INL/CON-23-70933-Revision-0

Efficient Subset Simulation using Hamiltonian Neural Network enhanced Markov Chain Monte Carlo Methods

July 2023

Denny Thaler, Franz Bamer, Bernd Markert, Michael Shields, Som LakshmiNarasimha Dhulipala

Changing the World's Energy Future

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/CON-23-70933-Revision-0

Efficient Subset Simulation using Hamiltonian Neural Network enhanced Markov Chain Monte Carlo Methods

Denny Thaler, Franz Bamer, Bernd Markert, Michael Shields, Som LakshmiNarasimha Dhulipala

July 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

14th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP14
Dublin, Ireland, July 9-13, 2023

Efficient Subset Simulation using Hamiltonian Neural Network enhanced Markov Chain Monte Carlo Methods

Denny Thaler
Ph.D. Candidate, Institute of General Mechanics, RWTH Aachen University, Aachen, Germany

Somayajulu L.N. Dhulipala
Computational Scientist, Computational Mechanics and Materials, Idaho National Laboratory, Idaho Falls, USA

Franz Bamer
Postdoctoral Researcher, Institute of General Mechanics, RWTH Aachen University, Aachen, Germany

Bernd Markert
Professor, Institute of General Mechanics, RWTH Aachen University, Aachen, Germany

Michael D. Shields
Associate Professor, Department of Civil and Systems Engineering, Johns Hopkins University, Baltimore, USA

ABSTRACT: The Monte Carlo method delivers an unbiased estimate of the probability of failure. However, the variance of the estimate depends on the number of evaluated samples. This number must be very large for estimations of a low probability of failure. If the evaluation of each sample is computationally expensive, the crude Monte Carlo simulation strategy is impracticable. Therefore, subset simulations are used to reduce the required number of evaluations. Subset simulations require a Markov Chain Monte Carlo sampler, such as the random walk Metropolis-Hastings algorithm. The algorithm, however, struggles with sampling in low-probability regions, especially if they are narrow. As a consequence, advanced Markov Chain Monte Carlo simulations have been developed. In particular, the Hamiltonian Monte Carlo method explores the target distribution rapidly. Driven by the idea of Hamiltonian dynamics, this sampler provides a non-random walk through the target distribution. The incorporation of subset simulation and Hamiltonian Monte Carlo methods has shown promising results for reliability analysis. One downside of the Hamiltonian Monte Carlo method is that gradient evaluations are computationally expensive, especially when dealing with high-dimensional problems and evaluating long trajectories. We show that integrating Hamiltonian neural networks in Hamiltonian Monte Carlo simulations significantly speeds up the sampling task. Furthermore, the enhancement of adaptive trajectory length within the Hamiltonian Monte Carlo results in the efficient proposal of the following states. Based on this recent enhancement, we provide a fast sampling strategy for subset simulations using Hamiltonian neural networks to replace the evaluation of the gradient and significantly speed up the Hamiltonian Monte Carlo simulation.

1. INTRODUCTION

Rigorous analysis to estimate the response statistics of structures and relevant structural parts is crucial in probabilistic engineering. To this end, engineers have to investigate the uncertainties in material properties, loads, and design. Reliability analysis aims to estimate the probability of failure considering these uncertainties, which is written as:

$$P(F) = \int_{g(\mathbf{x})\leq 0} \cdots \int f_X(\mathbf{x})\, \mathrm{d}x_1 \ldots \mathrm{d}x_n \,, \tag{1}$$

where $f_X(\mathbf{x})$ is the probability density function of the random vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$. The failure region is defined using the limit state function $g(\mathbf{x})$. Failure occurs if this function is negative, i.e., $g(\mathbf{x}) \leq 0$. However, due to complex response functions and probability density functions, this integration is sometimes impossible. Therefore, it requires simulation methods. The Monte Carlo method evaluates independent samples from the distribution to provide an unbiased estimate of the probability of failure.

