Chapter 26
Finite Element Model Updating Using an Evolutionary Markov Chain Monte Carlo Algorithm

I. Boulkaibet, L. Mthembu, T. Marwala, M.I. Friswell, and S. Adhikari

Abstract One challenge in the finite element model (FEM) updating of a physical system is to estimate the values of the uncertain model variables. For large systems with multiple parameters this requires simultaneous and efficient sampling from multiple a prior unknown distributions. A further complication is that the sampling method is constrained to search within physically realistic parameter bounds. To this end, Markov Chain Monte Carlo (MCMC) techniques are popular methods for sampling from such complex distributions. MCMC family algorithms have previously been proposed for FEM updating. Another approach to FEM updating is to generate multiple random models of a system and let these models evolve over time. Using concepts from evolution theory this evolution process can be designed to converge to a globally optimal model for the system at hand. A number of evolution-based methods for FEM updating have previously been proposed. In this paper, an Evolutionary based Markov chain Monte Carlo (EMCMC) algorithm is proposed to update finite element models. This algorithm combines the ideas of Genetic Algorithms, Simulated Annealing, and Markov Chain Monte Carlo techniques. The EMCMC is global optimisation algorithm where genetic operators such as mutation and crossover are used to design the Markov chain to obtain samples. In this paper, the feasibility, efficiency and accuracy of the EMCMC method is tested on the updating of a real structure.

Keywords Bayesian • Finite element model updating • Markov chain Monte Carlo • Evolutionary Markov chain Monte Carlo • Simulated annealing • Genetic algorithms

26.1 Introduction

A Finite Element Model (FEM) [1, 2] is by definition a particular mathematical approximation of some real system. This approximation can be reasonably accurate for simple systems. For sufficiently complex systems the accuracy of this approximation degrades. Complex systems are those where a number of uncertain variables have to be defined, modelled and valued, and in dynamics systems this can be very difficult to do.

If it is clear which variables are uncertain, one approach is to understand and/or define their probability distributions. This is possible for FE models of real systems because the uncertain variable values have to be of realistic magnitudes. The sampling techniques can be employed to quantify these uncertain parameters. However, the challenge is that sometimes the distribution of these uncertain variables is not known a prior, and sometimes the updated models have multiple optimal (or near optimal) solutions.

The most popular sampling algorithms are of the Markov chain Monte Carlo type [1, 3, 4]. They have the advantage that the sampling procedures draw samples with an element of randomness while being guided by their performance on the problem objective function (also known as the Posterior function). Thus some samples will be rejected while others will be accepted and this effectively forms a chain of samples. The learning part of the sampling procedure proposed in this paper uses the paradigm of evolution-based algorithms [5–7].

These algorithms are based on the notion that in evolution, new individuals are an improvement based on their parents' performance in a particular problem. As such, new individuals can be developed if we can understand/model the characteristics/variables that made their parents good. The analogy of individuals in evolution algorithms to FEM context is

I. Boulkaibet (✉) • L. Mthembu • T. Marwala
The Centre for Intelligent System Modelling (CISM), Electrical and Electronic Engineering Department, University of Johannesburg, PO Box 524, Auckland Park 2006, South Africa
e-mail: iboulkaibet@student.uj.ac.za

M.I. Friswell • S. Adhikari
College of Engineering, Swansea University, Singleton Park, Swansea SA2 8PP, UK

J. Caicedo, S. Pakzad (eds.), *Dynamics of Civil Structures, Volume 2*, Conference Proceedings of the Society for Experimental Mechanics Series, DOI 10.1007/978-3-319-15248-6_26

the particular combination of uncertain variable values that form a potentially correct solution to the FEM problem. Thus individuals in evolution algorithms are all the potential solutions to the FE model. The individual's performance on the problem is evaluated by the fitness function. In this paper, the fitness function is defined by the logarithm of the posterior Density Distribution Function (PDF), see next section.

There are two fundamental mechanisms of developing new individuals in evolution algorithms; mutation and crossover. Mutation is when a variable, randomly selected, changes its value. The crossover is the process where two individuals exchange their variable values at certain positions, also known as crossover points, in the uncertain variable vector. The selection of the positions can either be done via a roulette wheel selection or a random selection [8, 9]. The algorithm proposed in this work has an extra operator called the exchange [8, 10]. This operator allows for the mixing of individuals within a population to eliminate premature convergence. In order to characterise the uncertainties of the update variables, we formulate the problem in the Bayesian context. The next section introduces the Bayesian formulation. Section 26.3 describes the Evolutionary MCMC algorithm while Sect. 26.4 presents the results when an Unsymmetrical H-shaped Structure is updated using the Evolutionary MCMC algorithm. The paper is concluded in Sect. 26.5.

26.2 The Bayesian Formulation

The Bayesian method is used to solve the FEM updating problem in the modal domain. Bayesian approaches are governed by Bayes rule [3, 4, 11–13]:

$$P\left(\boldsymbol{\theta}\middle|\mathcal{D},\mathcal{M}\right) \propto P\left(\mathcal{D}\middle|\boldsymbol{\theta},\ \mathcal{M}\right) P\left(\boldsymbol{\theta}\middle|\mathcal{M}\right) \tag{26.1}$$

$\mathcal{M}$ represents the model class for the target system which is defined by the model updated variables $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathcal{R}^d$. $\mathcal{D}$ is the measured modal data (typically the natural frequencies f_i^m and mode shapes ϕ_i^m) in the FE model field. The quantity $P\left(\boldsymbol{\theta}\middle|\mathcal{M}\right)$ is the prior probability distribution function (PDF). It represents our knowledge of the uncertain variables given a particular model $\mathcal{M}$ in the absence of any measured data $\mathcal{D}$. The quantity $P\left(\boldsymbol{\theta}\middle|\mathcal{D},\ \mathcal{M}\right)$ is the posterior PDF of the variables once we have observed the data and the assumed model class $\mathcal{M}$. $P\left(\mathcal{D}\middle|\boldsymbol{\theta},\ \mathcal{M}\right)$ is the likelihood function [3, 12–15] which calculates the difference between the measured data and the FE model results for given variables and assumed model. The dependence on the model class $\mathcal{M}$ is only relevant in cases where more than one model class is investigated and in this paper only one model class is considered. For simplicity we ignore the class notation in subsequent notations.

