An Evaluation of Estimation Techniques for Probabilistic Verification

Mariia Vasileva and Paolo Zuliani

School of Computing, Newcastle University, Newcastle upon Tyne, UK
{m.vasileva2,paolo.zuliani}@newcastle.ac.uk

Abstract. Formal techniques for verifying stochastic systems (e.g., probabilistic model checking) do not generally scale well with respect to the system size. Therefore, simulation-based techniques such as statistical model checking are often used in practice. In this paper, we focus on stochastic hybrid systems and evaluate Monte Carlo and Quasi-Monte Carlo (QMC) methods for computing probabilistic reachability. We compare a number of interval estimation techniques based on the Central Limit Theorem (CLT), and we also introduce a new approach based on the CLT for computing confidence intervals for probabilities near the borders of the [0,1] interval. We empirically show that QMC techniques and our CLT approach are accurate and efficient in practice. Our results readily apply to any stochastic system and property that can be checked by simulation, and are hence relevant for statistical model checking.

1 Introduction

Verification techniques for stochastic systems such as probabilistic model checking can be very precise and can deal with a variety of stochastic systems (*e.g.*, discrete-time Markov chains [23], continuous-time Markov chains [6] and Markov decision processes [7]). However, as for standard, non-probabilistic model checking, these techniques suffer from the state explosion problem, which limits their applicability in many practical cases. Statistical model checking [24] is often used in practice on stochastic systems that exceed the limits of probabilistic model checking, or for which no formal technique is available (*e.g.*, *nonlinear* stochastic hybrid systems). In this paper, we focus on the probabilistic reachability problem for hybrid systems that depend on random parameters, which amounts to computing the probability that the system reaches a *goal* state.

Checking reachability in hybrid discrete/continuous systems is an undecidable problem for all but the simplest systems (timed automata) [2]. (See [12] for an up to date survey.) Formal verification of hybrid systems can include checking the satisfiability of formulas involving real variables, which is known to be an undecidable problem when, *e.g.*, trigonometric functions are involved [21]. The notion of δ-complete decision procedure was introduced to combat the undecidability of general sentences over the reals [13]. This approach has been extended to a bounded probabilistic reachability method with statistically valid enclosures [19]. Essentially, this technique amounts to computing (multi-dimensional)

integrals of indicator functions, which can be done in three possible ways: rigorous, Monte-Carlo (MC) and Quasi-Monte Carlo (QMC). The computational complexity of rigorous (*i.e.*, numerically precise) computation of integrals grows exponentially with respect to the number of dimensions [21]. This motivates the use of QMC methods, which are asymptotically more efficient than MC methods. While MC methods are based on the Law of Large Numbers and random sampling, QMC methods are based on *deterministic* sampling from so-called quasi-random sequences. A drawback of QMC methods is that their integration error is difficult to estimate in practice, so one instead estimates the error of *Randomised* Quasi-Monte Carlo methods via confidence interval techniques based on the Central Limit Theorem (CLT). However, a problem of many such techniques is that the actual coverage probability of the interval near the boundaries (0 and 1) can be poor [17,9].

To summarise, in this paper we make the following contributions:

- we compare several confidence interval techniques for estimating probabilities in MC and QMC methods, and we show that QMC methods are more efficient in general;
- we propose a simple but effective modification of the CLT interval for estimating probabilities close to 0 or 1, and we empirically show that it performs well in practice.

While we focus on hybrid systems and reachability, our results readily apply to any stochastic system and property whose truth can be checked by simulation.

Probabilistic Reachability. Hybrid systems provide a framework for modelling real-world systems that combine continuous and discrete dynamics [2]. In particular, parametric hybrid systems (PHS) [19] represent continuous and discrete dynamic behaviour dependent on parameters that remain unchanged during the system evolution. Such systems can flow, described by differential equations, and jump, described by difference equations or control graphs. (See Appendix A.1 for the formal definition of PHS). In this paper, we consider *stochastic* PHS, which introduce random parameters to an otherwise deterministic PHS. Bounded k-step reachability in stochastic PHS aims at finding the **probability** that for the given initial conditions the system reaches a goal state in k discrete transitions within a given finite time. It can be shown that this probability can be computed as an integral of the form $\int_G d\mathbb{P}$, where G denotes the set of all random parameter values for which the system reaches a goal state in k steps, and $\mathbb{P}$ is the probability measure of the random parameters [19].

