

A DIRECT HAMILTONIAN MCMC APPROACH FOR RELIABILITY ESTIMATION

Hamed Nikbakht¹, and Konstantinos G. Papakonstantinou¹

¹Department of Civil and Environmental Engineering
The Pennsylvania State University, University Park, PA 16802, USA
e-mail: {hun35, kpapakon}@psu.edu

Abstract. *Accurate and efficient estimation of rare events probabilities is of significant importance, since often the occurrences of such events have widespread impacts. The focus in this work is on precisely quantifying these probabilities, often encountered in reliability analysis of complex engineering systems, by introducing a gradient-based Hamiltonian Markov Chain Monte Carlo (HMCMC) framework, termed Approximate Sampling Target with Post-processing Adjustment (ASTPA). The basic idea is to construct a relevant target distribution by weighting the high-dimensional random variable space through a one-dimensional likelihood model, using the limit-state function. To sample from this target distribution we utilize HMCMC algorithms that produce Markov chain samples based on Hamiltonian dynamics rather than random walks. We compare the performance of typical HMCMC scheme with our newly developed Quasi-Newton based mass preconditioned HMCMC algorithm that can sample very adeptly, particularly in difficult cases with high-dimensionality and very small failure probabilities. To eventually compute the probability of interest, an original post-sampling step is devised at this stage, using an inverse importance sampling procedure based on the samples. The involved user-defined parameters of ASTPA are then discussed and general default values are suggested. Finally, the performance of the proposed methodology is examined in detail and compared against Subset Simulation in a series of static and dynamic low- and high-dimensional benchmark problems.*

Keywords: Hamiltonian MCMC, Quasi-Newton, Rare Event Probability, High-dimensional Parameter Space, Reliability Estimation.

1 INTRODUCTION

In this work, we investigate Hamiltonian Markov Chain Monte Carlo (HMC MC) schemes for estimation of rare events probabilities, a commonly encountered important problem in several engineering and scientific applications, most often observed in the form of failure probability, or alternatively, reliability estimation. Calculating such small probabilities with accuracy presents many numerical and mathematical challenges, particularly in cases with high dimensional random spaces and/or expensive computational models, that practically limit the afforded number of model calls. The well known gradient based First Order Reliability Method (FORM), and variants, have a very long history in reliability estimation problems, with numerous successes [1, 2, 3, 4]. Such asymptotic approximation methods naturally have of course limitations, however, in general settings. Hence, numerous sampling based methods have been also suggested in the literature to tackle the problem in its utmost generality, e.g. [5]. The current state-of-the-art sampling method for problems of this type is termed Subset Simulation (SuS) [6] and belongs to the family of MCMC techniques. Within the context of Subset Simulation, various random-walk and non-random-walk-based MCMC proposal steps [7, 8] have been explored and suggested, to improve the sampling efficiency of SuS, including Hamiltonian steps [9].

In this work we completely deviate from SuS and we introduce a gradient-based Hamiltonian Markov Chain Monte Carlo (HMC MC) sampling framework, termed *Approximate Sampling Target with Post-processing Adjustment* (ASTPA) [10], that is directly used for rare events probabilities estimation. The basic idea of ASTPA is to construct a relevant target distribution to sample from, by weighting the high-dimensional random variable space through a one-dimensional likelihood model, using the limit-state function, and to then utilize an original post-sampling step, using an inverse importance sampling procedure based on the acquired samples. Hamiltonian MCMC schemes are employed to perform the sampling. The Hamiltonian Monte Carlo (HMC) method, originally developed by [11], and more recently popularized mainly through the works of [12, 13, 14], is characterized by scalability [13, 15], fast mixing rates, weak sample auto-correlation, even in complex high-dimensional parameter spaces [16], and has achieved broad-spectrum successes in most general settings e.g. [17, 18, 19, 20]. Herein, we compare the performance of the typical HMC MC scheme with our newly developed Quasi-Newton based mass preconditioned HMC MC algorithm that also exploits the information about the localized geometry of the failure region, through an inexpensive BFGS approximation. The involved user-defined parameters of ASTPA are also discussed in the paper and general default values are suggested. The performance of the proposed methodology is finally examined and compared successfully against Subset Simulation, in a series of static and dynamic, low- and high-dimensional benchmark problems.

2 CONCEPTS BEHIND HAMILTONIAN MARKOV CHAIN MONTE CARLO

In HMC MC methods, Hamiltonian dynamics are used to produce distant state steps for the Metropolis proposals, thereby avoiding the slow exploration of the state space that results from the diffusive behavior of simple random-walk proposals. Given a parameter of interest θ with (unnormalized) density $\pi_\theta(\cdot)$, the Hamiltonian Markov Chain Monte Carlo method introduces an auxiliary momentum variable $\mathbf{z}$ and samples from the joint distribution characterized by:

$$\pi(\theta, \mathbf{z}) \propto \pi_\theta(\theta) \pi_{\mathbf{z}|\theta}(\mathbf{z}|\theta) \quad (1)$$

where $\pi_{Z|\Theta}(\cdot|\boldsymbol{\theta})$ is proposed to be a symmetric distribution. With $\pi_{\Theta}(\boldsymbol{\theta})$ and $\pi_{Z|\Theta}(\mathbf{z}|\boldsymbol{\theta})$ being uniquely described up to normalizing constants, the functions $U(\boldsymbol{\theta}) = -\log \pi_{\Theta}(\boldsymbol{\theta})$ and $K(\boldsymbol{\theta}, \mathbf{z}) = -\log \pi_{Z|\Theta}(\mathbf{z}|\boldsymbol{\theta})$ are introduced as the potential energy and kinetic energy, owing to the physical laws which motivate the Hamiltonian Markov Chain Monte Carlo algorithm. The total energy $H(\boldsymbol{\theta}, \mathbf{z})$ can be thus expressed as:

$$H(\boldsymbol{\theta}, \mathbf{z}) = U(\boldsymbol{\theta}) + K(\boldsymbol{\theta}, \mathbf{z}) \quad (2)$$

and is often termed the Hamiltonian H . The kinetic energy function is unconstrained and can be formed in various ways based on the implementation. In most typical cases, the momentum is given by a zero-mean normal distribution [13, 16], and accordingly the kinetic energy can be written as: $K(\boldsymbol{\theta}, \mathbf{z}) = -\log \pi_{Z|\Theta}(\mathbf{z}|\boldsymbol{\theta}) = -\log \pi_Z(\mathbf{z}) = \frac{1}{2}\mathbf{z}^T \mathbf{M}^{-1}\mathbf{z}$, where the $\mathbf{M}$ is a symmetric, positive-definite covariance (mass) matrix.