In the FEM context the likelihood PDF form is given by:

$$P\left(\mathcal{D}\middle|\boldsymbol{\theta}\right) = \frac{1}{\left(\frac{2\pi}{\beta_c}\right)^{N_m/2}\prod_{i=1}^{N_m} f_i^m} \exp\left(-\frac{\beta_c}{2}\sum\nolimits_i^{N_m}\left(\frac{f_i^m - f_i}{f_i^m}\right)^2\right) \tag{26.2}$$

where β_c is a constant, N_m is the number of measured modes, f_i^m and f_i are the ith analytical natural frequency and the ith measured natural frequency.

The prior PDF represents our prior knowledge of the updating variables $\boldsymbol{\theta}$ and it is chosen as a Gaussian distribution

$$P\left(\boldsymbol{\theta}\right) = \frac{1}{(2\pi)^{Q/2}\prod_{i=1}^{Q}\frac{1}{\sqrt{\alpha_i}}} \exp\left(-\sum\nolimits_i^{Q}\frac{\alpha_i}{2}\left\|\boldsymbol{\theta}^i - \boldsymbol{\theta}_0^i\right\|^2\right) = \frac{1}{(2\pi)^{Q/2}\prod_{i=1}^{Q}\frac{1}{\sqrt{\alpha_i}}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^{\boldsymbol{T}}\,\Sigma^{-1}\,(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\right) \tag{26.3}$$

where Q is the number of variables to be updated, $\boldsymbol{\theta}_0$ represents the mean value of the updated vector and α_i is the coefficient of the prior PDF for the ith updating variables. The notation * denotes the Euclidean norm of *.

The posterior PDF of the variables $\boldsymbol{\theta}$ given the observed data $\mathcal{D}$ is denoted as $P\left(\boldsymbol{\theta}\middle|\mathcal{D}\right)$ and is obtained by applying Bayes' theorem in Eq. (26.1). The distribution $P\left(\boldsymbol{\theta}\middle|\mathcal{D}\right)$ is calculated by substituting Eqs. (26.2) and (26.3) into Eq. (26.1) to give:

$$P\left(\boldsymbol{\theta}\middle|\mathcal{D}\right) \propto \frac{1}{Z_s\left(\alpha, \beta_c\right)} \exp\left(-\frac{\beta_c}{2}\sum\nolimits_i^{N_m}\left(\frac{f_i^m - f_i}{f_i^m}\right)^2 - \sum\nolimits_i^Q \frac{\alpha_i}{2}\left\|\boldsymbol{\theta}^{\boldsymbol{i}} - \boldsymbol{\theta}_0^{\boldsymbol{i}}\right\|^2\right) \tag{26.4}$$

where

$$Z_s\left(\alpha, \beta_c\right) = \left(\frac{2\pi}{\beta_c}\right)^{N_m/2}\prod\nolimits_{i=1}^{N_m} f_i^m\left(2\pi\right)^{Q/2}\prod\nolimits_{i=1}^{Q}\frac{1}{\sqrt{\alpha_i}} \tag{26.5}$$

In complex systems (e.g. those that have a large number of uncertain variables and model intricate systems) obtaining a posterior PDF analytical form is not possible. Sampling techniques, such as Markov chain Monte Carlo (MCMC) methods, can provide a numerical approximation to this function [1, 12, 13, 15–17].

If Y is an observation of certain variables at different discrete time instants, then the prediction of the future responses of this parameter Y at different time instants can be achieved by the total probability theorem:

$$P\left(Y\middle|D\right) = \int\limits_{\theta} P\left(Y\middle|\boldsymbol{\theta}\right) P\left(\boldsymbol{\theta}\middle|D\right) d\boldsymbol{\theta} \tag{26.6}$$

Equation (26.6) depends on the posterior PDF. Therefore, given a set of N_s random variable vectors drawn from a probability distribution function ($P\left(\boldsymbol{\theta}\middle|D\right)$, the expectation value of any observed function Y can be easily estimated.

The integral in Eq. (26.6) can be solved using sampling algorithms [1, 3, 4, 12, 13]. These algorithms are used to generate a sequence of vectors $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{N_s}\}$, where N_s is the number of samples of these vectors, that can be used to form a Markov chain. This generated vector is then used to predict the form of the posterior distribution function $P\left(\boldsymbol{\theta}\middle|D\right)$. The integral in Eq. (26.6) can be approximated as

$$\widetilde{Y} \cong \frac{1}{N_s}\sum\nolimits_{i=1}^{N_S} G\left(\boldsymbol{\theta}_i\right) \tag{26.7}$$

where G is a function that depends on the updated variables $\boldsymbol{\theta}_i$. As an example, if $\mathbf{G} = \boldsymbol{\theta}$ then $\widetilde{Y}$ becomes the expected value of $\boldsymbol{\theta}$. Generally, $\widetilde{Y}$ is the vector that contains the modal properties and N_s is the number of retained states. In this paper, the evolutionary Markov chain Monte Carlo (EMCMC) method is used to sample from the posterior PDF.