In general, most structural entities in our daily life are designed so that they fail rarely. At the same time, the construction cost must remain reasonable. However, in the case of high-damage events, such as the collapse of a building or the failure of key functional parts in aerospace engineering, the failure probability should be extremely low. For a reliable estimate of low probabilities, the number of required samples for the crude Monte Carlo simulations becomes huge. In the case of complex systems, the computational cost of these simulations becomes infeasible. To decrease the computational burden, there are two approaches:

- Methods to reduce the computational burden of each sample evaluation: The complexity of structures can be simplified, and model order reduction methods applied (e.g. Bamer and Bucher (2012); Bamer et al. (2017)). Furthermore, machine learning models can learn the structural behavior and enable many evaluations (Thaler et al. (2021); Bamer et al. (2021)).
- Methods to reduce the number of samples: Established methods are, for example, importance sampling and subset simulations (Au and Beck (2001)).

In this contribution, we focus on subset simulations. This algorithm requires Markov Chain Monte Carlo sampling to ensure the correct mapping of the probability density in the subset. The basic, traditional approach is the Metropolis-Hastings sampling algorithm (Hastings (1970)). However, for complex and high-dimensional distributions, as well as for very thin distributions, this random-walk method results in a low acceptance rate, which leads often to an underestimation of the failure probability. To this end, more elaborate sampling strategies have been developed, e.g., Stretch-sampling and Hamiltonian Monte Carlo. Both approaches have shown promising results in subset simulations (Shields et al. (2021); Wang et al. (2019)). However, the Hamiltonian Monte Carlo simulation requires expensive gradient calculation for each step during the integration procedure for the proposal of a new state.

One possibility to decrease the computational burden is to use neural networks for the gradient calculation, cf. Li et al. (2019). Recently, researchers focused on more elaborate machine learning architectures, e.g., physics-informed neural networks (Raissi et al. (2017)), and gradient-based neural networks have gained popularity. The Hamiltonian neural network is a gradient-based architecture, which has been proposed by Greydanus et al. (2019) to solve the dynamics of mechanical systems.

Dhulipala et al. (2022) proposed to use latent Hamiltonian neural networks for Bayesian inference problems using both the Hamiltonian Monte Carlo and the No U-Turn sampler since the architecture shows good estimations over long trajectories in classic mechanics problems. The applicability of Hamiltonian neural networks was further demonstrated by (Thaler et al.) for both the traditional and the dynamic Hamiltonian Monte Carlo simulation.

Considering that the Hamiltonian neural network is suitable to use in Hamiltonian Monte Carlo, we

propose to use the enhanced method in subset simulations. In the following, we show the basic theory and demonstrate the strategy on numerical examples.

2. HAMILTONIAN NEURAL NETWORK SUBSET SIMULATIONS

2.1. Methodology

2.1.1. Probability of Failure

As stated in Equation 1, the failure probability is generally written as the integral over the failure domain. However, one can also use an indicator function $I_g(\mathbf{x})$, which is 1 if $g(\mathbf{x}) \leq 0$, and else 0. Thus, one integrates over the entire domain (cf. Bucher (2009)) using the indicator function:

$$P(F) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} I_g(x_1,\ldots,x_n) \, f_{X_1,\ldots,X_n}(x_1,\ldots,x_n)\,\mathrm{d}x_1\ldots\mathrm{d}x_n\,. \quad (2)$$

Unfortunately, the solution to this integral is, in many cases, not straightforward, and often there is no analytical solution. Therefore, one needs to draw samples from the distribution and evaluate them. This procedure is the crude Monte Carlo simulation. The estimation of the failure probability is then the number of failures divided by the total number of experiments:

$$P(F) = \frac{1}{m}\sum_{k=1}^{m} I_g(\mathbf{x}^{(k)})\,. \quad (3)$$

The Monte Carlo method delivers an unbiased estimate of the failure probability. However, for low probabilities and complex distributions, the method has a high variance, except one evaluates a vast number of samples. To reduce the computational burden, subset simulations were proposed (Au and Beck (2001)).