2 Integral Estimation Methods

Monte Carlo Method. Consider an integrable function f, the integral $I = \int_a^b f(y)dy < \infty$, and a random variable U on $[a, b]$ with density φ. The expectation of $f(U)$ is $\mathbb{E}[f(U)] = \int_a^b f(y)\varphi(y)dy$. If U is uniformly distributed on $[a, b]$, then the integral becomes: $I = \int_a^b f(y)dy = (b - a)\mathbb{E}[f(U)]$. Now, if we take N samples

$\{u_1, \ldots, u_N\}$ from U and compute the sample mean $\frac{1}{N}\sum_{i=1}^{N} f(u_i)$, we obtain the MC estimate:

$$\int_a^b f(y)dy \approx (b-a)\frac{1}{N}\sum_{i=1}^{N} f(u_i). \tag{1}$$

The Strong Law of Large Numbers states that this approximation is convergent to I with probability 1 (for $N \to \infty$). The variance of the MC estimator (1) is:

$$Var(MC) = \int_a^b \ldots \int_a^b \left(\frac{1}{N}\sum_{i=1}^{N} f(u_i) - I\right)^2 du_1 \ldots du_N = \frac{\sigma_f^2}{N} \tag{2}$$

where σ_f^2 is the integrand variance, which is assumed to exist. In practice, the integrand variance is often unknown, and that is why the next estimation for the CI is instead used: $\widehat{\sigma}_f^2 = \frac{1}{N-1}\sum_{i=1}^{N}(f(u_i) - \frac{1}{N}\sum_{j=1}^{N} f(u_j))^2$, which enjoys the unbiasedness property $\mathbb{E}[\widehat{\sigma}_f^2] = \sigma_f^2$.

Quasi-Monte Carlo Method. QMC methods can be regarded as a deterministic counterpart to classical MC methods. Unlike MC integration, which uses estimates (1) with randomly selected points, QMC methods use (1) but select the points u_i *deterministically*. In particular, QMC techniques produce deterministic sequences of points that provide the best-possible spread over the integration domain. These deterministic sequences are often referred to as low-discrepancy sequences, of which the Sobol sequence [20] is a well-known example. An effective way to use the QMC method is by performing a change of variables to reduce the integration to the $[0, 1]$ domain. When we need to integrate over a large domain $[a, b]$, that avoids multiplying the QMC estimate by a large factor $(b - a)$ as required by (1).

A QMC advantage with respect to MC is that its error is $O(1/N)$, while the MC error (see Eq. (2)) is $O(1/\sqrt{N})$, where N is the sample size. The Koksma-Hlawka inequality bounds the error of QMC estimates, but in practice it is very hard to estimate [14], thereby hampering the use of QMC methods (see Appendix A.2). As such, other methods for estimating the QMC error need to be developed. For example, Ermakov and Antonov [5] have recently introduced the *Qint* method for QMC variance estimation, based on a set of random quadrature formulas (see below and Appendix A.3).

Randomised Quasi-Monte Carlo. As discussed earlier, the practical application of QMC is limited by the difficulty of computing an estimate of the integration error. However, allowing randomisation into the deterministic QMC procedure enables constructing confidence intervals. A *Randomised* QMC (RQMC) procedure can be described as follows. Suppose that $\mathfrak{X} = \{x_1, ..., x_n\}$ is a deterministic low-discrepancy set. By means of a transformation $\tilde{\mathfrak{X}} = \Gamma(\mathfrak{X}, \epsilon)$ a finite set $\tilde{\mathfrak{X}}$ is generated by the random variable ϵ with the same quasi-random properties of the set $\mathfrak{X}$ (see Figure 1). For a randomised set $\tilde{\mathfrak{X}}_i$ of size n we construct a RQMC