26.3 The Evolutionary MCMC Algorithm

Genetic algorithms (GA) make use of the two operators; crossover and mutation to succeed in many hard optimization problems [8, 18]. The EMCMC [9, 19] algorithm combines these GA operators with the dynamics of MCMC algorithms to better sample and learns from the sampling procedure. In this section, the basics of the EMCMC technique are described.

Let $\boldsymbol{\theta} = \left\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^{\boldsymbol{i}}, \cdots, \boldsymbol{\theta}^N\right\}$ denote a population of samples, where $\boldsymbol{\theta}^{\boldsymbol{i}} = \left\{\boldsymbol{\theta}_1^{\boldsymbol{i}}, \boldsymbol{\theta}_2^{\boldsymbol{i}}, \cdots, \boldsymbol{\theta}_d^{\boldsymbol{i}}\right\}$ is a d-dimensional vector called an individual or a chromosome in GA and N is the population size. In Bayesian statistics, $\boldsymbol{\theta}^{\boldsymbol{i}}$ is often a vector of variables while the fitness function $F(\boldsymbol{\theta}^{\boldsymbol{i}})$ is the negative of the log-posterior of $\boldsymbol{\theta}^{\boldsymbol{i}}$. In EMCMC, a different temperature T_i is attached to each individual $\boldsymbol{\theta}^{\boldsymbol{i}}$, and the temperatures form a ladder with the ordering $T_1 > T_1 > T_2 > \cdots > T_N = 1$.

A Boltzmann distribution for each individual $\boldsymbol{\theta}^{\boldsymbol{i}}$ can be defined as:

$$g\left(\boldsymbol{\theta}^{\boldsymbol{i}}\right) = \frac{1}{Z_i\left(T_i\right)} exp\left(-\frac{F\left(\boldsymbol{\theta}^{\boldsymbol{i}}\right)}{T_i}\right) \tag{26.8}$$

where $Z_i(T_i)$ is the normalizing constant, $Z_i\left(T_i\right) = \sum\nolimits_{\{\boldsymbol{\theta}^{\boldsymbol{i}}\}} exp\left(-\frac{F\left(\boldsymbol{\theta}^{\boldsymbol{i}}\right)}{T_i}\right)$

26.3.1 Mutation

In the mutation operator [9], an individual, say $\boldsymbol{\theta}^k$ is randomly selected from the current population $\boldsymbol{\theta} = \{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^k, \cdots, \boldsymbol{\theta}^N\}$, it is then mutated to a new individual $\widetilde{\boldsymbol{\theta}}^k$ by changing the values of some bits which are also chosen randomly. A new population is thus created as $\widetilde{\boldsymbol{\theta}} = \left\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \widetilde{\boldsymbol{\theta}}^k, \cdots, \boldsymbol{\theta}^N\right\}$, and it is accepted with probability $min(1, r_m)$ according to the Metropolis rule, where r_m is the Metropolis-Hastings [20, 21] ratio which is given by [9]:

$$r_m = \frac{g\left(\widetilde{\boldsymbol{\theta}}^k\right)}{g\left(\boldsymbol{\theta}^k\right)} \cdot \frac{Tr\left(\boldsymbol{\theta}^k \middle| \widetilde{\boldsymbol{\theta}}^k\right)}{Tr\left(\widetilde{\boldsymbol{\theta}}^k \middle| \boldsymbol{\theta}^k\right)} = \exp\left(-\frac{\left(F\left(\widetilde{\boldsymbol{\theta}}^k\right) - F\left(\boldsymbol{\theta}^k\right)\right)}{T_k}\right) \cdot \frac{Tr\left(\boldsymbol{\theta}^k \middle| \widetilde{\boldsymbol{\theta}}^k\right)}{Tr\left(\widetilde{\boldsymbol{\theta}}^k \middle| \boldsymbol{\theta}^k\right)} \tag{26.9}$$

and $Tr\left(.\middle|.\right)$ denotes the transition probability between populations. If the proposal is accepted, the current population $\boldsymbol{\theta}$ is replaced by $\widetilde{\boldsymbol{\theta}}$, otherwise the population $\boldsymbol{\theta}$ is unchanged. In this paper, the 2-point mutation operator is performed which means that $Tr\left(\boldsymbol{\theta}^k \middle| \widetilde{\boldsymbol{\theta}}^k\right) = Tr\left(\widetilde{\boldsymbol{\theta}}^k \middle| \boldsymbol{\theta}^k\right)$.

26.3.2 Crossover

In the crossover operator [9]; two individuals, say $\boldsymbol{\theta}^a$ and $\boldsymbol{\theta}^b$ $(a \neq b)$, are selected from the current population $\boldsymbol{\theta}$ according to some selection procedure. In the analysis we assume $F\left(\boldsymbol{\theta}^a\right) \geq F\left(\boldsymbol{\theta}^b\right)$. From these two individuals, now known as parents, two new individuals are generated according to the crossover operator. This can be done by randomly selecting a crossover position in the uncertain variable vector. That is if we let position p in the size d element variable vector be the crossover position then the elements from position p + *1* in $\boldsymbol{\theta}^a$ will be moved to individual $\boldsymbol{\theta}^b$ and vice versa. The offspring with a smaller fitness value is denoted $\widetilde{\boldsymbol{\theta}}^b$ and the other $\widetilde{\boldsymbol{\theta}}^a$. A new population of individuals is proposed as $\widetilde{\boldsymbol{\theta}} = \left\{\boldsymbol{\theta}^1, \cdots, \widetilde{\boldsymbol{\theta}}^a, \cdots, \widetilde{\boldsymbol{\theta}}^b, \cdots, \boldsymbol{\theta}^N\right\}$. According to the Metropolis rule, the new population is accepted with probability $in(1, r_c)$ [9],