2.1.2. Subset Simulations

In subset simulations, intermediate "failure" regions are used to sample gradually towards the region of failure (Au and Wang (2014)). After each subset evaluation, only the samples which are closest to the failure region, in other words, which reached the intermediate region, are considered. Thus, the probability of failure is composed of the probabilities of reaching the next intermediate state, given that the sample is in the current region:

$$\begin{aligned} P(F) =& P(F_n) = P(\cap_{i=1}^{n} F_i) \\ =& P(F_n \mid \cap_{i=1}^{n-1} F_i) P(\cap_{i=1}^{n-1} F_i) \\ =& P(F_n \mid \cap F_{n-1}) P(\cap_{i=1}^{n-1} F_i) \\ =& \cdots = P(F_1)\prod_{i=1}^{n-1} P(F_{i+1} \mid F_i) \end{aligned} \quad (4)$$

The intermediate failure regions can be chosen adaptively, e.g., by choosing the probability of reaching the next region $P(F_{i+1} \mid F_i) = 0.1$. The threshold for the region is then based on the highest value of the limit state function of the samples. The samples for each set are simulated using the state from the previous set by Monte Carlo Markov Chain samplers, i.e., the Hamiltonian Monte Carlo.

2.1.3. Hamiltonian Monte Carlo

In general, any Monte Carlo Markov Chain sampler can be used to propagate samples through the distribution. However, classic approaches, such as the Metropolis-Hastings sampler, suffer from poor acceptance rates and high correlations between samples for complex distributions. To this end, the Hamiltonian Monte Carlo sampler provides a non-random walk sampling that efficiently explores the target distribution. We briefly introduce the method in this section. More details are provided by Brooks et al. (2011) and Betancourt (2017).

Instead of a random sample proposal, the sampler conserves energy-like quantities of the current state to provide a new state. The idea of the sampler is inspired by Hamiltonian mechanics. The Hamiltonian H is the sum of the potential U and kinetic energy K of a system:

$$H(\mathbf{q},\mathbf{p}) = U(\mathbf{q}) + K(\mathbf{p})\,. \quad (5)$$

Hereby, in classical mechanics, $\mathbf{q}$ is the position, and $\mathbf{p}$ is the momentum of the system. Using the derivatives of the Hamiltonian with respect to the phase space variables $\mathbf{q}$ and $\mathbf{p}$ enables to evaluate the canonical equations of the system:

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}} ,$$
$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}} . \quad (6)$$

Incorporating the idea of Hamiltonian dynamics in the Markov Chain Monte Carlo concept generates a Hamiltonian Monte Carlo Markov sampler. Thus, for the Hamiltonian Monte Carlo algorithm, a mathematical equivalent model is built to evaluate the samples in the target probability space. The vector $\mathbf{q}$ is then defined as the variable of interest having a target distribution $\pi(\mathbf{q})$. The vector $\mathbf{p}$ is used as an auxiliary variable to introduce randomness to the sampler. The Hamiltonian is reformulated based on the probability distribution as:

$$H(q,p) = -\log \pi(\mathbf{q},\mathbf{p}) . \quad (7)$$

Analogous to the classical (mechanical) system, the "potential energy" is defined as the negative of the logarithmic probability density of the target distribution:

$$U(\mathbf{q}) = -\log \pi(\mathbf{q}) , \quad (8)$$

and the "kinetic energy" is formulated as a function of the auxiliary variable $\mathbf{p}$, usually chosen as:

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}\mathbf{M}^{-1}\mathbf{p} . \quad (9)$$

The initial state of the variable of interest $\mathbf{q}$ is chosen randomly, e.g., from multivariate normal Gaussian distribution. Starting from that state, the algorithm requires a random Gaussian vector $\mathbf{p}$ to "walk" through the distribution. Using $\mathbf{q}$ and $\mathbf{p}$, the derivatives of the Hamiltonian can be determined. In general, the leapfrog integrator is used to evaluate the trajectory. After finishing the chosen step number L, the updated auxiliary variable $\mathbf{p^L}$ is negated $\mathbf{p}^* = -\mathbf{p^L}$ to make the proposal state $(\mathbf{q}^*, \mathbf{p}^*)$ reversible.

