Multifidelity Cross-Entropy Estimation of Rare Events in Steady Heat Conduction

Terrence Alsup and Frederick Law

May 10, 2019

Abstract

For rare event estimation in models of physical systems simple Monte Carlo estimates require a huge number of samples. However, if the model is very complex, such as requiring the solution to a PDE, then each sample will be expensive to compute. The cross-entropy (CE) method estimates the rare event probability with importance sampling by approximating the optimal biasing density. This significantly reduces the variance of our estimator allowing us to draw fewer samples. Multifideltity cross-entropy (MFCE) is a multilevel extension of CE, which further reduces computational cost by using coarser approximations of our model as pre-conditioners for the CE method. In this report we show that both CE and MFCE vastly outperform Monte Carlo and that MFCE is a cheaper alternative to CE for the model problem of steady-state heat flow.

1 Introduction

1.1 Model Problem

The main model problem we are interested in investigating is steady-state heat conduction on the unit square $\Omega = [0,1]^2$, with zero Dirichlet data on the top, zero Neumann data on the sides, and unit Neumann data on the bottom.

$$\begin{cases} \nabla \cdot (k(x,\omega)\nabla u(x)) = -1, \quad x \in \Omega \\ u = 0, \quad x \in \Gamma_1 \\ u \cdot \hat{n} = 0, \quad x \in \Gamma_2 \\ u \cdot \hat{n} = 1, \quad x \in \Gamma_3 \end{cases}$$

Here $k(x,\omega)$ is our random field heat conduction parameter. The PDE is discretized using linear finite elements, with $2^l + 1$ elements in each dimension, so $\sim 4^\ell$ points total. We consider models $\ell = 3,4,5$ with the maximal level $L = 5$. Moreover, we treat the random field $k(x,\omega)$ to be piecewise constant where we cut the domain Ω in d equally sized squares, with $d = 1,4,16$. As a result we can represent this piecewise random field k by a d-dimensional random vector ξ.

Our quantity of interest (QoI) in this model problem is the total heat at the base of the domain: $f^{(\ell)} = \int_{\Gamma_1} u^{(\ell)}\, dx$ where the superscript ℓ denotes the solution and QoI for level ℓ. We simulate the QoI by realizing $\xi \sim p$, solving the PDE with this realization of ξ as our parameter and repeating this many times. Here we choose the distribution on ξ as d-dimensional log-normal $p \sim \text{LogNormal}_d(\mu, S)$ where μ is a vector of all 1.6 and S is diagonal matrix with each entry $(0.25)^2$. We choose log-normal ξ instead of the somewhat more intuitive option of ξ being uniform. This choice of distribution is elaborated on in Section 2.2.

The event of interest is when the QoI is above a certain threshold t, which represents when the total heat at the base is above some given tolerance. This could represent material failure (i.e. the domain has some material properties which should not allow such high heat) or could represent some safety tolerance (i.e. experimental equipment is not suited to handle such large heat). So the quantity

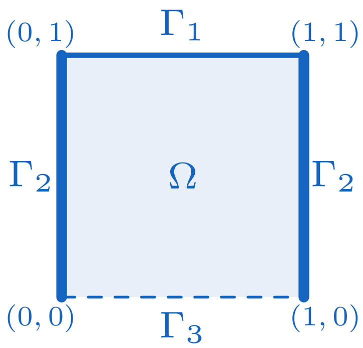


Figure 1: Domain of steady state heat conduction. We have a forcing -1 term in $\Omega = (0,1)^2$ with 0 Dirichlet data on Γ_1, 0 Neumann data on Γ_2, and Neumann data 1 on Γ_3.

we are interested in computing is $P_t^{(L)} = \mathbb{P}\left(f^{(L)} \geq t\right)$. Ideally we would estimate this by simulating at the maximal level L, but since the number of points in the model increases exponentially in ℓ, model evaluations become increasingly expensive.