HMCMC generates a Metropolis proposal on the joint state-space $(\boldsymbol{\theta}, \mathbf{z})$ by sampling the momentum and simulating trajectories of Hamiltonian dynamics in which the time evolution of the state $(\boldsymbol{\theta}, \mathbf{z})$ is governed by Hamilton's equations, expressed typically by:

$$\frac{d\boldsymbol{\theta}}{dt} = \frac{\partial H}{\partial \mathbf{z}} = \frac{\partial K}{\partial \mathbf{z}} = \mathbf{M}^{-1}\mathbf{z}, \quad \frac{d\mathbf{z}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\theta}} = -\frac{\partial U}{\partial \boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \quad (3)$$

where $\mathcal{L}(\boldsymbol{\theta})$ denotes the log-density of the target distribution. Hamiltonian dynamics prove to be an effective proposal generation mechanism because the distribution $\pi(\boldsymbol{\theta}, \mathbf{z})$ is invariant under the dynamics of Eq. (3). These dynamics enable a proposal state, obtained by an approximate solution of Eq. (3), to be distant from the current state, yet having high probability of acceptance. The solution to Eq. (3) is in general analytically intractable and thus the Hamiltonian equations need to be numerically solved by discretizing time, using some small step size, ε . A symplectic integrator that can be used for the numerical solution is the leapfrog one, as follows:

$$\mathbf{z}_{t+\varepsilon/2} = \mathbf{z}_t - \left(\frac{\varepsilon}{2}\right) \frac{\partial U}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_t), \quad \boldsymbol{\theta}_{t+\varepsilon} = \boldsymbol{\theta}_t + \varepsilon \frac{\partial K}{\partial \mathbf{z}}(\mathbf{z}_{t+\varepsilon/2}), \quad \mathbf{z}_{t+\varepsilon} = \mathbf{z}_{t+\varepsilon/2} - \left(\frac{\varepsilon}{2}\right) \frac{\partial U}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_{t+\varepsilon}) \quad (4)$$

The main advantages of using the leapfrog integrator are its simplicity, its volume-preserving feature, and its reversibility, due to its symmetry, by simply negating $\mathbf{z}$, facilitating a valid Metropolis proposal. See [13], [16] and [21] for details on energy-conservation, reversibility and volume-preserving integrators and their connections to HMCMC. It is noted that in the above leapfrog integration algorithm, the computationally expensive part is to acquire the $\frac{\partial U}{\partial \boldsymbol{\theta}}$ term at the updated location $\boldsymbol{\theta}$. Taking $L = \tau/\varepsilon$ steps of the leapfrog integrator approximates the evolution $(\boldsymbol{\theta}(0), \mathbf{z}(0)) \rightarrow (\boldsymbol{\theta}(\tau), \mathbf{z}(\tau))$, where τ is the trajectory length or path length, and provides the exact solution in the limit $\varepsilon \rightarrow 0$.

As discussed, the typical HMCMC version is based on a Gaussian momentum $\pi_{Z|\Theta}(\mathbf{z}|\boldsymbol{\theta}) = \pi_Z(\mathbf{z}) \sim N(\mathbf{0}, \mathbf{M})$ (or $\mathbf{z} \sim N(\mathbf{0}, \mathbf{M})$). The mass matrix $\mathbf{M}$ is often set to the identity matrix, $\mathbf{I}$, but can also be adapted to precondition the sampler when relevant information about the target distribution is available (see Section 4). A standard procedure for drawing N_{iter} samples via HMCMC is described in Algorithm 1, where $\mathcal{L}(\boldsymbol{\theta})$ is the log-density of the target distribution of interest. $\boldsymbol{\theta}^0$ are the initial values for the $\boldsymbol{\theta}$, and L is the number of leapfrog steps, as explained before. For each HMCMC step, we first resample the momentum and then implement the L leapfrog updates (Leapfrog($\tilde{\boldsymbol{\theta}}, \tilde{\mathbf{z}}, \varepsilon$)) before we accept or reject the Metropolis proposal at the pertinent step.

Algorithm 1 Hamiltonian Markov Chain Monte Carlo

```

1: procedure HMCMC( $\theta^0, \varepsilon, L, \mathcal{L}(\theta), N_{\text{Iter}}$ )
2:   for  $m = 1$  to  $N_{\text{Iter}}$  do
3:      $\mathbf{z}^0 \sim N(0, \mathbf{I})$  ▷ momentum sampling from standard normal distribution
4:      $\theta^m \leftarrow \theta^{m-1}, \tilde{\theta} \leftarrow \theta^{m-1}, \tilde{\mathbf{z}} \leftarrow \mathbf{z}^0$ 
5:     for  $i = 1$  to  $L$  do
6:        $\tilde{\theta}, \tilde{\mathbf{z}} \leftarrow \text{Leapfrog}(\tilde{\theta}, \tilde{\mathbf{z}}, \varepsilon)$  ▷ leapfrog integration
7:     end for
8:     with probability:
9:        $\alpha = \min \left\{ 1, \frac{\exp(\mathcal{L}(\tilde{\theta}) - \frac{1}{2}\tilde{\mathbf{z}} \cdot \tilde{\mathbf{z}})}{\exp(\mathcal{L}(\theta^{m-1}) - \frac{1}{2}\mathbf{z}^0 \cdot \mathbf{z}^0)} \right\}$  ▷ Metropolis step
10:       $\theta^m \leftarrow \tilde{\theta}, \mathbf{z}^m \leftarrow -\tilde{\mathbf{z}}$ 
11:    end for
12: end procedure