Finally, the evaluated next state is accepted or rejected based on a reformulation of the Metropolis-Hastings criteria for Hamiltonian Monte Carlo. The new state $(\mathbf{q}^*, \mathbf{p}^*)$ is accepted with a probability of

$$\alpha = \min\left[1, \exp(H((\mathbf{q},\mathbf{p}) - H(\mathbf{q}^*,\mathbf{p}^*)\right] . \quad (10)$$

The algorithm for the next state proposal in Hamiltonian Monte Carlo simulation is provided in Algorithm 1.

Algorithm 1 Hamiltonian Monte Carlo

Initial state: q, Hamiltonian: H, Steps: L, Step size: ε
$p_0 \leftarrow \mathcal{N}(0,1)$
$q_0 \leftarrow q$
$p_{\frac{1}{2}} \leftarrow p_0 - \frac{\varepsilon}{2}\frac{\partial H}{\partial q}$
for $1 \leq N < L$ **do** ▷ Leapfrog integrator
  $q_n \leftarrow q_{n-1} + \varepsilon p_n$
  $p_{n+\frac{1}{2}} \leftarrow p_{n-\frac{1}{2}} - \varepsilon\frac{\partial H}{\partial q}$
end for
$p_L \leftarrow p_{L-1} - \frac{\varepsilon}{2}\frac{\partial H}{\partial q}$
$q* \leftarrow q_L$
$p* \leftarrow -p_L$
$\alpha = \min\left[1, \exp\left(-H(q*,p*) + H(q,p)\right)\right]$
if $\alpha \geq \mathcal{U}(0,1)$ **then**
  return $(q*, p*)$
else
  return (q_0, p_0)
end if

Performing the Hamiltonian Monte Carlo simulation, one has to provide the step size ε and the number of leapfrog steps L. This choice influences the efficiency and reliability of the Monte Carlo simulation. As a consequence, for a complex distribution, the dynamic Hamiltonian Monte Carlo sampler or the No U-Turn sampler (Homan and Gelman (2014)) is preferable.

Assuming well-chosen ε and L, the sampler provides a more effective exploration of the target distribution compared to the Metropolis-Hastings algorithm. However, one drawback of the Hamiltonian Monte Carlo method is the high computational cost. For each new state, the leapfrog integrator takes L steps, and in each of these steps, the numerical gradient of the Hamiltonian must be calculated to update the state. For the sake of computational efficiency, surrogate models can be used to predict the next state.

2.1.4. Hamiltonian Neural Networks

For the speed-up of the Hamiltonian Monte Carlo algorithm, any surrogate model which is able to predict the gradient of the Hamiltonian accurately can replace the numerical gradient evaluation. To this end, traditional neural networks have been used (Li et al. (2019)). However, more elaborate architectures have been created in recent research. Hamiltonian neural networks have shown to conserve the Hamiltonian well (Greydanus et al. (2019)). Indeed, conservation of the Hamiltonian is needed to provide good proposals for the next state within the Hamiltonian Monte Carlo simulation. Therefore, using Hamiltonian neural networks in Hamiltonian Monte Carlo simulation is particularly promising.

In Figure 1, the traditional feedforward neural network architectures and the Hamiltonian neural network architecture are shown.

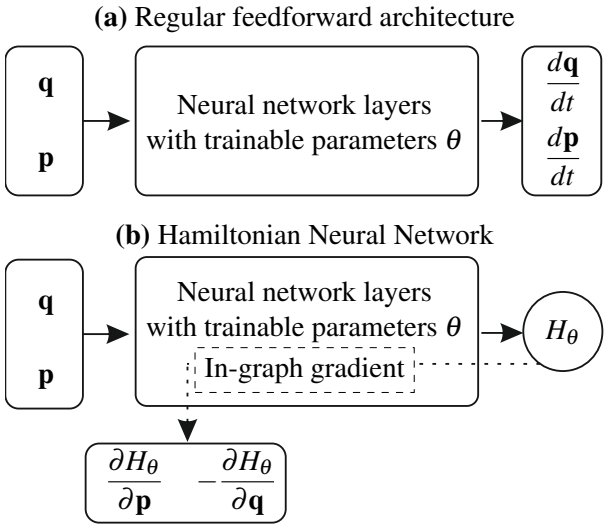

Figure 1: Architecture of artificial neural networks: (a) The "traditional" feedforward architecture predicts the targets in the output layer; (b) The Hamiltonian neural network architecture uses the in-graph gradient of the output node with respect to the input nodes to predict the targets (cf. Greydanus et al. (2019)).