1.2 Rare Event Estimation with Monte Carlo

A simple Monte Carlo estimator of a rare event probability is given by

$$\hat{P}_t^{\mathrm{MC}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{f^{(L)}(\xi^{(i)}) \geq t\}, \quad \text{where} \quad \xi^{(i)} \overset{\text{i.i.d.}}{\sim} p. \tag{1}$$

The variance of this estimator is given by $\mathrm{Var}_p[\hat{P}_t^{\mathrm{MC}}] = \frac{P_t^{(L)}(1-P_t^{(L)})}{N}$, and so the squared coefficient of variation (SQCoV) is given by

$$\mathrm{SQCoV}(\hat{P}_t^{\mathrm{MC}}) = \frac{1 - P_t^{(L)}}{N P_t^{(L)}}. \tag{2}$$

For this quantity to be small we require that $N \sim O(1/P_t^{(L)})$, but for rare events $P_t^{(L)}$ may be extremely small. Since the estimator (1) requires an evaluation of the high-fidelity model for each sample, this method quickly becomes intractable for very small probabilities. To alleviate this problem we switch to using importance sampling to reduce the variance of our estimators. In particular, we look at the CE method and its multilevel extension MFCE.

1.3 Cross-Entropy and Multifidelity Cross-Entropy

For a general biasing density q we can estimate the rare event probability with importance sampling:

$$\hat{P}_t^{\mathrm{IS}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{f^{(L)}(\xi^{(i)}) \geq t\}\frac{p(\xi^{(i)})}{q(\xi^{(i)})}, \quad \text{where} \quad \xi^{(i)} \overset{\text{i.i.d.}}{\sim} q. \tag{3}$$

However, the performance of this estimator depends strongly on which biasing density we choose. In particular, if we use

$$q^*(\xi) = \frac{\mathbf{1}\{f^{(L)}(\xi) \geq t\}\ p(\xi)}{P_t^{(L)}}, \tag{4}$$

then the estimator (3) has zero variance and we would need only 1 high-fidelity model evaluation. Of course, we do not know the optimal biasing density because it requires the quantity we are trying to estimate. The cross-entropy (CE) method simultaneously approximates the optimal biasing density and estimates the rare event probability. The CE method is described in detail in both Algorithm 8.2.1. of [2] and in section 2.3.2 of [1]. For our problem we optimize over a family of Log-normal distributions as our biasing densities since the target p is Log-normal. This is a convenient family to use because at each step in the CE method we have a closed-form solution for the optimal mean μ^* and covariance Σ^*.

Although CE performs much better than simple Monte Carlo, it may require many iterations, each of which requires drawing N high-fidelity model evaluations. If we have access to lower-fidelity models $f^{(\ell)}$ that are good approximations to $f^{(L)}$, then we should be able to offload most of the work to these cheaper models and then use them as pre-conditioners to speed-up CE at the highest level. This is the main idea behind MFCE and many other multilevel methods in general. Our Matlab implementation follows Algorithm 1 in [1]. The main difference is that we are estimating upper tail probabilities whereas Algorithm 1 estimates lower tail probabilities. The final estimator produced is

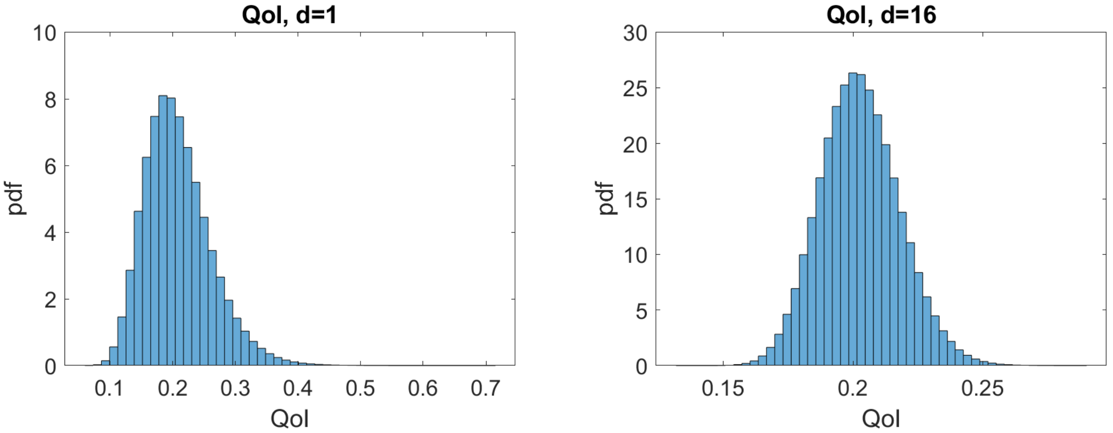
$$\hat{P}_t^{\text{MFCE}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{f^{(L)}(\xi^{(i)}) \geq t\} \frac{p(\xi^{(i)})}{q^{(L)}_{\mu^*,\Sigma^*}(\xi^{(i)})}, \qquad \text{where} \qquad \xi^{(i)} \overset{\text{i.i.d.}}{\sim} q^{(L)}_{\mu^*,\Sigma^*}, \tag{5}$$