$$r_c = \frac{g\left(\widetilde{\boldsymbol{\theta}}^a\right) \cdot g\left(\widetilde{\boldsymbol{\theta}}^b\right)}{g\left(\boldsymbol{\theta}^a\right) \cdot g\left(\widetilde{\boldsymbol{\theta}}^b\right)} \cdot \frac{Tr\left(\boldsymbol{\theta} \middle| \widetilde{\boldsymbol{\theta}}\right)}{Tr\left(\widetilde{\boldsymbol{\theta}} \middle| \boldsymbol{\theta}\right)} = \exp\left(-\frac{\left(F\left(\widetilde{\boldsymbol{\theta}}^a\right) - F\left(\boldsymbol{\theta}^a\right)\right)}{T_a} - \frac{\left(F\left(\widetilde{\boldsymbol{\theta}}^b\right) - F\left(\boldsymbol{\theta}^b\right)\right)}{T_b}\right) \cdot \frac{Tr\left(\boldsymbol{\theta} \middle| \widetilde{\boldsymbol{\theta}}\right)}{Tr\left(\widetilde{\boldsymbol{\theta}} \middle| \boldsymbol{\theta}\right)} \tag{26.10}$$

where $Tr\left(\boldsymbol{\theta} \middle| \widetilde{\boldsymbol{\theta}}\right) = P\left(\left(\boldsymbol{\theta}^a, \boldsymbol{\theta}^b\right) \middle| \boldsymbol{\theta}\right) \cdot P\left(\left(\widetilde{\boldsymbol{\theta}}^a, \widetilde{\boldsymbol{\theta}}^b\right) \middle| \left(\boldsymbol{\theta}^a, \boldsymbol{\theta}^b\right)\right)$. The $P\left(\left(\boldsymbol{\theta}^a, \boldsymbol{\theta}^b\right) \middle| \boldsymbol{\theta}\right)$ denotes the selection probability of individuals $(\boldsymbol{\theta}^a, \boldsymbol{\theta}^b)$ from the population $\boldsymbol{\theta}$, and $P\left(\left(\widetilde{\boldsymbol{\theta}}^a, \widetilde{\boldsymbol{\theta}}^b\right) \middle| \left(\boldsymbol{\theta}^a, \boldsymbol{\theta}^b\right)\right)$ denotes the generating probability of individuals $\left(\widetilde{\boldsymbol{\theta}}^a, \widetilde{\boldsymbol{\theta}}^b\right)$ from the parents $(\boldsymbol{\theta}^a, \boldsymbol{\theta}^b)$.

In this paper, the parental individuals are chosen as follows; the first individual $\boldsymbol{\theta}^a$ is chosen according to a roulette wheel procedure with Boltzmann weights, i.e. $\boldsymbol{\theta}^a$ is chosen with probability defined in Eq. (26.8). The second individual $\boldsymbol{\theta}^b$ is chosen randomly from the rest of the population. The selection probability of $\boldsymbol{\theta}^b$ is then [9]:

$$P\left(\left(\boldsymbol{\theta}^a, \boldsymbol{\theta}^b\right) \middle| \boldsymbol{\theta}\right) = \frac{1}{(\boldsymbol{N} - 1) \cdot Z(\boldsymbol{\theta})} \cdot \left(exp\left(-\frac{H\left(\boldsymbol{\theta}^a\right)}{T_a}\right) + exp\left(-\frac{H\left(\boldsymbol{\theta}^b\right)}{T_b}\right)\right) \tag{26.11}$$

where $Z(\boldsymbol{\theta}) = \sum_{\{\boldsymbol{\theta}^a\}} exp\left(-\frac{H\left(\boldsymbol{\theta}^a\right)}{T_a}\right)$. $P\left(\left(\widetilde{\boldsymbol{\theta}}^a, \widetilde{\boldsymbol{\theta}}^b\right) \middle| \widetilde{\boldsymbol{\theta}}\right)$. The crossover operator used is a 2-point crossover.

26.3.3 Exchange

This operation was first introduced in the parallel tempering sampling method in [22] and exchange Monte Carlo sampling method [10]. Given the current population $\boldsymbol{\theta}$ and the temperature ladder $\mathbf{T}$, $(\boldsymbol{\theta},\ \mathbf{T}) = \left\{\boldsymbol{\theta}^1, T_1,\ \boldsymbol{\theta}^2,\ T_2, \cdots,\ \boldsymbol{\theta}^N, T_N\right\}$, we make an exchange between individuals $\boldsymbol{\theta}^{\boldsymbol{a}}$ and $\boldsymbol{\theta}^{\boldsymbol{b}}$ without changing their temperatures, i.e., we try to change $(\boldsymbol{\theta},\ \mathbf{T}) = \left\{\boldsymbol{\theta}^1, T_1,\ \boldsymbol{\theta}^{\boldsymbol{a}},\ T_a, \cdots,\ \boldsymbol{\theta}^{\boldsymbol{b}},\ T_b, \cdots,\ \boldsymbol{\theta}^N, T_N\right\}$ to $\left(\overline{\boldsymbol{\theta}},\ \mathbf{T}\right) = \left\{\boldsymbol{\theta}^1, T_1,\ \boldsymbol{\theta}^{\boldsymbol{b}},\ T_a, \cdots,\ \boldsymbol{\theta}^{\boldsymbol{a}},\ T_b, \cdots,\ \boldsymbol{\theta}^N, T_N\right\}$. The new population is accepted with probability $min(1, r_e)$ according to the Metropolis rule, where [9]:

$$r_e = \frac{g\left(\overline{\boldsymbol{\theta}}\right)}{g\left(\boldsymbol{\theta}\right)} \cdot \frac{Tr\left(\boldsymbol{\theta}\middle|\overline{\boldsymbol{\theta}}\right)}{Tr\left(\overline{\boldsymbol{\theta}}\middle|\boldsymbol{\theta}\right)} = \exp\left(-\left(H(\boldsymbol{\theta}^{\boldsymbol{a}}) - H(\boldsymbol{\theta}^{\boldsymbol{b}})\right)\cdot\left(\frac{1}{T_a} - \frac{1}{T_b}\right)\right)\cdot\frac{Tr\left(\boldsymbol{\theta}\middle|\overline{\boldsymbol{\theta}}\right)}{Tr\left(\overline{\boldsymbol{\theta}}\middle|\boldsymbol{\theta}\right)} \tag{26.12}$$

and $Tr\left(.\middle|.\right)$ denotes the transition probability between populations. Typically, the exchange is only performed on states with neighbouring temperature values, i.e., $|a - b| = 1$ where $Tr\left(\boldsymbol{\theta}\middle|\overline{\boldsymbol{\theta}}\right) = Tr\left(\overline{\boldsymbol{\theta}}\middle|\boldsymbol{\theta}\right)$ [9].

The EMCMC algorithm is summarized as follows:

1. Initialization: generate N individuals and evaluate the posterior (fitness) of each individual.
2. Selection: Certain individuals are selected from the current population.
3. Crossover: Offspring are produced by a recombination of the vector elements in the mating individuals where this operation is accepted by $min(1, r_c)$.
4. Mutation: Offspring are produced by random changes in individuals in the population, these offspring are accepted by $min(1, r_m)$.
5. Exchange: The exchange is performed for each set of neighboring individuals in the population, and these are accepted by $min(1, r_e)$.
6. Repeat 2–4 steps until a N_s samples are obtained.

In next sections we present the application and results of this algorithm. The real system modelled is an unsymmetrical H-shaped beam structure. The objective is to predict the uncertain variable values along with their errors. The performance of the proposed algorithm, EMCMC, is compared with the performance of the classic MCMC algorithm Metropolis-Hasting on the same beam.

26.4 Application: Unsymmetrical H-Shaped Structure

In this paper, an unsymmetrical H-shaped aluminium structure (see Fig. 26.1) is updated using the EMCMC algorithm. The structure was divided into 12 elements and each was modelled as an Euler-Bernoulli beam. The Structural Dynamics Toolbox SDT®, which is a Matlab package, is used to model the beams. The position indicated by the double arrow is the location where the structure was excited. The acceleration was measured at 15 different positions. The structure was excited using an electromagnetic shaker and a roving accelerometer was used to measure the response. A set of 15 frequency-response functions were calculated. See [14] for more details about the structure.

26.4.1 H-Beam Simulation

The measured frequencies are: 53.9, 117.3, 208.4, 254.0 and 445.0 Hz. the moments of inertia and the cross section areas of the left, middle and right subsections of the beam are selected to be updated. The updating parameter vector is then given by: $\boldsymbol{\theta} = \{I_{x1}, I_{x2}, I_{x3}, A_{x1}, A_{x2}, A_{x3}\}$. The Young's modulus for the structure is set to $7.2 \times 10^{10}\mathrm{N/m}^2$, the density is set to 2,785 kg/m^3 and the structure dimensions are presented in Fig. 26.1. The updating parameters θ_i are bounded by maximum values equal to $\left[3.73 \times 10^{-8},\ 3.73 \times 10^{-8},\ 3.73 \times 10^{-8}, 4.16 \times 10^{-4},\ 4.16 \times 10^{-4},\ 4.16 \times 10^{-4}\right]$ and minimum values are equal to $\left[1.73 \times 10^{-8},\ 1.73 \times 10^{-8},\ 1.73 \times 10^{-8},\ 2 \times 10^{-4},\ 2.16 \times 10^{-4},\ 2.16 \times 10^{-4}\right]$. These boundaries help to keep the updated vector physically realistic.

Fig. 26.1 The H-shaped aluminium structure

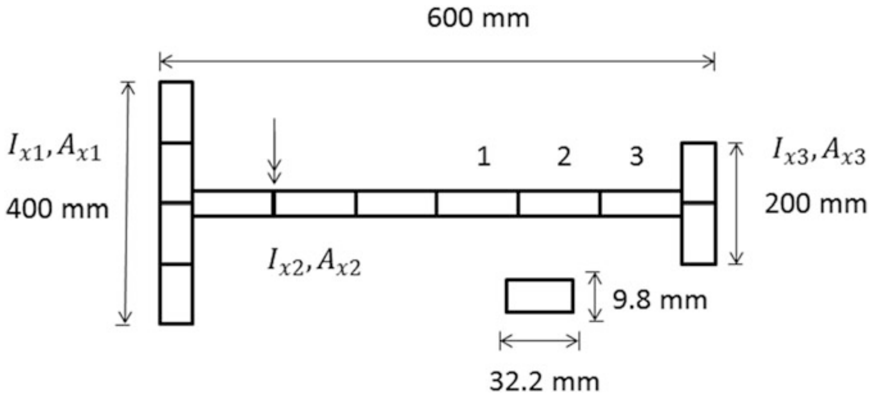

Table 26.1 Initial and updated parameters using EMCMC and M-H algorithms

	Initial $\boldsymbol{\theta}_0$	$\boldsymbol{\theta}$ vector EMCMC method	$\frac{\sigma_i}{\mu_i}$ (%)	$\boldsymbol{\theta}$ vector M-H method	$\frac{\sigma_i}{\mu_i}$ (%)
I_{x1}	2.73×10^{-8}	2.578×10^{-8}	0.75	2.31×10^{-8}	22.59
I_{x2}	2.73×10^{-8}	2.576×10^{-8}	0.78	2.68×10^{-8}	15.25
I_{x3}	2.73×10^{-8}	2.533×10^{-8}	3.84	2.17×10^{-8}	13.96
A_{x1}	3.16×10^{-4}	3.693×10^{-4}	0.04	2.85×10^{-4}	14.36
A_{x2}	3.16×10^{-4}	2.157×10^{-4}	0.001	2.83×10^{-4}	14.36
A_{x3}	3.16×10^{-4}	3.006×10^{-4}	0.003	2.77×10^{-4}	13.08