The main difference between the two architectures is the way the target is estimated. The regular architecture predicts the targets in the output layer, i.e. the derivative of the input with respect to time. On the other hand, the Hamiltonian neural network uses an in-graph gradient of the output with respect to the input. Thus, it predicts the gradient of the Hamiltonian with respect to the input quantities, which is equivalent to the input with respect to time, cf. Equation 6.

The gradient prediction requires a small fraction of the time necessary to evaluate the numerical gradient traditionally. This enables fast evaluation of the trajectory. However, traditional numerical gradient evaluation is necessary to provide training data for supervised learning of the Hamiltonian neural network.

2.2. Hamiltonian Neural Network sampler

The speed of the Hamiltonian Monte Carlo simulation can be increased by using Hamiltonian neural networks to predict the gradient instead of the traditional numerical gradient calculation (Dhulipala et al. (2022); (Thaler et al.)). Algorithm 1 is also used for the proposal of the next state with the faster gradient evaluation using the Hamiltonian neural network. The sampler is used to create an initial sample set and thereafter to simulate the next states within the subset simulation. In the following chapter, we show the application of the proposed strategy.

2.3. Application

The numerical examples are implemented in Python using the environment for subset simulations within the package UQPy for uncertainty quantification (Olivier et al. (2020)). The code for Hamiltonian neural networks for accelerating Hamiltonian Monte Carlo (HMC) are available through the Bayesian inference with Hamiltonian Neural Networks (BIhNNs) Github repository (https://github.com/IdahoLabResearch/BIhNNs).

2.3.1. Gaussian Distribution

For the numerical example for subset simulations, a multivariate normal distribution with zero mean $\boldsymbol{\mu}$ and identical correlation ρ between all of its components is chosen:

$$\boldsymbol{\mu} = [0, 0, \cdots, 0, 0]; \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \rho \\ \rho & \dots & \rho & 1 \end{bmatrix}. \tag{11}$$

The probability density function of this distribution is written as:

$$p(\mathbf{q}) = \frac{exp\left(-0.5\,(\mathbf{q}-\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}\,(\mathbf{q}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^2\,\|\boldsymbol{\Sigma}\|}}\,. \quad (12)$$

We choose a linear limit state function with a specified reliability index β:

$$g(\mathbf{x}) = \beta\sqrt{\sigma_{max}n} - \sum_{i=1}^{n} x_i\,. \quad (13)$$

Hereby, σ_{max} denotes the maximum eigenvalue of the covariance matrix $\boldsymbol{\Sigma}$ of the distribution. Figure 2 shows the described problem for uncorrelated $\rho = 0$ and highly correlated $\rho = 0.95$ Gaussian variables in two-dimensional space. The value of β is chosen to be 4 in the example.

Using the above-mentioned settings, we perform the subset simulation for two and ten variables. The Hamiltonian neural network requires supervised training before it can be used within the sampler. The training data is collected using 40 samples and evaluating them using the traditional gradient calculation. The Hamiltonian neural network architecture has three hidden layers with ten neurons in each layer. This setting is chosen for all examples. Furthermore, the settings for the Hamiltonian neural network Monte Carlo sampling are also kept constant for all examples. We chose a step size of $\varepsilon = 0.025$ and calculated the trajectory length and adaptively changed it within the subsets. For this purpose, we used the algorithms proposed by Wang et al. (2019) for Hamiltonian Monte Carlo subset simulations. The result of one subset simulation for two dimensions using Hamiltonian neural networks in the Hamiltonian Monte Carlo sampler is shown in Figure 3.