where $q^{(L)}_{\mu^*,\Sigma^*}$ is the optimal Log-normal biasing density found at the highest level. In the following sections we will show that we can obtain much more accurate estimates using CE and that we can further reduce the cost by using MFCE.

2 Numerical Preliminaries

2.1 Quantity of Interest and Rare Event Thresholds

To determine the thresholds corresponding to low probabilities, we sampled our high-fidelity model $N = 10^6$ times for $d = 1, 4, 16$ and looked at empirical upper quantiles which corresponded to somewhat large rare events. Histograms showing the distribution of the QoI induced by ξ, as well as some of the upper quantiles are in Figure 2.



Upper tail with probability:	1e-02	5e-03	1e-03	5e-04	1e-04	5e-05	1e-05
$d = 1, L = 5, N = 10^6$	0.3609	0.3842	0.4374	0.4607	0.5139	0.5320	0.5895
$d = 4, L = 5, N = 10^6$	0.2830	0.2937	0.3173	0.3267	0.3485	0.3561	0.3780
$d = 16, L = 5, N = 10^6$	0.2408	0.2454	0.2553	0.2594	0.2678	0.2719	0.2786

Figure 2: Histograms and approximate upper quantiles for the distribution of $f^{(L)}$. Estimates of 1e-05 are likely not very strong for $N = 10^6$, as there are only 10 values above that threshold

With the approximate upper quantiles found in Figure 2, we use these to build numerical reference truths for relatively large rare events ($P_t \approx$ 1e-03,5e-04,1e-04). This is because using $N = 10^6$ samples, these are the sufficiently large P_t that SQCoV of the standard Monte Carlo estimator (2) is small enough so that we can guarantee at least 1 digit of accuracy. We will then later cross reference our CE code on these numerical references, and then use CE to build references for even smaller rare events that Monte Carlo cannot handle.

We see that as the dimension increases, the distribution of the QoI becomes significantly less right-skewed. This makes sense, smaller ξ correspond to larger f, which also has the physical intuition that the slower the heat conductivity in the domain, the larger heat we expect to find in steady state. Larger ξ would correspond to faster dissipation.