Table 26.2 Natural frequencies and errors when using EMCMC and M-H algorithms

Mode	Measured frequency (Hz)	Initial frequency (Hz)	Error (%)	Frequencies EMCMC method (Hz)	Error (%)	Frequencies M-H method (Hz)	Error (%)
1	53.9	51.40	4.63	53.1	1.49	53.92	0.04
				(0.013 %)		(3.96 %)	
2	117.3	116.61	0.59	119.99	2.29	122.05	4.05
				(0.041 %)		(4.28 %)	
3	208.4	201.27	3.42	210.23	0.88	210.93	1.22
				(0.172 %)		(4.95 %)	
4	254.0	247.42	2.59	249.01	1.97	258.94	1.94
				(0.091 %)		(4.81 %)	
5	445	390.33	12.28	431.84	2.96	410.33	7.79
				(0.044 %)		(4.74 %)	
Total average error			4.70		1.92		3.01

The constant β_c of the posterior distribution (see Eq. (26.4)) is set equal 1 for the EMCMC algorithm while the β_c is set to 10 for the M-H algorithm. In this paper, the EMCMC results will be compared with those obtained by M-H algorithm (see [23] for more details about the M-H algorithm results). The coefficients α_i are set equal to $\frac{1}{\sigma_i^2}$, where σ_i^2 is the variance of the i th parameter and the variance vector is defined as $\boldsymbol{\sigma} = \left[5 \times 10^{-8},\ 5 \times 10^{-8},\ 5 \times 10^{-8},\ 5 \times 10^{-4},\ 5 \times 10^{-4},\ 5 \times 10^{-4}\right]$. The number of samples N_s is set to 1,000. The population size for the EMCMC algorithm is set to 12; the mutation rate is 0.2 while the selection rate is set to 0.5. The temperatures used in this implementation are given as follows: T = [1,557.8, 1,277.4, 1,054.8, 915.4, 880.7, 815.8, 651, 308.2, 212.2, 189.4, 138.5, 1].

The Bayesian simulation results are presented and evaluated in terms of the mean values of the obtained samples for each method (see Table 26.1). The updated natural frequencies and the prediction error percentage are stored in Table 26.2. The acceptance rate for the EMCMC algorithm is 61.9 % while the acceptance rate of M-H algorithm is 46.9 %. The reason that the M-H algorithm gave a poor acceptance rate is that the move step used for this algorithm was chosen relatively large to have acceptable results with fast convergence (the variance of the proposal distribution is relatively large). The EMCMC algorithm has an acceptable acceptance rate where the reason could be the exchange probabilities used in this implementation (accept with a probability equal to 0.8 when the selected chromosome (or individual) is the first and the last. Otherwise, the probability is equal to 0.5).

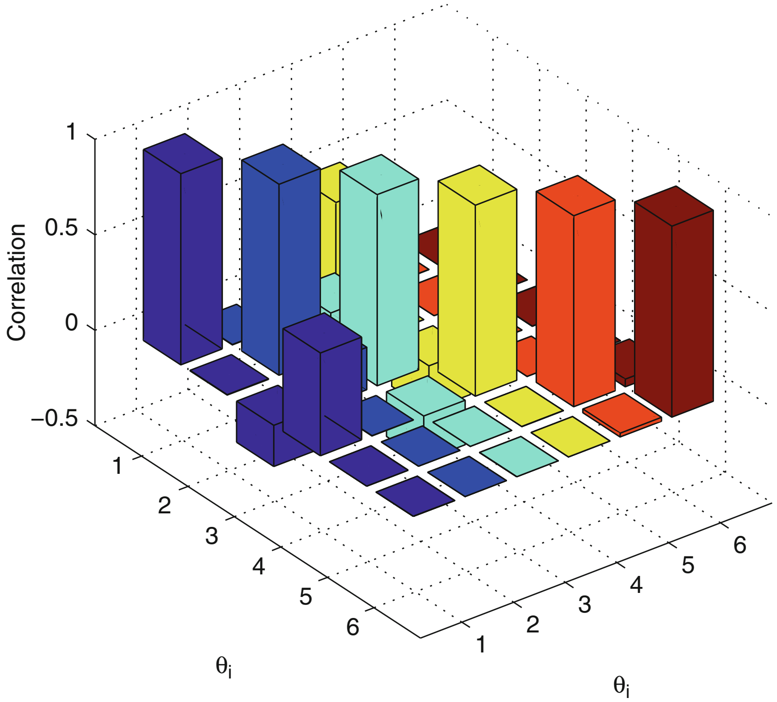

Fig. 26.2 The correlation between the updated parameters (EMCMC algorithm)