The probability of failure of this simulation is $2.64 * 10^{-5}$ which corresponds to $\beta = 4.04$. This subset simulation was highly accurate since there is only an error of 0.04 in beta.

Figure 4 shows the result of the Hamiltonian neural network subset simulation with high correlation $\rho = 0.95$. The result for the failure probability is $1.48 * 10^{-6}$ which is equivalent to a value of $\beta = 4.67$. Even though there is a higher error compared to the uncorrelated case, this result is in a reasonable range. In both simulations, the subsets evaluate over the intermediate failure domains and show fair acceptance rates. The fundamental applicability of Hamiltonian neural networks for subset simulations works out. To verify these good findings, we executed various simulations for two- and ten-dimensional problems.

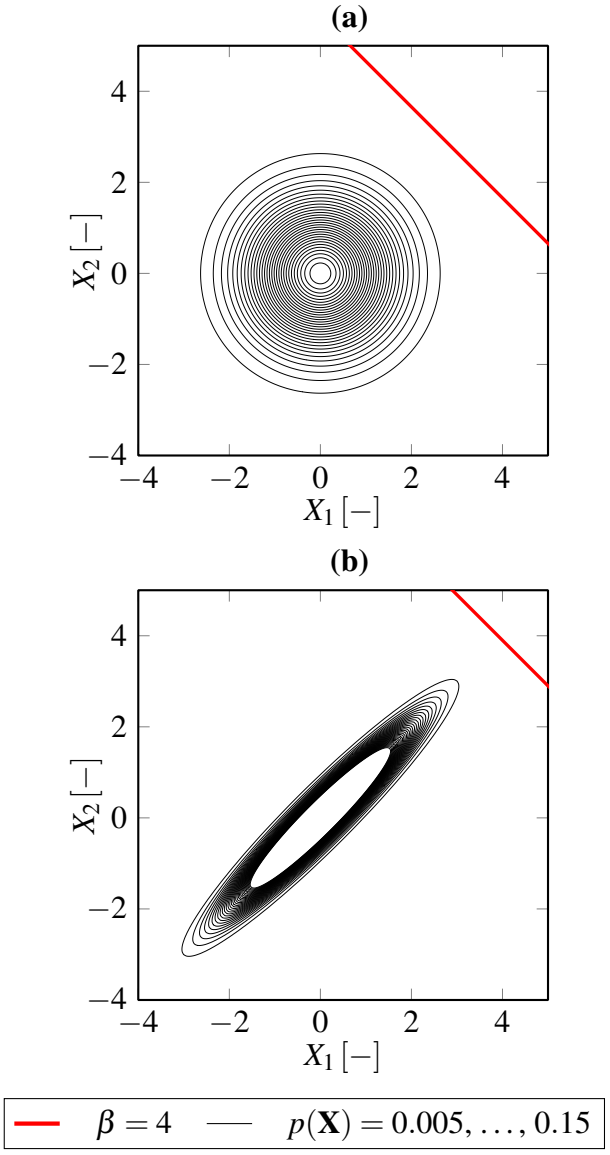

Figure 2: Probability density levels for (a) uncorrelated $\rho = 0$ and (b) highly correlated $\rho = 0.95$ Gaussian variables are shown in black down to a probability of 0.005. The limit state function g is highlighted in red for the reliability index $\beta = 4$.

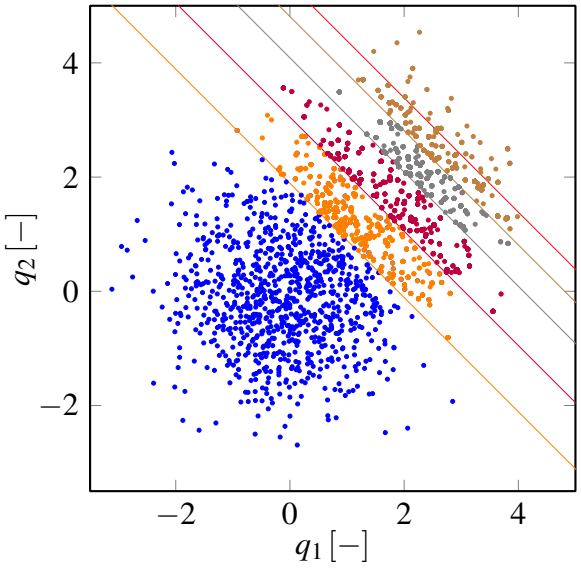

Figure 3: Subset results for one subset simulation using two-dimensional Gaussian distribution with zero correlation $\rho = 0$.