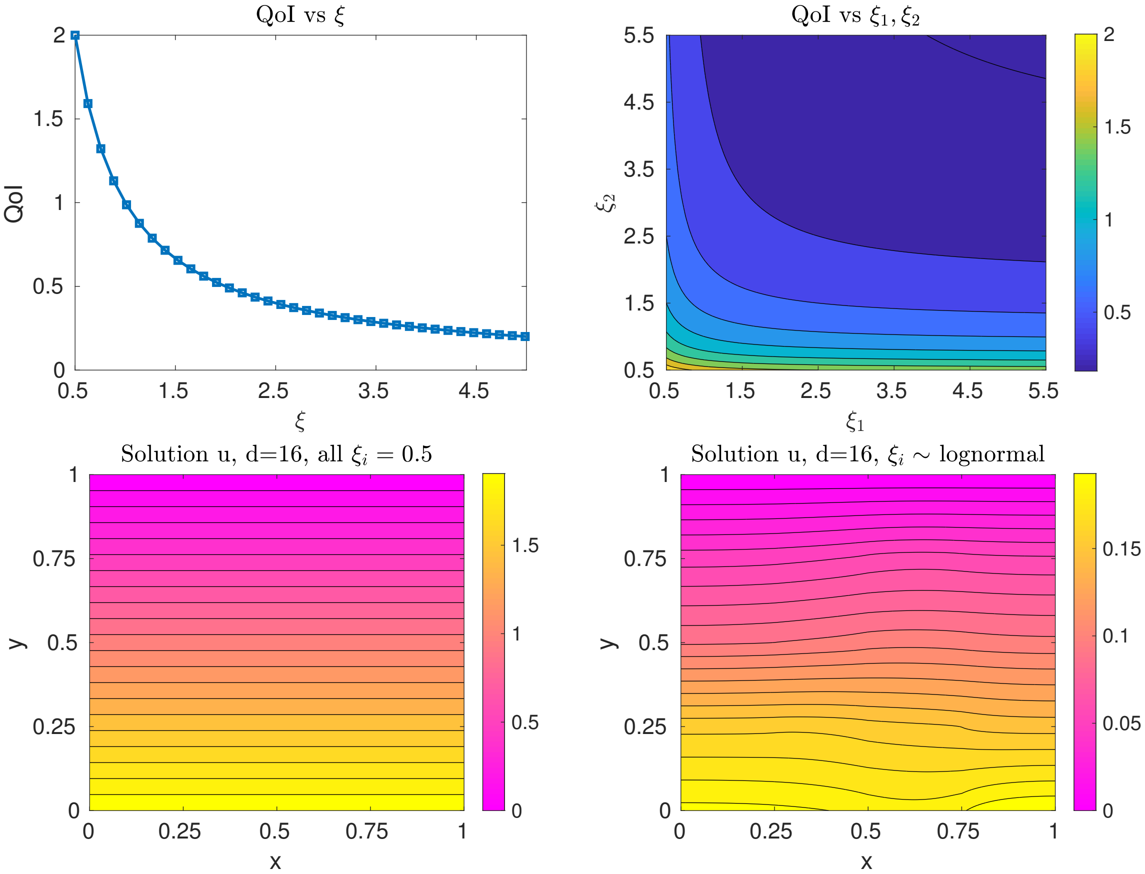


Figure 3: **Top:** Dependence of quantity of interest f on the parameter ξ in dimensions $d = 1, 2$. We see that smaller ξ corresponds to larger f, which agrees with physical intuition. **Bottom:** Dependence of solution spatially on ξ. Even in 16 dimensions, we see $\xi_i = 0.5$ for all i gives $u \equiv 1$, on the base, which would give $f = 2$. But on the right we see a more typical behavior with random ξ, that u has large spatial variation, and that the maximum of u anywhere in Ω is typically bounded away from 1. This supports that large f become more rare as d increases.

Looking in Figure 3, this is verified. Moreover, we see the region which corresponds to large QoI (i.e. small ξ) becomes smaller as dimension increases. In the $d = 2$ case, we see that area of the parameter space which corresponds to large QoI shrinks, and we can expect this behavior to continue as d increases, since the volume of space which yield larger QoI will become increasingly small.

We also see this intuition in the plot of the solution u in Figure 3 as well. We see in the case where $\xi_i \equiv 0.5$ that the solution is effectively a paraboloid which is constant in x and takes a maximum of

2 at $y = 0$, which gives the QoI as 2. But for a typical sample of ξ distributed as log-normal, we see spatial variation in the solution u, and we see that even though the max heat is typically at the base, it is significantly smaller, usually concentrated around 0.2, which is the mode we see in Figure 2.

2.2 Log-Normal vs. Uniform Underlying Densities

CE and MFCE can sometimes run into trouble whenever $\mathbf{1}\{f^{(\ell)}(\xi^{(i)}) \geq t\}p(\xi^{(i)})$ is zero for all samples $\xi^{(i)}$. Even if a sample $\xi^{(i)}$ is such that $f^{(\ell)}(\xi^{(i)})$ is a rare event it may still be rejected from the estimate due to p not being supported in that region. This leads to difficulties when trying to estimate a rare event from a target density that is uniform using a family of Gaussian biasing densities. Figure 4 shows the issue pictorially. For this reason we only consider the case where our target density p is Log-normal and is in our family of Log-normal biasing densities. Thus, the only way in which our estimators can fail now is if N is too small so that no rare events occur.