```

The efficiency of HMC_{MC} relies significantly on selecting suitable values for ε and L . In this work we select the stepsizes ε in such a way that the corresponding average acceptance rates are approximately 65%, as values between 60% and 80% are typically assumed optimal [13, 14, 15]. The dual averaging algorithm of [14] was adopted here to find these stepsizes. However, in contrast to [14] we adapt these stepsizes throughout the analysis, in both the burn-in and stationary phases of the MCMC algorithm. To determine the value of L , we estimate the trajectory length τ so as to have a sufficient so called normalized Expected Square Jumping Distance (*ESJD*) $\tau^{-1/2} \mathbb{E} \|\theta^{(t+1)}(\tau) - \theta^{(t)}(\tau)\|^2$, as introduced in [22], and then we randomly perturb each trajectory length $\tau^{(t)}$ in the range $[0.9\tau, 1.1\tau]$ to avoid periodicity (t denotes the t -th iteration of HMC_{MC}). In all our experiments we determine L and control the trajectory length in this manner, as we have found it to work well in practice. The role of these parameters (ε and τ (or L)) and techniques for determining them have been quite extensively studied and for more details we refer the readers to [13, 14, 15].

3 METHODOLOGY TO CALCULATE THE FAILURE PROBABILITY

The failure probability P_F for a system, that is the probability of a defined unacceptable system performance, can be expressed as a d -fold integral, as:

$$P_F = \mathbb{E}[I_F(\theta)] = \int_{g(\theta) \leq 0} I_F(\theta) \pi_{\theta}(\theta) d\theta \quad (5)$$

where θ is the random vector $[\theta_1, \dots, \theta_d]^T$; $F \subset \mathbb{R}^d$ is the failure event in the parameter space; $g(\theta)$ is the limit-state function that can include one or several distinct failure modes and defines the failure of the system by $g(\theta) \leq 0$; $I(\cdot)$ is the indicator function with: $I_F(\theta) = 1$ if $\theta \in g(\theta) \leq 0$ and $I_F(\theta) = 0$ otherwise; $\mathbb{E}$ is the expectation operator, and π_{θ} is the joint probability density function (PDF) for Θ . It is common practice in reliability analysis to have the joint PDF of Θ be the standard normal one, due to its rotational symmetry and exponential probability decay. In most cases, this is not restrictive, since it is uncomplicated to transform the original random variables $\mathbf{X}$ to Θ , e.g. [23]. When this is not the case however, but the probabilistic characterization of $\mathbf{X}$ can be defined in terms of marginal distributions and correlations, the Nataf

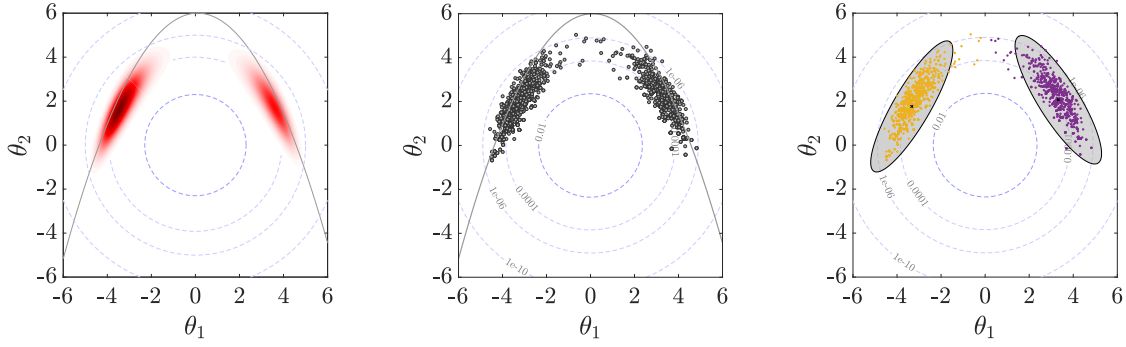


Figure 1: The above figures represent the analytical target distribution, the simulated target distribution samples based on our HMC-MC-based method, and the fitted Gaussian Mixture Model describing the simulated samples, from left to right, respectively.

distribution (equivalent to Gaussian copula) can be used to model the joint PDF, and the mapping to the standard normal space can be then accomplished [24].

The main idea of our approach to calculate the failure probability is to construct an appropriate approximate target distribution to sample from, based on Hamiltonian MCMC methods that can quickly reach regions of interest and can keep the number of model calls to a minimum, and to then utilize a post-sampling step to acquire the exact probability estimation, without any additional model calls. We construct this approximate target distribution by combining the multidimensional parameter space Θ with a one-dimensional likelihood function, using the limit-state expression. This one-dimensional likelihood function is expressed as a Gaussian PDF with mean $= \mu_{g(\theta)} = 0$, where $g(\theta)$ is the limit-state function, and a dispersion factor σ :

$$\mathbf{N}\left(\frac{g(\theta)}{g_c} \mid \mu_{g(\theta)} = 0, \sigma\right), \quad g_c = \begin{cases} g(\mathbf{0}), & \text{if } g(\mathbf{0}) > 8 \text{ or } g(\mathbf{0}) < 1 \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

where g_c is a normalizing constant. The reason for this normalization, $g(\theta)/g_c$, is to control the suggested upper and lower bounds of σ . The target PDF is then defined as:

$$\text{Target probability distribution} \propto \mathbf{N}\left(\frac{g(\theta)}{g_c} \mid \mu_{g(\theta)} = 0, \sigma\right) \times \left(\theta \sim \mathbf{N}(\mathbf{0}, \mathbf{I})\right) \quad (7)$$

Having the total number of model calls in mind, as well as the coefficient of variation of the estimator (C.O.V), the suggested value for σ is in the range $[0.1 \ 0.7]$. Fine tuning σ in that range is not generally necessary. It is recommended, in general, to use higher σ values ($0.6 - 0.7$) in nonlinear high-dimensional problems and multi-modal cases when $1 \leq g(\mathbf{0}) \leq 8$, since a larger σ usually allows longer state jumps and fewer required model calls. On the other hand, a lower σ generally increases the accuracy of the estimator, at the expense of a slightly increased number of model calls.