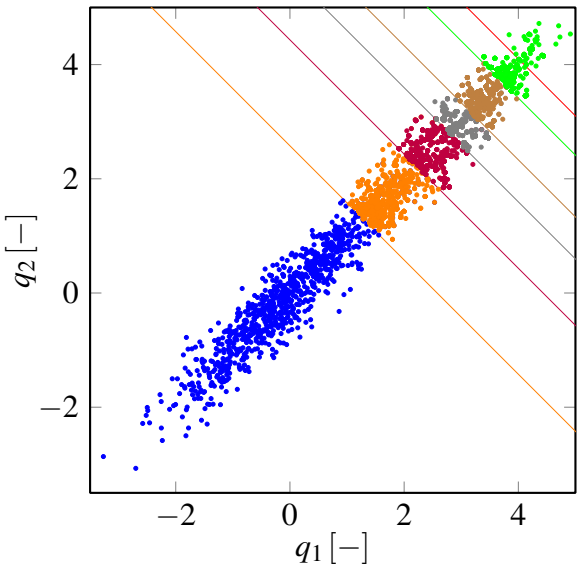

Figure 4: Subset results for one subset simulation using two-dimensional Gaussian distribution with high correlation $\rho = 0.95$.

The simulations are performed 100 times for the two-dimensional and the ten-dimensional case. Here, we used uncorrelated $\rho = 0$ and highly correlated $\rho = 0.95$ distributions again. For comparison, we ran the same simulations using the modified Metropolis-Hastings algorithm. The results in terms of the mean failure index β are shown in Table 1. The correct solution for all four simulations is $\beta = 4$. Furthermore, the coefficient of variation (CV) is of interest, as it shows how the results vary in relation to the calculated mean value.

beta	$\rho = 0$		$\rho = 0.95$	
4	mean	CV	mean	CV
HNN 2-D	4.43	0.039	4.19	0.050
MMH 2-D	4.33	0.052	4.23	0.075
HNN 10-D	4.42	0.039	4.28	0.066
MMH 10-D	4.63	0.061	6.25	0.074

Table 1: Mean values and coefficient of variation (CV) of 100 subset simulations using the Hamiltonian Neural Network sampler (HNN) and the modified Metropolis-Hastings algorithm (MMH) for two-dimensional (2-D) and ten-dimensional (10-D) correlated ρ distribution.

We observe that both samplers provide reliable results in the case of zero correlation $\rho = 0$ for two and ten dimensions. Also, the variance is relatively low in these cases. However, if we choose highly correlated distribution $\rho = 0.95$ the results differ a lot. In the case of two dimensions, the Modified Metropolis-Hastings algorithm gives an appropriate estimate of the failure. However, the algorithm struggles with the high correlation distribution if ten variables are included. In contrast, the Hamiltonian Neural Network Monte Carlo simulation is able to estimate the failure probability, or in our case the reliability index with a mean error of 0.63. Furthermore, the coefficient of variation of this approach is more stable compared to the modified Metropololis-Hastings algorithm, see MMH 10-D and HNN 10-D in Table 1.

3. CONCLUSIONS

This contribution corroborates the potential of gradient-based artificial neural networks for Hamiltonian Monte Carlo simulations. The results show the applicability of Hamiltonian neural networks for subset simulations, which enables fast evaluations. However, the examples are shown on two- and ten-dimensional examples with different correlations. Future work should focus on higher dimensions and incorporate different distributions. Furthermore, the regions close to the failure region might have low acceptance rates if the error of the

gradient calculation decreases typically in supervised learning models (cf. Thaler et al. (2022)).

