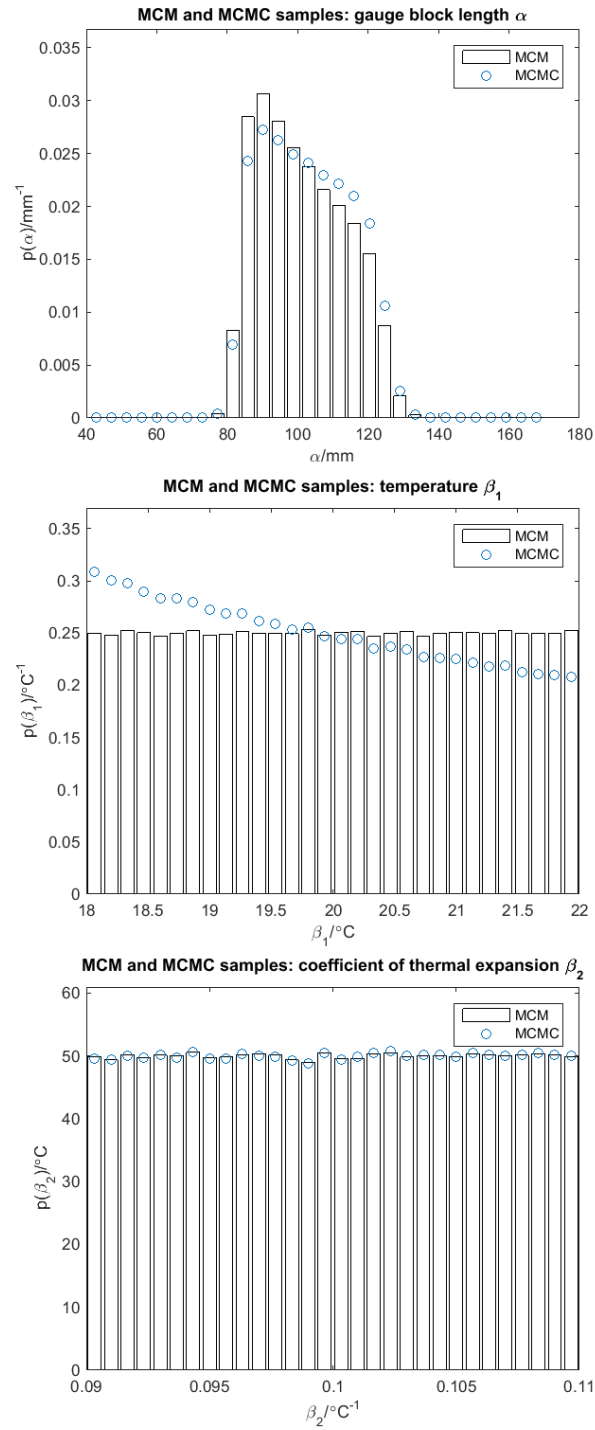


Figure 3: Posterior distributions estimated from MCM and MCMC samples for gauge block example 2.

4.1.3 Example 3: Likelihood specified by a symmetric Beta distribution

The likelihood of observing ζ , given η , is specified by

$$\zeta|\eta \sim \text{Beta}(\eta - \kappa, \eta + \kappa, p_S, p_S),$$

where $\text{Beta}(\eta - \kappa, \eta + \kappa, p_S, p_S)$ is the symmetric standard Beta distribution defined by shape parameters $p_{S,1} = p_S$ and $p_{S,2} = p_S$, shifted and scaled such that it is non-zero only on the interval $(\eta - \kappa, \eta + \kappa)$. The value $\kappa = \sigma(2p_S + 1)^{1/2}$ is chosen so that the standard deviation of the distribution is σ .

An indication $\zeta = 100$ is observed, and σ and p_S are taken to be 2 and 5, respectively, so that $\eta|\zeta$ is assigned the distribution

$$\eta|\zeta \sim \text{Beta}(100 - \kappa, 100 + \kappa, 5, 5),$$

where $\kappa = 2\sqrt{11}$. The MCM approach samples

$$b_{1,q,r} \in \mathcal{R}(18, 22), \quad b_{2,q,r} \in \mathcal{R}(0.09, 0.11) \quad \text{and} \quad y_{q,r} \in \text{Beta}(100 - \kappa, 100 + \kappa, 5, 5),$$

and evaluates

$$a_{0,q,r} = \frac{y_{q,r}}{1 + b_{2,q,r}(b_{1,q,r} - b_1)}.$$

In order to implement the MCMC algorithm, it is also required to evaluate

$$D(q, r) = |1 + b_{2,q,r}(b_{1,q,r} - b_1)|.$$

The array A_0 is of dimension $M \times N \times 3$, comprising N chains of length M for each of the three quantities α , β_1 and β_2 .