Fig. 1 concisely portrays the overall approach by using a bimodal target distribution. The gray curves represent the parabolic limit-state function $g(\theta)$ of this problem, with the failure domain being outside $g(\theta)$. The left figure displays the constructed target distribution, by adopting the previously described approach, which in this simple 2D case can be visualized. The middle figure shows drawn samples from the target distribution by our suggested Hamiltonian MCMC variant, described in Section 4. For their initial stage, our HMC-MC samplers have an adaptive annealed phase, mainly in order to automatically tune parameters and reduce the computational

cost, overall, and then follow the typical Hamiltonian approach, except in our Quasi-Newton case (Section 4) the mass matrix is appropriately preconditioned. As such, during the burn-in period, we initialize the spread of likelihood, σ_0 , equal to 1 that then follows an *exponential decay* throughout the burn-in period, while at the end of this initial period, σ takes its constant value, as described above, for the stationary phase of the algorithms.

To finally compute the failure probability we have to adjust Eq. (5) accordingly, since the samples have been sampled based on our constructed approximate target distribution. An original post-sampling step is devised at this stage using our inverse importance sampling procedure, i.e. having the samples, choose a pertinent Importance Sampling Density (ISD) automatically, based on the samples. Given that, the probability of failure after some algebra (see [10] for details) can be computed as follows:

$$P_F = \int I_F(\boldsymbol{\theta}) \pi_{\boldsymbol{\theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int I_F(\boldsymbol{\theta}) \frac{C \cdot \tilde{h}(\boldsymbol{\theta})}{\ell(\boldsymbol{\theta})} d\boldsymbol{\theta} \quad (8)$$

where $C = \frac{1}{N} \sum_{i=1}^N \frac{\tilde{h}(\theta_i)}{Q(\theta_i)}$; $\tilde{h}(\cdot)$ denotes the non-normalized target PDF, $\ell(\boldsymbol{\theta})$ is our likelihood function, and $Q(\cdot)$ is a computed Gaussian Mixture Model (GMM), based on the already available samples and the generic Expectation Maximization (EM) algorithm, as indicatively seen in the right plot of Fig. 1.

Our described newly proposed method is termed ASTPA (Approximate Sampling Target with Post-processing Adjustment) and, as a summary, comprises of constructing a target distribution model, performing HMCMC sampling, and finally applying a post-sampling step. For more details on supplementary justifications about this method, we refer readers to [10].

4 QUASI-NEWTON EXTENSIONS AND CONNECTIONS TO HMCMC

In high-dimensional problems, the computational cost of the typical HMCMC sampler may increase considerably and a prohibitive number of model calls per leapfrog step may be required. In this work, we address this issue in a developed Newton-type context, where the Hessian information is approximated without any required additional model calls per leapfrog step. To this end, the well-known BFGS approximation [25] is used in our Quasi-Newton type Hamiltonian MCMC approach. Let $\boldsymbol{\theta} \in \mathbb{R}^d$, consistent with the previous section. Given the k -th estimate $\mathbf{W}_k$, where $\mathbf{W}_k$ is an approximation to the inverse Hessian at $\boldsymbol{\theta}_k$, the BFGS update $\mathbf{W}_{k+1}$ can be expressed as:

$$\mathbf{W}_{k+1} = (\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}) \mathbf{W}_k (\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \quad (9)$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{s}_k = \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k$, and $\mathbf{y}_k = \nabla f(\boldsymbol{\theta}_{k+1}) - \nabla f(\boldsymbol{\theta}_k)$ where $f : \mathbb{R}^d \rightarrow \mathbb{R}$ denotes any relevant target distribution function in this case. Our developed Quasi-Newton preconditioned Hamiltonian Markov Chain Monte Carlo (QNp-HMCMC) method is presented in detail in Algorithm 2. In the burn-in phase we are still sampling the momentum from an identity mass matrix but the ODEs of Eq. (3) now become:

$$\dot{\boldsymbol{\theta}} = \mathbf{W} \mathbf{M}^{-1} \mathbf{z}, \quad \dot{\mathbf{z}} = \mathbf{W} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}). \quad (10)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the symmetric positive definite matrix of Eq. (9) and being the inverse Hessian matrix provides an informed approximation of the local geometry of the parameter space,

accelerating exploration of the domain. The final estimation of the approximated inverse of the Hessian matrix, $\mathbf{W}$, from the burn-in phase is then used to define the preconditioned covariance matrix to sample the momentum variable for the stationary, non-adaptive stage of the chain. It can be shown that all utilized dynamics in both phases of the algorithm enable us to maintain the desired target distribution as the invariant one. In Section 5 we empirically evaluate and compare the QNp-HMCMC performance in various settings. For further details on the QNp-HMCMC method, its performance in different settings, and its validity, see [10].