4. ACKNOWLEDGEMENT

Denny Thaler was supported by a fellowship of the German Academic Exchange Service (DAAD) during his research stay at Johns Hopkins University. Som Dhulipala and Michael Shields were supported through INL's Laboratory Directed Research and Development (LDRD) Program under U.S. Department of Energy (DOE) Idaho Operations Office contract no. DE-AC07-05ID14517.

5. REFERENCES

Au, S.-K. and Beck, J. L. (2001). "Estimation of small failure probabilities in high dimensions by subset simulation." *Probabilistic Engineering Mechanics*, 16(4), 263–277.

Au, S.-K. and Wang, Y. (2014). *Engineering Risk Assessment with Subset Simulation*. John Wiley & Sons Singapore Pte. Ltd., <https://doi.org/10.1002/9781118398050> (April).

Bamer, F., Amiri, A. K., and Bucher, C. (2017). "A new model order reduction strategy adapted to nonlinear problems in earthquake engineering." *Earthquake Engineering & Structural Dynamics*, 46(4), 537–559.

Bamer, F. and Bucher, C. (2012). "Application of the proper orthogonal decomposition for linear and nonlinear structures under transient excitations." *Acta Mechanica*, 223(12), 2549–2563.

Bamer, F., Thaler, D., Stoffel, M., and Markert, B. (2021). "A monte carlo simulation approach in nonlinear structural dynamics using convolutional neural networks." *Frontiers in Built Environment*, 7.

Betancourt, M. (2017). "A conceptual introduction to hamiltonian monte carlo, <https://arxiv.org/abs/1701.02434>.

S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, eds. (2011). *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, <https://doi.org/10.1201/b10905>.

Bucher, C. (2009). *Computational Analysis of Randomness in Structural Mechanics: Structures and Infrastructures Book Series, Vol. 3 (1st ed.)*. CRC Press, <https://doi.org/10.1201/9780203876534>.

Dhulipala, S. L. N., Che, Y., and Shields, M. D. (2022). "Bayesian inference with latent hamiltonian neural networks, <https://arxiv.org/abs/2208.06120>.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). "Hamiltonian neural networks, <https://arxiv.org/abs/1906.01563>.

Hastings, W. K. (1970). "Monte Carlo sampling methods using Markov chains and their applications." *Biometrika*, 57(1), 97–109.

Homan, M. D. and Gelman, A. (2014). "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo." *J. Mach. Learn. Res.*, 15(1), 1593–1623.

Li, L., Holbrook, A., Shahbaba, B., and Baldi, P. (2019). "Neural network gradient hamiltonian monte carlo." *Computational Statistics*, 34(1), 281–299.

Olivier, A., Giovanis, D. G., Aakash, B., Chauhan, M., Vandanapu, L., and Shields, M. D. (2020). "Uqpy: A general purpose python package and development environment for uncertainty quantification." *Journal of Computational Science*, 47, 101204.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, <https://arxiv.org/abs/1711.10561>.

Shields, M. D., Giovanis, D. G., and Sundar, V. (2021). "Subset simulation for problems with strongly non-gaussian, highly anisotropic, and degenerate distributions." *Computers & Structures*, 245, 106431.

Thaler, D., Dhulipala, S. L. N., Bamer, F., Markert, B., and Shields, M. D. "Enhanced hamiltonian monte carlo simulations using hamiltonian neural networks. PAMM, 202200188, (Accepted).

Thaler, D., Elezaj, L., Bamer, F., and Markert, B. (2022). "Training data selection for machine learning-enhanced monte carlo simulations in structural dynamics." *Applied Sciences*, 12(2).

Thaler, D., Stoffel, M., Markert, B., and Bamer, F. (2021). "Machine-learning-enhanced tail end prediction of structural response statistics in earthquake engineering." *Earthquake Engineering & Structural Dynamics*, 50(8), 2098–2114.

Wang, Z., Broccardo, M., and Song, J. (2019). "Hamiltonian monte carlo methods for subset simulation in reliability analysis." *Structural Safety*, 76, 51–67.