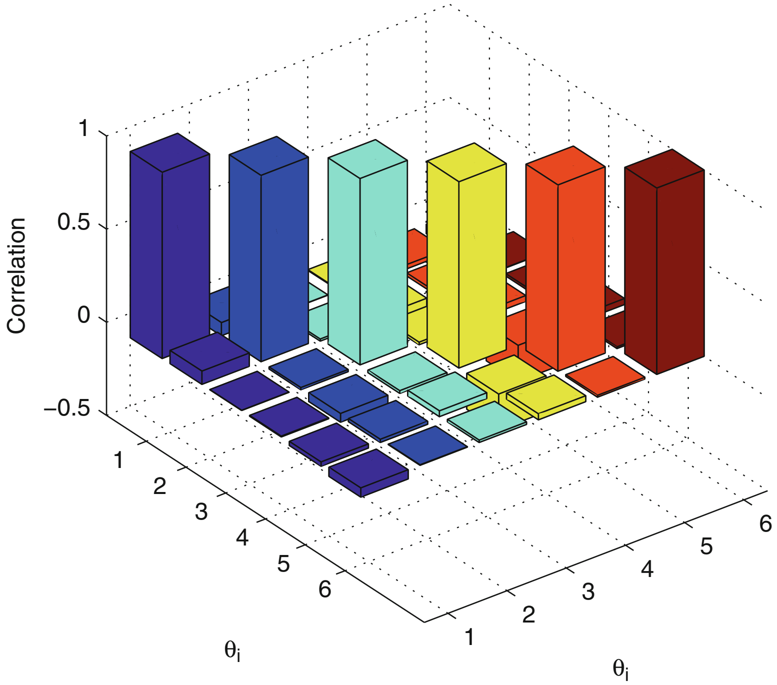

Fig. 26.3 The correlation between the updated parameters (M-H algorithm)

Table 26.1 shows the initial values of the parameters, the updated values and the coefficient of variation (c.o.v) of both EMCMC and M-H algorithms. All algorithms updated the $\boldsymbol{\theta}$ vector where the updated values are different than the initial $\boldsymbol{\theta}_0$. The updated vectors obtained by both algorithms are physically realistic. The coefficients of variance obtained by M-H algorithm are large compared with those obtained by EMCMC algorithm. The reason for this is that large move steps were used for both M-H algorithm to ensure a fast convergence and to improve the total average errors, while the EMCMC algorithm uses the mutation and the crossover operators to move from one sample to the next.

Figures 26.2 and 26.3 show the correlation between all updated parameters for both algorithms. The figures indicate that all parameters are correlated for all algorithms (all values are not 0), where most of the parameters are weakly correlated except the pair (A_{x1}, A_{x3}) which is found to be correlated (the correlation is 0.51) by the EMCMC algorithm.

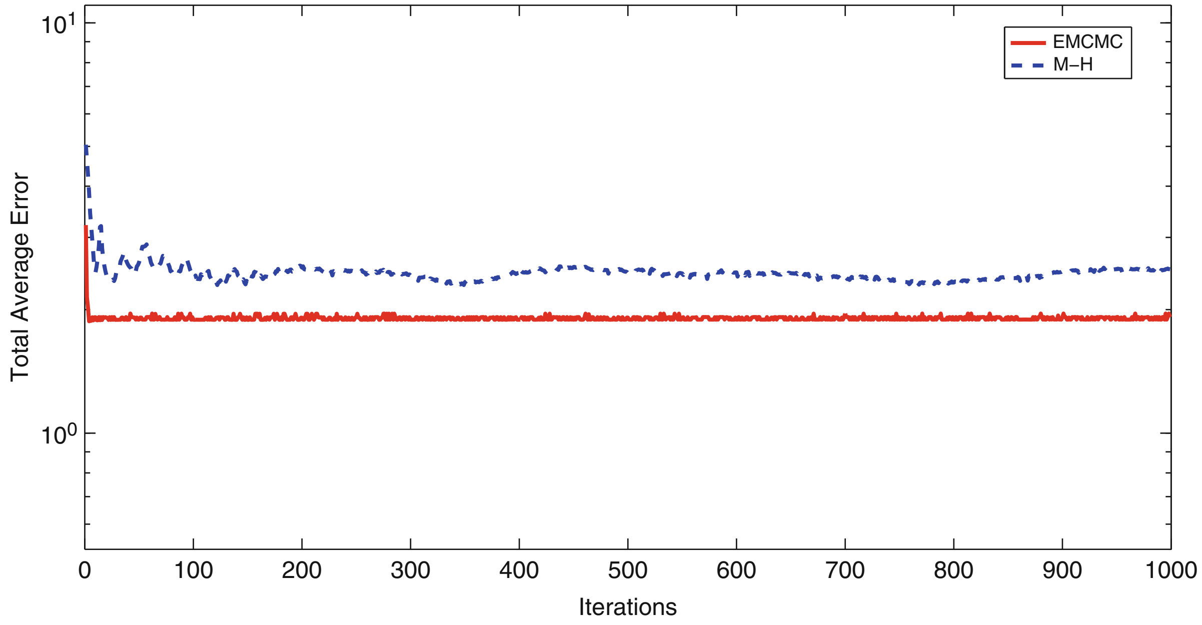

Fig. 26.4 The total average error using HMC, SS and M-H methods

Table 26.2 shows the absolute mode errors, the total average errors, the FEM updated frequencies and the coefficient of variations in brackets when both EMCMC and M-H algorithms are implemented to update the structure. The absolute mode error is given by $\frac{|f_i^m - f_i|}{f_i^m}$, while the Total Average Error (TAE) is given by: TAE $= \frac{1}{N_m}\sum_{i=1}^{N_m} \frac{\left|f_i^m - f_i\right|}{f_i^m}$. The error between the first measured natural frequency and that of the initial model was 4.63 %. When applying EMCMC algorithm this error was reduced to 1.49 %; however the M-H algorithm reduced the error to 0.04 %. The overall updated FEM natural frequencies for both algorithms are better than the initial FEM. However, The EMCMC algorithms produced better total average error results than the M-H algorithm. The updating procedure using the EMCMC method improved the error from 4.7 % to 1.92 %. The c.o.v values obtained by the EMCMC algorithm are very small compared to those obtained by the M-H algorithm which indicates that the EMCMC algorithms produces more accurate results than the M-H algorithm.