5 NUMERICAL RESULTS

In this section, four numerical examples are implemented to illustrate the efficiency of the proposed methods. In all examples, the tuning parameters $(\varepsilon, \tau, \sigma)$ are systematically used as mentioned in Sections 2 and 3. In the context of reliability problems, we use the default value $\tau = 0.7$ as a starting point and then employ the ESJD metric [22] as described in Section 2. The burn-in period is chosen to be on average 15% of the total number of model calls, while the upper bound of the burn-in size is limited to 20%. The described methods are compared to the Component-wise Metropolis-Hastings based Subset Simulation (CWMH-SuS). For the sake of comparison, we use two proposal distributions in CWMH-SuS, a uniform distribution of width 2 and a standard normal one. The parameters of Subset Simulation are chosen as $n_s = 1,000$ and 2,000 for low- and high-dimensional simulations respectively, where n_s is the number of samples for each subset level, and $p_0 = 0.1$, where p_0 is the percentile of the samples that determines the intermediate subsets [6]. Comparisons are illustrated in terms of accuracy and computational cost. In particular, the tables show the P_F estimation, including the mean number of limit-state function calls in order to calculate the value and gradient of the target distribution in the HMCMC-based algorithms, and the value of the limit-state function in SuS. The analytical gradients are provided in all examples, hence one model call can provide both the value and the gradient of the target distribution. In all examples, the number of limit-state function evaluations for all methods has been set to be roughly the same to each other for comparison purposes. Results are based for all examples on 500 independently performed simulations, so that the sample mean and C.O.V of the results can be acquired. It should be noted that the ASTPA parameters are carefully chosen for all examples but are not optimized for any one. Hence, comparative and perhaps improved alternate performance might be achieved with a different set of parameters.

5.1 Example 1: parabolic/concave limit-state function

The first example is expressed by the following limit state function for two standard normal random variables [26]:

$$g(\theta) = r - \theta_2 - \kappa (\theta_1 - e)^2 \quad (11)$$

where r , κ and e are deterministic parameters chosen as $r = 6$, $\kappa = 0.3$ and $e = 0.1$. The probability of failure is 3.95E-5 and the limit-state function consists of two design points (failure modes), as seen in Fig. 1. For the HMCMC-based algorithms, the likelihood dispersion factor, σ , is 0.7 and the burn-in sample size is taken as 200. Consistent to the discussion in Section 3, the trajectory length is set to $\tau = 1$.

Table 1 compares the number of model calls, the coefficient of variation and the $\mathbb{E}[\hat{P}_F]$ obtained by all tested methods. The Subset Simulation results are based on $n_s = 1,000$. It is shown that the HMCMC approach gives significantly smaller C.O.V. than SuS and also

Algorithm 2 Quasi-Newton preconditioned Hamiltonian Markov Chain Monte Carlo

```

1: procedure QNP-HMCMC( $\theta^0, \varepsilon, L, \mathcal{L}(\theta), BurnIn, NIter$ )
2:    $\mathbf{W} = \mathbf{I}$ 
3:   for  $m = 1$  to  $NIter$  do
4:     if  $m \leq BurnIn$  then
5:        $\mathbf{z}^0 \sim \mathbf{N}(\mathbf{0}, \mathbf{M})$  ▷ where  $\mathbf{M} = \mathbf{I}$ 
6:        $\theta^m \leftarrow \theta^{m-1}, \tilde{\theta} \leftarrow \theta^{m-1}, \tilde{\mathbf{z}} \leftarrow \mathbf{z}^0, \mathbf{B} \leftarrow \mathbf{W}$ 
7:       for  $i = 1$  to  $L$  do
8:          $\tilde{\theta}, \tilde{\mathbf{z}} \leftarrow \text{Leapfrog-BurnIn}(\tilde{\theta}, \tilde{\mathbf{z}}, \varepsilon, \mathbf{B})$ 
9:         Update  $\mathbf{W}$  using Eq. (9)
10:      end for
11:      with probability:
12:      
$$\alpha = \min \left\{ 1, \frac{\exp(\mathcal{L}(\tilde{\theta}) - \frac{1}{2}\tilde{\mathbf{z}}.\tilde{\mathbf{z}})}{\exp(\mathcal{L}(\theta^{m-1}) - \frac{1}{2}\mathbf{z}^0.\mathbf{z}^0)} \right\}$$

13:       $\theta^m \leftarrow \tilde{\theta}, \mathbf{z}^m \leftarrow -\tilde{\mathbf{z}}$  ▷ If proposal rejected:  $\mathbf{W} \leftarrow \mathbf{B}$ 
14:    else ▷ If  $m > BurnIn$ 
15:       $\mathbf{z}^0 \sim \mathbf{N}(\mathbf{0}, \mathbf{M})$  ▷ where  $\mathbf{M} = \mathbf{W}^{-1}$ 
16:       $\theta^m \leftarrow \theta^{m-1}, \tilde{\theta} \leftarrow \theta^{m-1}, \tilde{\mathbf{z}} \leftarrow \mathbf{z}^0$ 
17:      for  $i = 1$  to  $L$  do
18:         $\tilde{\theta}, \tilde{\mathbf{z}} \leftarrow \text{Leapfrog}(\tilde{\theta}, \tilde{\mathbf{z}}, \varepsilon, \mathbf{M})$ 
19:      end for
20:      with probability:
21:      
$$\alpha = \min \left\{ 1, \frac{\exp(\mathcal{L}(\tilde{\theta}) - \frac{1}{2}\tilde{\mathbf{z}}.\mathbf{M}^{-1}.\tilde{\mathbf{z}})}{\exp(\mathcal{L}(\theta^{m-1}) - \frac{1}{2}\mathbf{z}^0.\mathbf{M}^{-1}.\mathbf{z}^0)} \right\}$$

22:       $\theta^m \leftarrow \tilde{\theta}, \mathbf{z}^m \leftarrow -\tilde{\mathbf{z}}$ 
23:    end if
24:  end for
25: end procedure
26:
27: function LEAPFROG-BURNIN( $\tilde{\theta}, \tilde{\mathbf{z}}, \varepsilon, \mathbf{B}$ )
28:    $\tilde{\mathbf{z}} \leftarrow \mathbf{z} + (\varepsilon/2)\mathbf{B}\nabla_{\theta}\mathcal{L}(\theta)$ 
29:    $\tilde{\theta} \leftarrow \theta + \varepsilon\mathbf{B}\tilde{\mathbf{z}}$ 
30:    $\tilde{\mathbf{z}} \leftarrow \mathbf{z} + (\varepsilon/2)\mathbf{B}\nabla_{\theta}\mathcal{L}(\tilde{\theta})$ 
31: return  $\tilde{\theta}, \tilde{\mathbf{z}}$ .
32: end function
33:
34: function LEAPFROG( $\tilde{\theta}, \tilde{\mathbf{z}}, \varepsilon, \mathbf{M}$ )
35:    $\tilde{\mathbf{z}} \leftarrow \mathbf{z} + (\varepsilon/2)\nabla_{\theta}\mathcal{L}(\theta)$ 
36:    $\tilde{\theta} \leftarrow \theta + \varepsilon\mathbf{M}^{-1}\tilde{\mathbf{z}}$ 
37:    $\tilde{\mathbf{z}} \leftarrow \mathbf{z} + (\varepsilon/2)\nabla_{\theta}\mathcal{L}(\tilde{\theta})$ 
38: return  $\tilde{\theta}, \tilde{\mathbf{z}}$ .
39: end function