The calculations are performed in `r_gauge_block_example_3.m` and the results are published in `r_gauge_block_example_3.html`. Convergence indices along with the summary statistics based on the posterior distribution are displayed below.

Figure 4 shows the posterior distributions for α , β_1 and β_2 estimated from the MCM and MCMC samples.

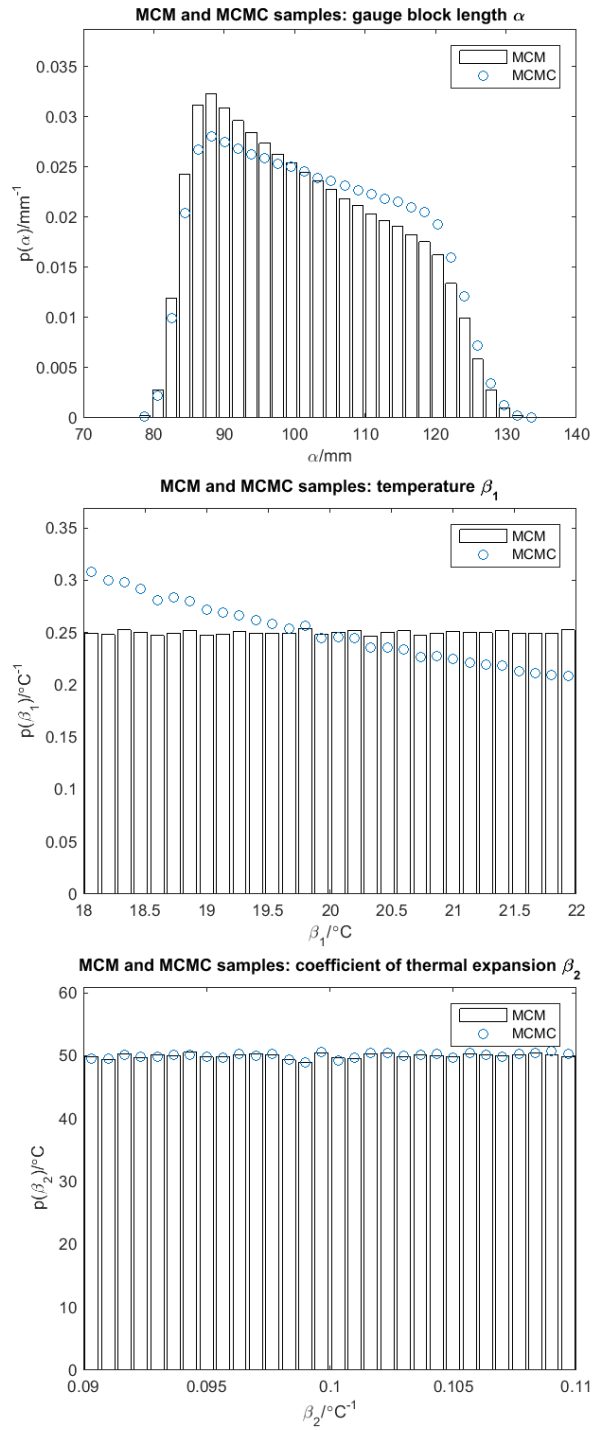


Figure 4: Posterior distributions estimated from MCM and MCMC samples for gauge block example 3.

4.2 Exponential examples

Two examples are considered:

1. A univariate model (Section 4.2.1)
2. A bivariate model (Section 4.2.2)

See also [6].

4.2.1 Example 1: Univariate problem

These calculations involve the response model

$$\eta = \phi(\alpha, \beta) = \alpha e^{-\beta}, \quad \zeta | \eta \sim N(\eta, \sigma^2), \quad \beta \sim N(b, \sigma_B^2),$$

for the case $b = 2$, $\sigma = 0.2$, $\sigma_B = 0.2$ and $\zeta = 50e^{-b}$. On this basis, $\eta | \zeta$ is assigned the distribution

$$\eta | \zeta \sim N(\zeta, \sigma^2).$$

A non-informative prior is assumed for α .

Given draws $y_{q,r} \in N(\zeta, \sigma^2)$ and $b_{q,r} \in N(b, \sigma_B^2)$, assign

$$a_{0,q,r} = y_{q,r} \exp(b_{q,r}), \quad D(q, r) = \exp(-b_{q,r}).$$

The array A_0 is of dimension $M \times N \times 2$, comprising N chains of length M for each of the two quantities α and β .

The calculations are performed in `r_exponential_example_1.m` and the results are published in `r_exponential_example_1.html`. Convergence indices along with the summary statistics based on the posterior distribution are displayed below.

Figure 5 shows the posterior distributions for α and β estimated from the MCM and MCMC samples.