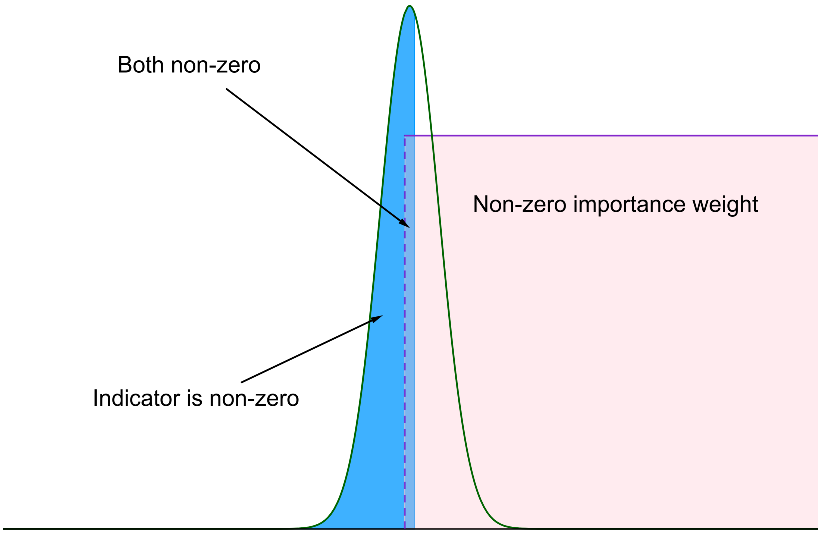


Figure 4: A visual example of why using importance sampling via Gaussians on rare uniform events is difficult. The dark shaded blue region represents samples above the threshold t. The lightly shaded pink region represents where the target density p has mass, so the overlap is the region of interest. We see that this region is increasingly thin, so even as we refine/sharpen our Gaussian, there is good odds that samples will either have 0 probability (blue but not pink) or be below the threshold (pink but not blue).

Another option that does not require p to be in the family of biasing densities is to take p as a Gaussian distribution, but our family still as Log-normal distributions. Because we need the diffusivity coefficient $k(x, \omega)$ to be positive, we choose the variance to be sufficiently small so that the probability that $k(x, \omega) \leq 0$ is well below machine precision. In this case, p is still supported on the entire real line so there will be no zero weights and we would see better performance than in the case where p is uniform.

3 Numerical Results

3.1 Monte Carlo vs. Cross-Entropy

Using the $N = 10^6$ samples generated earlier, when finding approximate thresholds in Section 2.1, we compute some references for rare events $P_t^{(L)}$ on the order of 1e-03, 5e-03, 1e-03. Again these were the sufficiently large rare events that the standard MC estimator could guarantee at least 1 digit accuracy on. Our references are $P_{0.40}^{(L)} \approx$ 3e-03, $P_{0.46}^{(L)} =\approx$ 5e-04, and $P_{0.50}^{(L)} \approx$ 1e-04 in $d = 1$. In $d = 4$ we get the references $P_{0.30}^{(L)} \approx$ 3e-03, $P_{0.32}^{(L)} \approx$ 8e-04, $P_{0.34}^{(L)} \approx$ 2e-04. And in $d = 16$ our references are $P_{0.255}^{(L)} \approx$ 1e-03, $P_{0.260}^{(L)} \approx$ 4e-04, and $P_{0.265}^{(L)} \approx$ 2e-04.

Looking at Figure 5, we see that our CE code replicates these references to 1 digit accuracy much faster than it took MC to build them. Moreover, we can use significantly fewer samples, i.e. $N = 10^3$ in $d = 1$, $N = 5 \times 10^3$ in $d = 4$, and $N = 5 \times 10^4$ in $d = 16$. Note that these values of N cannot be compared directly, since CE runs N model evaluations each iteration. But we see that even in CPU time CE outperforms MC.