```

Table 1: Performance of various methods for the parabolic/concave limit-state function

500 Independent Simulations		CWMH-SuS		HMCMC	QNp-HMCMC
		$U(-1, 1)$	$N(0, 1)$		
$\sigma = 0.7$ $\tau = 1$	Number of model calls	4,559	4,565	4,391	4,926
	C.O.V	0.62	0.65	0.35	0.39
	$\mathbb{E}[\hat{P}_F]$ (Exact $P_F \sim 3.95\text{E-}5$)	4.19E-5	4.14E-5	3.86E-5	3.47E-5

Table 2: Performance of various methods for the four-branch series system

500 Independent Simulations		CWMH-SuS		HMCMC	QNp-HMCMC
		$U(-1, 1)$	$N(0, 1)$		
$\sigma = 0.7$ $\tau = 1$ $n_s = 1,000$	Number of model calls	2,841	2,852	2,867	2,887
	C.O.V	0.26	0.30	0.29	0.26
	$\mathbb{E}[\hat{P}_F]$ (Exact $P_F \sim 2.20\text{E-}3$)	2.23E-3	2.26E-3	1.98E-3	1.91E-3
$\sigma = 0.7$ $\tau = 1$ $n_s = 2,000$	Number of model calls	5,634	5,657	5,688	5,740
	C.O.V	0.19	0.22	0.13	0.17
	$\mathbb{E}[\hat{P}_F]$ (Exact $P_F \sim 2.20\text{E-}3$)	2.24E-3	2.23E-3	2.16E-3	2.11E-3

outperforms it in terms of the $\mathbb{E}[\hat{P}_F]$. Fig. 1 also demonstrates that the QNp-HMCMC samples accurately describe the two important failure regions.

5.2 Example 2: four-branch series system

This example is a well-known benchmark system reliability problem, defined by the following limit-state function in the standard normal space:

$$g(\boldsymbol{\theta}) = \min \begin{cases} 3 + 0.1(\theta_1 - \theta_2)^2 - (\theta_1 - \theta_2)/\sqrt{2} \\ 3 + 0.1(\theta_1 - \theta_2)^2 + (\theta_1 - \theta_2)/\sqrt{2} \\ (7/\sqrt{2}) + (\theta_1 - \theta_2) \\ (7/\sqrt{2}) + (\theta_2 - \theta_1) \end{cases} \quad (12)$$

The trajectory length is chosen as $\tau = 1$ and the likelihood dispersion factor, σ , is fixed to 0.7. The burn-in is set to 200 samples. Table 2 shows that the SuS with uniform proposal gives more accurate P_F estimation with smaller C.O.V than the HMCMC-based methods for the case of $n_s = 1,000$. However, by increasing the sample size to $n_s = 2,000$, it is seen that the HMCMC algorithm exhibits lower C.O.V compared to both SuS implementations. For the case of QNp-HMCMC, both the bias and C.O.V of the probability estimate considerably decrease with the sample size increase. Fig. 2 shows the analytical target density of the four-branch limit-state function problem and samples from the target distribution using the HMCMC approach. As seen, the method achieves to efficiently sample all four important failure regions.

5.3 Example 3: SDOF oscillator under impulse load

In this example, a nonlinear undamped single-degree-of-freedom (SDOF) oscillator subjected to a rectangular impulse load is analysed, as described in [27, 28]. The limit-state function is given as:

$$g(k_1, k_2, M, r, T_1, F_1) = 3r - \left| \frac{2F_1}{M\omega_0^2} \sin\left(\frac{\omega_0 T_1}{2}\right) \right| \quad (13)$$

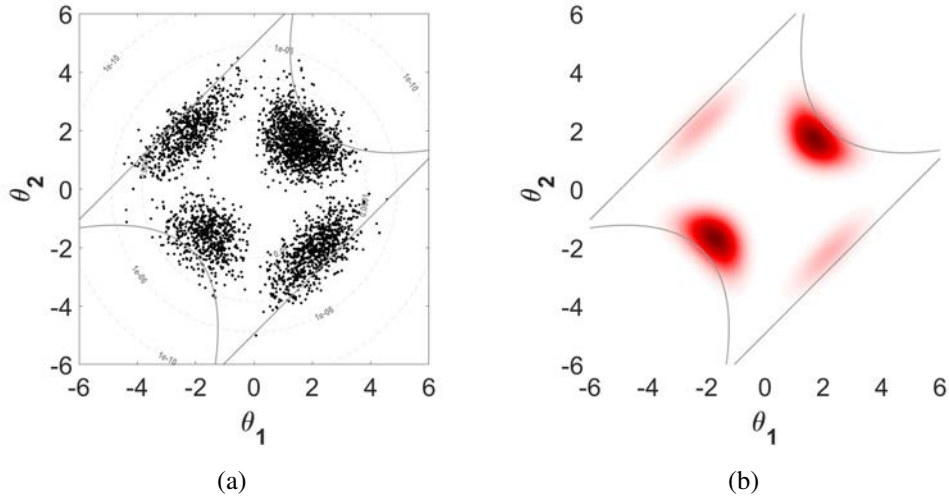


Figure 2: (a) Simulated samples from the target distribution, (b) Analytical target distribution.