Figure 26.4 shows the absolute total average error for the three algorithms for 1,000 iterations. The y-axis (Total Average Error) is plotted by using the base 10 logarithmic scale. To obtain Fig. 26.4, the samples previously accepted are used to compute the mean value of samples which is given by: $\hat{\boldsymbol{\theta}} = E(\boldsymbol{\theta}) \cong \frac{1}{N_s}\sum_{j=1}^{i} \boldsymbol{\theta}^i$ where i represents the current iteration. Then, the absolute total average error is computed according to TAE (i) $= \frac{1}{N_m}\sum_{j=1}^{N_m} \frac{\left|f_j^m - f_j\right|}{f_j^m}$. The results obtained show that both algorithms converge fast (within the first 100 iterations for the M-H algorithm while the EMCMC algorithm uses less than 50 iteration to start converging). The reason that the M-H algorithm has a high convergence rate is that large move steps are used in these algorithms. However, the mutation and the crossover procedures helped the EMCMC algorithm to converge faster. The figure also shows that the error obtained by the EMCMC algorithm is smaller than that obtained by the M-H algorithm.

26.5 Conclusion

This paper presented the applicability and performance of the Evolutionary MCMC algorithm on the finite element model updating problem. We briefly stated the limitation of the initial finite element model and the need to update it. We then explained the origins of the EMCMC algorithm; it is a combination of the characteristics found in evolution – type algorithms (e.g. Genetic algorithm) and the dynamics of Markov chain Monte Carlo sampling methods. The resulting EMCMC is then posed in the Bayesian context where the objective function is a probability distribution. This is advantageous as this allows for the different variable distributions and samples to be calculated in a formal way.

The EMCMC algorithm was tested on a real beam structure with a number of uncertain variables. This method was compared with a classic MCMC algorithm; the Metropolis-Hasting. The structural example shows the efficiency of the EMCMC approach to sampling over the M-H algorithm methods. The M-H algorithm performance decreases with the complexity of the system and the size of the uncertain vector. As results, the total average error of the M-H algorithm was relatively large compared to that obtained by the EMCMC algorithm. Moreover the acceptance rate of the M-H algorithm was small. Future work will introduce further operators to this algorithm and the results obtained will be compared with those of other sampling techniques. To get further incite into the performance characteristics (e.g. stability and consistency) of these algorithms and their combinations; they will be tested on significantly more complex structures.

References

1. Marwala T (2010) Finite element model updating using computational intelligence techniques. Springer Verlag, London
2. Friswell MI, Mottershead JE (1995) Finite element model updating in structural dynamics. Kluwer, Dordrecht
3. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
4. Boulkaibet I, Marwala T, Mthembu L, Friswell MI, Adhikari S (2012) Sampling techniques in Bayesian finite element model updating. Proc Soc Exp Mech 29:75–83
5. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Boston
6. Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, MIT Press, Cambridge, MA
7. Mitchell M (1996) An introduction to genetic algorithms. MIT Press, Cambridge, MA
8. Davis L (1991) Handbook of genetic algorithms. Van Nostrand Reinhold, New York
9. Liang F, Wong WH (2000) Evolutionary Monte Carlo: application to Cp model sampling and change point problem. Stat Sin 10:317–342
10. Hukushima K, Nemoto K (1996) Exchange Monte Carlo method and application to spin glass simulations. J Physical Soc Japan 65:1604–1608
11. Yuen KV (2010) Bayesian methods for structural dynamics and civil engineering. Wiley, New York
12. Boulkaibet I, Mthembu L, Marwala T, Friswell MI, Adhikari S (2014) Finite element model updating using the shadow hybrid Monte Carlo technique, Mech Syst Signal Process, 52:115–132
13. Boulkaibet I, Mthembu L, Marwala T, Friswell MI, Adhikari S (2014) Finite element model updating using Hamiltonian Monte Carlo techniques. J Sound Vib (submitted for publication)
14. Marwala T (2010) Finite element model updating using computational intelligence techniques. Springer, London
15. Marwala T, Sibisi S (2005) Finite element model updating using Bayesian approach. In: Proceedings of the international modal analysis conference, Orlando. ISBN: 0-912053-89-5
16. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
17. Ching J, Leu SS (2009) Bayesian updating of reliability of civil infrastructure facilities based on condition-state data and fault-tree model. Reliab Eng Syst Saf 94(12):1962–1974
18. Chatterjee S, Carrera C, Lynch LA (1996) Genetic algorithms and traveling salesman problems. Eur J Oper Res 93:490–510
19. Liang F, Wong WH (2001) Real parameter evolutionary Monte Carlo with applications in Bayesian mixture models. J Am Stat Assoc 96:653–666
20. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equations of state calculations by fast computing machines. J Chem Phys 21:1087–1092
21. Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57(1):97–109
22. Geyer CJ (1991) Markov chain Monte Carlo maximum likelihood. In: Keramigas EM (ed) Computing science and statistics: proceedings of the 23rd symposium on the interface. Interface Foundations, Fairfax, pp 156–163
23. Cheung SH, Beck JL (2009) Bayesian model updating using hybrid Monte Carlo simulation with application to structural dynamic models with many uncertain parameters. J Eng Mech 135(4):243–255