$d = 1$	$t = 0.40$	$t = 0.46$	$t = 0.50$	CPU Time [s]
MC (reference)	$P_{0.40}^{(L)} \approx$ 3.13e-03	$P_{0.46}^{(L)} \approx$ 5.12e-04	$P_{0.50}^{(L)} \approx$ 1.45e-04	441
high-fidelity CE	$P_{0.40}^{(L)} \approx$ 3.35e-03	$P_{0.46}^{(L)} \approx$ 5.23e-04	$P_{0.50}^{(L)} \approx$ 1.42e-04	∼0.9
$d = 4$	$t = 0.30$	$t = 0.32$	$t = 0.34$	CPU Time [s]
MC (reference)	$P_{0.30}^{(L)} \approx$ 3.30e-03	$P_{0.32}^{(L)} \approx$ 8.28e-04	$P_{0.34}^{(L)} \approx$ 1.83e-04	436
high-fidelity CE	$P_{0.30}^{(L)} \approx$ 3.20e-03	$P_{0.32}^{(L)} \approx$ 8.00e-04	$P_{0.34}^{(L)} \approx$ 1.78e-04	∼5
$d = 16$	$t = 0.255$	$t = 0.260$	$t = 0.265$	CPU Time [s]
MC (reference)	$P_{0.255}^{(L)} \approx$ 1.05e-03	$P_{0.260}^{(L)} \approx$ 4.48e-04	$P_{0.265}^{(L)} \approx$ 1.75e-04	437
high-fidelity CE	$P_{0.255}^{(L)} \approx$ 1.03e-03	$P_{0.260}^{(L)} \approx$ 4.20e-04	$P_{0.265}^{(L)} \approx$ 1.68e-04	∼103

Figure 5: Numerical reference values computed using $N = 10^6$ samples for MC on the high-fidelity model $L = 5$. This guarantees at least 1 digit of accuracy for $P_t \geq$ 1e-04. Runtime for MC is for computing N samples. Runtime for high-fidelity CE is the average over runtime for each of the three thresholds. In $d = 1$, we used $N = 10^3$ samples, $d = 4$ we used $N = 5 \times 10^3$, and $d = 16$ we used $N = 5 \times 10^4$, chosen to be the minimum N needed to see clear 1 digit accuracy to MC references. Note smaller P_t and larger d require more loops in CE, which jumps the runtime.

We see that CE's performance is much better in lower dimensions than in higher dimensions. One reason for this is that importance sampling may have trouble in high dimensions when the region of interest in the target density has such small volume, which again connects to the fact that the distribution of the QoI becomes more concentrated as d increases.

3.2 Cross-Entropy for Small Rare Events

Seeing that CE is able to reproduce our references from MC, we now use CE on the high-fidelity model with large $N = 10^6$ to generate numerical truths for even smaller rare events. The thresholds were found by running CE with $N = 10^5$ multiple times and slowly decreasing the threshold until we found interesting P_t on the orders of 1e-06, 1e-08, and 1e-09. Moreover, this served as stability check for our code in that as we slowly increased the threshold t, we saw an appropriate slow and consistent decrease in P_t.

$d = 1$	$t = 0.60$	$t = 0.77$	$t = 0.90$	CPU Time [s]
	$P_{0.60}^{(L)} \approx$ 6.59e-06	$P_{0.77}^{(L)} \approx$ 4.28e-08	$P_{0.90}^{(L)} \approx$ 1.12e-09	∼1504
$d = 4$	$t = 0.40$	$t = 0.45$	$t = 0.49$	CPU Time [s]
	$P_{0.40}^{(L)} \approx$ 1.74e-06	$P_{0.45}^{(L)} \approx$ 3.16e-08	$P_{0.49}^{(L)} \approx$ 1.27e-09	∼1913
$d = 16$	$t = 0.285$	$t = 0.305$	$t = 0.315$	CPU Time [s]
	$P_{0.285}^{(L)} \approx$ 2.94e-06	$P_{0.305}^{(L)} \approx$ 3.25e-08	$P_{0.315}^{(L)} \approx$ 2.99e-09	∼2131

Figure 6: Numerical reference values computed using high fidelity CE, $N = 10^6$. Runtime for each dimension is the average over runtime for each of the three thresholds. Note smaller P_t and larger d require more loops in CE, which jumps the runtime.