Table 3: Random variables of the undamped oscillator

Variable	Distribution	Mean	C.O.V
M	Gaussian	1	0.05
k_1	Gaussian	1	0.1
k_2	Gaussian	0.1	0.1
r	Gaussian	0.5	0.1
T_1	Gaussian	1	0.2
F_1	Gaussian	0.6-0.45	$\frac{1}{6}$

Table 4: Performance of various methods for the undamped oscillator example

500 Independent Simulations			CWMH-SuS		HMCMC	QNp-HMCMC
			$U(-1, 1)$	$N(0, 1)$		
$\sigma = 0.1$	Number of model calls		5,170	5,160	5,132	5,119
$\tau = 0.7$	C.O.V		0.67	0.51	0.14	0.11
$\mu_{F_1} = 0.6$	$\mathbb{E}[\hat{P}_F]$	(Exact $P_F \sim 9.09\text{E-}6$)	9.68E-6	9.55E-6	9.10E-6	9.08E-6
$\sigma = 0.1$	Number of model calls		7,583	7,617	7,523	7,515
$\tau = 0.7$	C.O.V		0.77	0.70	0.21	0.15
$\mu_{F_1} = 0.45$	$\mathbb{E}[\hat{P}_F]$	(Exact $P_F \sim 1.55\text{E-}8$)	1.67E-8	1.50E-8	1.52E-8	1.51E-8

where $\omega_0 = \sqrt{(k_1 + k_2)/M}$ is the natural frequency of the oscillator, T_1 is the duration of the impulse load, M is the mass, k_1 and k_2 are the stiffnesses of the primary and secondary springs, r is the displacement at which one of the springs yields, and F_1 is the amplitude of the force. The description of all random variables is listed in Table 3. SuS results for both proposals are based on $n_s = 1,000$. All variables are first transformed to the standard normal space. Results are shown in Table 4 for two cases, by changing the mean value, μ_{F_1} , of F_1 . For the HMCMC-based methods, the trajectory length and the likelihood dispersion factor are chosen as $\tau = 0.7$ and $\sigma = 0.1$ respectively. The burn-in sample size is set to 500. It is shown in this example that the QNp-HMCMC approach provides significantly more accurate and stable results in terms of the C.O.V. and $\mathbb{E}[\hat{P}_F]$. Particularly for the lowest failure probability level, QNp-HMCMC

Table 5: Performance of various methods for SDOF oscillator under white noise

500 Independent Simulations		CWMH-SuS		HMC MC	QNp-HMC MC
		$U(-1, 1)$	$N(0, 1)$		
$\sigma = 0.2$	Number of model calls	11,000	11,011	11,063	11,059
$\tau = 0.9$	C.O.V	0.32	0.35	0.30	0.24
$R = 1.8$	$\mathbb{E}[\hat{P}_F]$ (Exact $P_F \sim 2.53\text{E-}6$)	2.58E-6	2.63E-6	2.57E-6	2.55E-6
$\sigma = 0.2$	Number of model calls	13,578	13,646	13,644	13,618
$\tau = 0.9$	C.O.V	0.41	0.48	0.34	0.29
$R = 2$	$\mathbb{E}[\hat{P}_F]$ (Exact $P_F \sim 1.11\text{E-}7$)	1.16E-7	1.14E-7	1.13E-7	1.12E-7

approach noticeably outperforms all other methods. As results indicate, the QNp-HMC MC method is roughly insensitive to the failure probability level and there is no negative influence on the method when changing μ_{F_1} . For the two SuS variants, it is noteworthy to say here that the SuS with the standard normal proposal distribution indicates reasonably better performance in this example than the one with the uniform proposal.

5.4 Example 4: SDOF oscillator under white noise excitation

In this last example, we consider a SDOF oscillator, initially at rest, with natural frequency $\omega = 7.85 \text{ rad/s}$ and damping ratio $\xi = 0.02$, subjected to a Gaussian white noise ($W(t)$) excitation with spectral density of magnitude $S_0 = 1$. The response of the system is computed at discrete time instants $\{t_j = (j - 1)\Delta t : j = 1, \dots, n\}$ with $\Delta t = 0.05$, and the duration of study is $T = 5 \text{ sec}$. Thus, the number of time instants is equal to $n = T/\Delta t + 1 = 101$. The state vector θ is the sequence of i.i.d. standard normal random variables that generate the $W(t_j) = \sqrt{\frac{2\pi S_0}{\Delta t}} \theta_j$ at the discrete time instants, resulting in 101 involved random variables in this example. Failure is characterized by the positive displacement response exceeding a threshold level R : $g(\theta) = R - \max\{Y(t)\}$.

The burn-in sample size is taken as 1,000 for the HMC MC-based methods. SuS results are based on $n_s = 2,000$. It is seen in Table 5 that the QNp-HMC MC approach shows more accurate and efficient results in terms of C.O.V. and $\mathbb{E}[\hat{P}_F]$. Compared to the HMC MC approach, this example agrees with the additional results in [10] and confirms that the application of QNp-HMC MC in high-dimensional reliability problems is in general more attractive. By decreasing the target failure probability, results also reveal that QNp-HMC MC gives us a substantially improved estimation in comparison to all other methods.