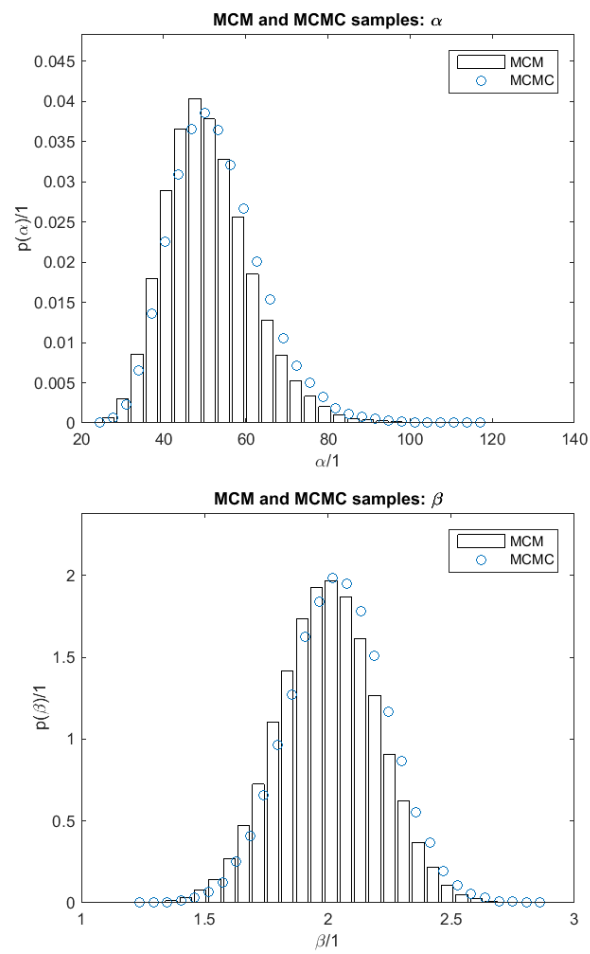


Figure 5: Posterior distributions estimated from MCM and MCMC samples for exponential example 1.

4.2.2 Example 2: Bivariate problem

These calculations involve the response model

$$\begin{aligned}\eta_1 &= \phi_1(\boldsymbol{\alpha}, \beta) = \alpha_1 e^{-\beta}, & \zeta_1 | \eta_1 &\sim N(\eta_1, \sigma^2), \\ \eta_2 &= \phi_2(\boldsymbol{\alpha}, \beta) = (\alpha_1 + \alpha_2) e^{-\beta}, & \zeta_2 | \eta_2 &\sim N(\eta_2, \sigma^2),\end{aligned}$$

where $\beta \sim N(b, \sigma_B^2)$, and there is a prior constraint that $\alpha_2 \geq 0$. This model corresponds, for example, to measuring a background response η_1 and a response η_2 involving an additional source α_2 that is necessarily non-negative.

The Jacobian matrix J of partial derivatives with respect to α_1 and α_2 and its determinant $|J|$ are given by

$$J = \begin{bmatrix} e^{-\beta} & 0 \\ e^{-\beta} & e^{-\beta} \end{bmatrix}, \quad |J| = e^{-2\beta}.$$

Given draws $y_{1,q,r} \in N(\zeta_1, \sigma^2)$, $y_{2,q,r} \in N(\zeta_2, \sigma^2)$ and $b_{q,r} \in N(b, \sigma_B^2)$, assign

$$a_{1,0,q,r} = y_{1,q,r} \exp(b_{q,r}), \quad a_{2,0,q,r} = (y_{2,q,r} - y_{1,q,r}) \exp(b_{q,r}), \quad D(q, r) = \exp(-2b_{q,r}),$$

and $P_{00}(q, r) = 1$ if $a_{2,0,q,r} \geq 0$, and equal to zero otherwise. Some adjustment to the samples may be required to ensure that $a_{2,0,1,r} > 0$ for all r so that $P_{00}(1, r) > 0$ as required by the MHIC algorithm. The array A_0 is of dimension $M \times N \times 3$, comprising N chains of length M for each of the three quantities α_1 , α_2 and β .

Calculations for the case $b = 2$, $\sigma = 0.2$, $\sigma_B = 0.2$, $\zeta_1 = 50e^{-b}$ and $\zeta_2 = \zeta_1 + \sigma$ are performed in `r_exponential_example_2.m` and the results are published in `r_exponential_example_2.html`. Convergence indices along with the summary statistics based on the posterior distribution are displayed below.

Figure 6 shows the posterior distributions for α_1 , α_2 and β estimated from the MCM and MCMC samples.