In Figure 6, we have our high-fidelity CE references as $P^{(L)}_{0.60} \approx$ 7e-06, $P^{(L)}_{0.77} \approx$ 4e-08, and $P^{(L)}_{0.90} \approx$ 1e-09 in $d = 1$. For $d = 4$, $P^{(L)}_{0.40} \approx$ 2e-06, $P^{(L)}_{0.45} \approx$ 3e-08, and $P^{(L)}_{0.49} \approx$ 1e-09. And in $d = 16$, $P^{(L)}_{0.285} \approx$ 3e-06, $P^{(L)}_{0.305} \approx$ 3e-08, and $P^{(L)}_{0.315} \approx$ 3e-09. Again we see a similar behavior where the runtime increases noticeably in higher-dimensions.

3.3 Multifidelity Cross-Entropy Speedup

Using the reference values from high-fidelity CE in the previous section we now compare our MFCE and look at the relative speed up. For all the references in Figure 6, MFCE is able to deliver a significant speed up, both in the allowable N to be used, and in the overall runtime. Since our interest is in the smaller rare events, for $t = 0.90$ in $d = 1$, $t = 0.49$ in $d = 4$, and $t = 0.315$ in $d = 16$ (thresholds corresponding to P_t on the order of 1e-09), we ran both CE and MFCE 30 times to estimate the SQCoV with respect to the computed references, and to compute average runtime for a fixed N.

With $d = 1$, for MFCE we used $N = 5 \times 10^1, 5 \times 10^2, 5 \times 10^3, 5 \times 10^4$, and for CE we use $N = 10^2, 10^3, 10^4, 10^5$. With $d = 4$, for MFCE we used $N = 10^2, 10^3, 10^4, 10^5$, and for CE we used $N = 10^3, 5 \times 10^3, 10^4, 5 \times 10^4$. With $d = 16$, for both MFCE and CE we used $N = 5 \times 10^3, 10^4, 5 \times 10^4, 10^5$. Note that even though we may have used different N for MFCE and CE, in the end we are comparing SQCoV against runtime, not N.