6 CONCLUSIONS

A novel approach for estimation of rare event probabilities termed Approximate Sampling Target with Post-processing Adjustment (ASTPA), is presented in this paper, suitable for low- and high-dimensional problems, very small probabilities and multiple failure modes. ASTPA can provide an accurate unbiased estimation of the failure probabilities with an efficient number of limit-state function evaluations. The basic idea of ASTPA is to construct a relevant target distribution by weighting the high-dimensional random variable space through a one-dimensional likelihood model, using the limit-state function. To sample from this target distribution we utilize gradient-based HMC MC schemes, including our newly developed Quasi-Newton based mass preconditioned HMC MC algorithm (QNp-HMC MC) that can sample very adeptly, particularly in difficult cases with high-dimensionality and very small failure probabilities. Finally, an

original post-sampling step is also devised, using an inverse importance sampling procedure based on the samples. The performance of the proposed methodology is examined and compared very successfully herein against Subset Simulation in a series of static and dynamic low- and high-dimensional benchmark problems. As a general guideline, QNp-HMCMC is recommended to be used for problems with more than 20 dimensions, where traditional HMCMC schemes may not perform that well. However, even in lower dimensions QNp-HMCMC performs reasonably well and is still a competitive algorithm. Since we are utilizing gradient-based sampling methods, all of our analyses and results are based on the fact that analytical gradients can be computed. In cases where numerical schemes are needed for the gradient evaluations, then HMCMC methods will not be competitive in relation to SuS. It should also be pointed out that different combinations of the HMCMC and QNp-HMCMC algorithms can be possible, based on problem-specific characteristics. Some of the ongoing and future work is directed towards exploring various ASTPA variants, and on estimating first-passage problems under numerous settings and high-dimensional parameter spaces.

REFERENCES

- [1] Rüdiger Rackwitz. Reliability analysis—A review and some perspectives. *Structural Safety*, **23**(4):365–395, 2001.
- [2] Armen Der Kiureghian. First-and second-order reliability methods. *Engineering Design Reliability Handbook*, **14**, 2005.
- [3] Pei-Ling Liu and Armen Der Kiureghian. Optimization algorithms for structural reliability. *Structural Safety*, **9**(3):161–177, 1991.
- [4] Karl Breitung. 40 years FORM: Some new aspects? *Probabilistic Engineering Mechanics*, **42**:71–77, 2015.
- [5] Gerhart I Schuëller and Helmuth J Pradlwarter. Benchmark study on reliability estimation in higher dimensions of structural systems—an overview. *Structural Safety*, **29**(3):167–182, 2007.
- [6] Siu-Kui Au and James L Beck. Estimation of small failure probabilities in high dimensions by Subset Simulation. *Probabilistic Engineering Mechanics*, **16**(4):263–277, 2001.
- [7] Iason Papaioannou, Wolfgang Betz, Kilian Zwirgmaier, and Daniel Straub. MCMC algorithms for Subset Simulation. *Probabilistic Engineering Mechanics*, **41**:89–103, 2015.
- [8] Konstantin M Zuev. Subset Simulation method for rare event estimation: An introduction. *Encyclopedia of Earthquake Engineering*, pages 1–25, 2015.
- [9] Ziqi Wang, Marco Broccardo, and Junho Song. Hamiltonian Monte Carlo methods for Subset Simulation in reliability analysis. *Structural Safety*, **76**:51–67, 2019.
- [10] Hamed Nikbakht and Konstantinos G. Papakonstantinou. Hamiltonian MCMC methods for estimating rare events probabilities in high-dimensional problems. *Journal of Computational Physics*, under review.
- [11] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, **195**(2):216–222, 1987.

- [12] Radford M Neal. Bayesian learning for neural networks. *PhD Thesis, University of Toronto*, 1995.
- [13] Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, **2**(11):2, 2011.
- [14] Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, **15**(1):1593–1623, 2014.
- [15] Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart. Optimal tuning of the Hybrid Monte Carlo algorithm. *Bernoulli*, **19**(5A):1501–1534, 2013.
- [16] Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.
- [17] Andrew Gelman, Hal S Stern, John B Carlin, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2013.
- [18] John Kruschke. *Doing Bayesian Data Analysis: A tutorial with R, JAGS, and Stan*. Academic Press, 2014.
- [19] Cole C Monnahan, James T Thorson, and Trevor A Branch. Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo. *Methods in Ecology and Evolution*, **8**(3):339–348, 2017.
- [20] Elena Akhmatskaya and Sebastian Reich. GSHMC: An efficient method for molecular simulation. *Journal of Computational Physics*, **227**(10):4934–4954, 2008.
- [21] Nilesh Tripurani, Mark Rowland, Zoubin Ghahramani, and Richard Turner. Magnetic Hamiltonian Monte Carlo. *arXiv preprint arXiv:1607.02738*, 2016.
- [22] Ziyu Wang, Shakir Mohamed, and Nando Freitas. Adaptive Hamiltonian and Riemann manifold Monte Carlo. In *International Conference on Machine Learning*, pages 1462–1470, 2013.
- [23] Michael Hohenbichler and Rudiger Rackwitz. Non-normal dependent vectors in structural safety. *Journal of the Engineering Mechanics Division*, **107**(6):1227–1238, 1981.
- [24] Armen Der Kiureghian and Pei-Ling Liu. Structural reliability under incomplete probability information. *Journal of Engineering Mechanics*, **112**(1):85–104, 1986.
- [25] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2006.
- [26] Armen Der Kiureghian and Taleen Dakessian. Multiple design points in first and second-order reliability. *Structural Safety*, **20**(1):37–49, 1998.
- [27] Christian G Bucher and Ulrich Bourgund. A fast and efficient response surface approach for structural reliability problems. *Structural Safety*, **7**(1):57–66, 1990.
- [28] Roland Schöbi and Bruno Sudret. Structural reliability analysis for p-boxes using multi-level meta-models. *Probabilistic Engineering Mechanics*, **48**:27–38, 2017.