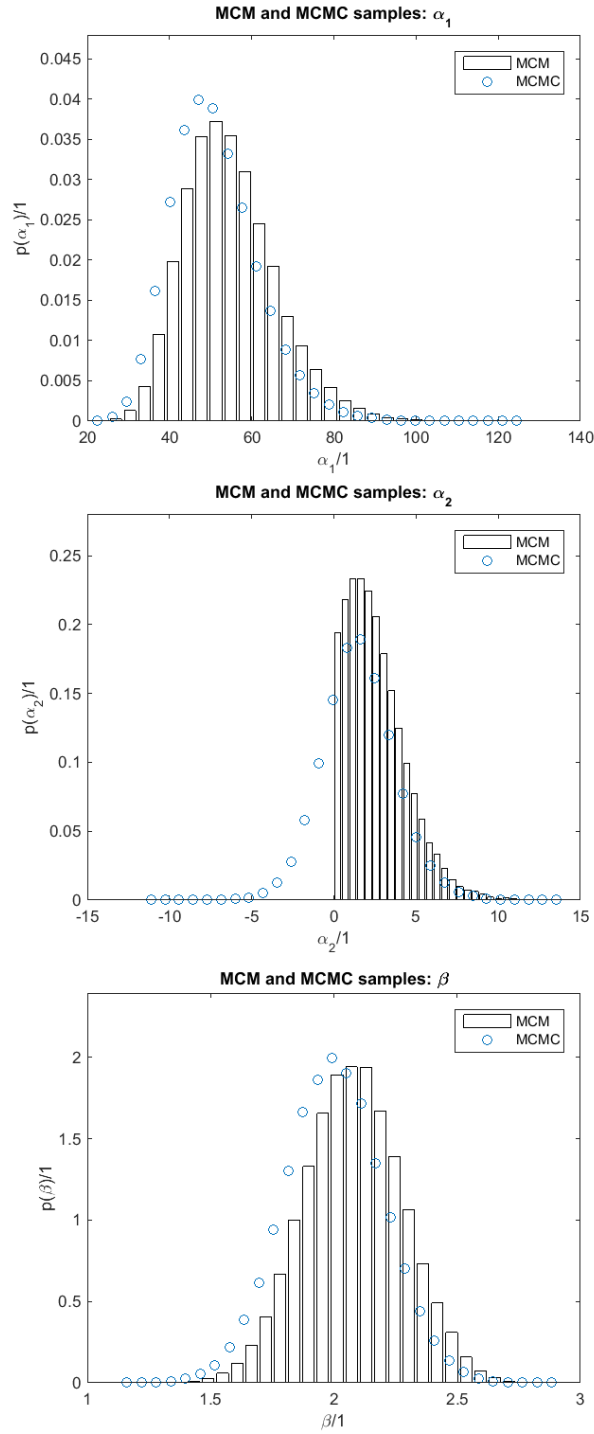


Figure 6: Posterior distributions estimated from MCM and MCMC samples for exponential example 2.

References

- [1] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. Evaluation of measurement data — Guide to the expression of uncertainty in measurement. Joint Committee for Guides in Metrology, JCGM 100:2008.
- [2] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. Evaluation of measurement data — Supplement 1 to the “Guide to the expression of uncertainty in measurement” — Propagation of distributions using a Monte Carlo method. Joint Committee for Guides in Metrology, JCGM 101:2008.
- [3] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML. Evaluation of measurement data — Supplement 2 to the “Guide to the expression of uncertainty in measurement” — extension to any number of output quantities. Joint Committee for Guides in Metrology, JCGM 102:2011.
- [4] M. G. Cox and P. M. Harris. SS/M Best Practice Guide No. 6, Uncertainty evaluation. Technical Report MS 6, National Physical Laboratory, Teddington, UK, 2010.
- [5] C. Elster, W. Wöger, and M. G. Cox. Draft GUM supplement 1 and Bayesian analysis. *Metrologia*, 44:L31–L32, 2007.
- [6] A. B. Forbes. An MCMC algorithm based on GUM supplement 1 for uncertainty evaluation. *Measurement*, 45:1188–1199, 2012.
- [7] A. B. Forbes. A two-stage MCM/MCMC algorithm for uncertainty evaluation. In F. Pavese, M. Bär, J.-R. Filtz, A. B. Forbes, L. Pendrill, and K. Shirono, editors, *Advanced Mathematical and Computational Tools for Metrology IX*, pages 159–170, 2012.
- [8] A. B. Forbes and J. A. Sousa. The GUM, Bayesian inference and the observation and measurement equations. *Measurement*, 44:1422–1435, 2011.
- [9] D. Gamerman. *Markov chain Monte Carlo: Stochastic Simulation for Bayesian inference*. Taylor & Francis, New York, 1997.
- [10] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, Boca Raton, Fl., third edition, 2014.