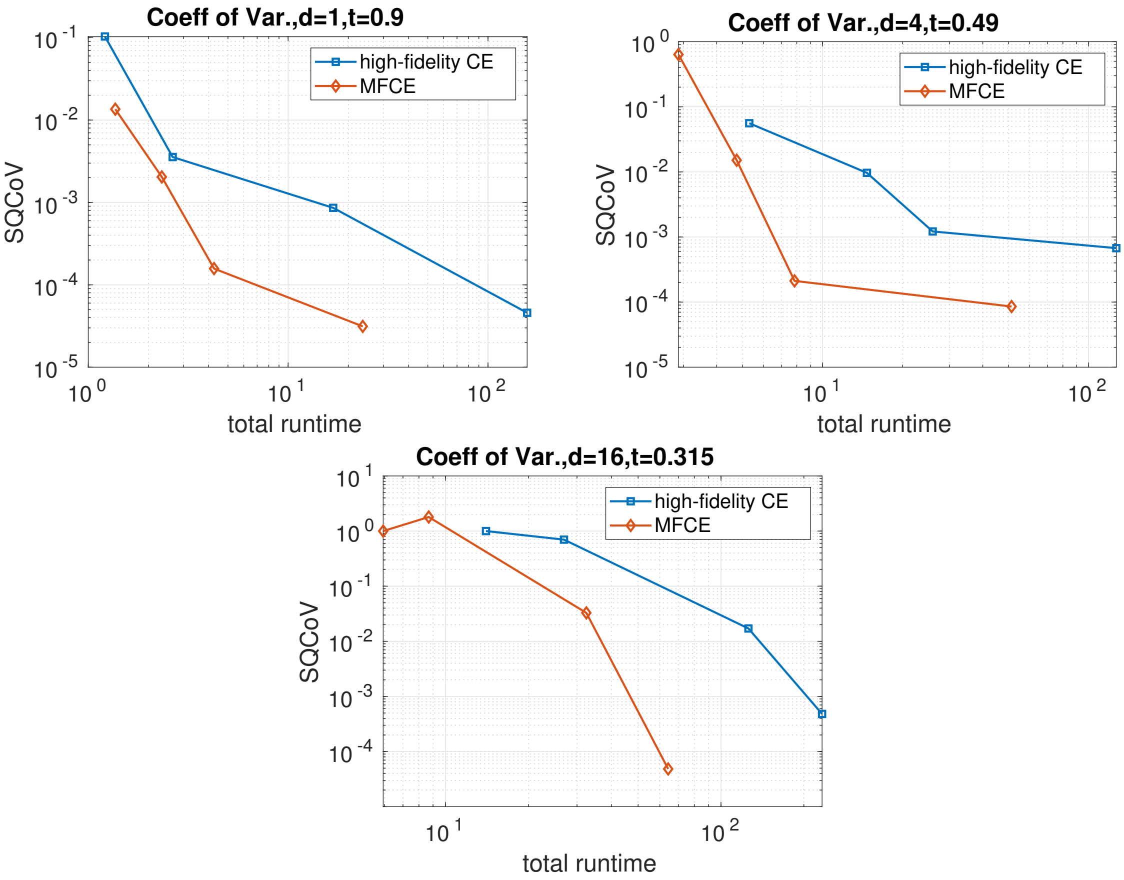


Figure 7: **Top Left:** SQCoV with respect to $P^{(L)}_{0.9} \approx$ 1.12e-09 in $d = 1$. **Top Right:** SQCoV with respect to $P^{(L)}_{0.49} \approx$ 1.27e-09 in $d = 4$. **Bottom:** SQCoV with respect to $P^{(L)}_{0.315} \approx$ 2.99e-09 in $d = 16$. In each case MFCE significantly outperforms CE, reaching smaller SQCoV in a much faster run time. The curves for CE essentially resemble those for MFCE, but shifted to the right (longer runtime) and up (worse error).

Looking at Figure 7, we see the speed up MFCE provides when estimate rare events on the order of 1e-09. In each dimension, MFCE provides a speed up on the order of a magnitude, getting smaller error, measured by SQCoV, in less time. The CE curves and MFCE curves are similar in shape, with CE giving worse errors and costing more time.

We note that in all the runs tested, MFCE shifts most of the work in constructing the biasing density to the lower level $\ell = 3$. That is, in counting the number of CE iterations, the $\ell = 3$ loop has the most, and the $\ell = 4, 5$ loops had only a single CE iteration each in every experiment ran. While the $\ell = 3$ loop often performs more CE iterations than high-fidelity single CE does, the $\ell = 3$ model is much faster to evaluate.

high-fidelity CE	$K \approx 8 - 11$
MFCE	$\ell = 3, K \approx 8 - 20$ $\ell = 4, K = 1$ $L = 5, K = 1$

Figure 8: Typical number K of CE iterations for both high-fidelity single CE and MFCE. Ranges include rare events on the order of 1e-06 down to 1e-09.

On average, we see high-fidelity single CE require anywhere from 8 to 11 iterations, whereas MFCE requires around anywhere from 6 to even 20 iterations at $\ell = 3$, but only 1 iteration each at $\ell = 4, 5$. Again, this is okay since the model $\ell = 3$ evaluations are so cheap, that the cost of doing many more steps to build an initially good biasing density is worth it as we save on expensive model evaluations.

4 Conclusion

Both CE and MFCE outperform standard Monte Carlo in regards to both accuracy and efficiency by using variance reduction. We have also seen that the performance of these methods is sensitive to the target density p and the choice of biasing distributions. Ideally one should choose a family that contains the target density but still allows a closed-form expression for the optimal parameters. Whenever a good family of biasing densities is chosen, these methods are able to estimate rare event probabilities as small as 10^{-9}, even in $d = 16$ dimensions. Monte Carlo on the other hand would require an intractable number of model evaluations. Furthermore, when cheaper model approximations are available, MFCE can offload much of the work onto lower levels and only needs a few CE iterations at the highest level resulting in a significant speedup.

The computations in this project were performed on the Courant servers, using four AMD Opteron 6272 CPUs, and utilizing 32 cores/workers to run the model evaluations in parallel.

References

[1] B. Peherstorfer, B. Kramer, and K. Wilcox. *Multifidelity Preconditioning of the Cross-Entropy Method for Rare Event Simulation and Failure Probability Estimation.* SIAM/ASA J. Uncertainty Quantification. 6(2), 737-761, 2018

[2] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method Third Edition.* Wiley Series in Probability and Statistics, 2017.