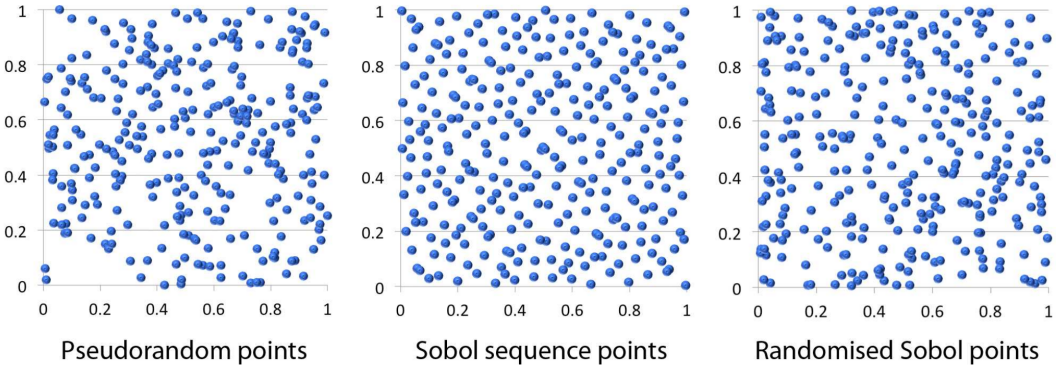


Fig. 1: Uniform pseudorandom, Sobol sequence and randomised Sobol sequence points (obtained by transformation $\Gamma = (\mathfrak{X} + \epsilon) \mod 1$, where ϵ is a random sample from MC sequence and $\mathfrak{X}$ is low-discrepancy sample from Sobol sequence) distribution in the 2-dimensional unit space. The comparison is based on the first 300 points of sequences.

estimate similar to (1):

$$RQMC_j = \frac{1}{n} \sum_{i=1}^{n} f(\tilde{\mathfrak{X}}_{i,j}) \tag{3}$$

for $0 < j \leqslant r$, where r is the total number of different pseudo-random sequences. Then, we take their average for overall RQMC estimation (3):

$$RQMC = \frac{1}{r} \sum_{j=1}^{r} RQMC_j \tag{4}$$

which is then built out of rn samples in total. If we choose the Γ transformation in such a way that each of the estimates $RQMC_j$ has the unbiasedness property, *i.e.*, $\mathbb{E}[RQMC_j] = I$ for all j, (*e.g.*, $\Gamma = (\mathfrak{X} + \epsilon) \mod 1$), then the estimator (4) will also be unbiased, *i.e.*, $\mathbb{E}[RQMC] = I$. By independence of the samples used in (3) and (4), we have that:

$$Var(RQMC) = \frac{Var(RQMC_j)}{r}.$$

Thus, we have the following variance estimation: $\widehat{Var}(RQMC) = \frac{1}{r(r-1)} \sum_{j=1}^{r} \left(RQMC_j - RQMC\right)^2$.

3 Confidence Interval Estimation and Error Analysis

In the following we shall use the notation below:

– $\tilde{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$ - sample mean;
– $C_a = Quant(1-\frac{a}{2})$ - inverse cumulative distribution function (quantile function) of a normal random variable with mean 0 and standard deviation 1; parameter a defines the confidence level at $1-a$;
– $\hat{p} = n_s/n$ - the binomially-distributed proportion, where: n_s - number of successes and n_f - number of failures in a Bernoulli trial process; n - total number of Bernoulli trials;
– $\hat{q} = 1-\hat{p}$, and CI = confidence interval.

3.1 Intervals Based on the Standard CLT Interval

Modified Central Limit Theorem (CLT) interval. First, we consider the case when the sample x_i is extracted from the normal distribution $N(\mu, \sigma^2)$ with unknown mean μ and known variance σ^2. Here, μ can be approximated by the sample mean: $\mu \approx \tilde{X}$. To clarify this approximation, the standard CI for μ with confidence level $1-a$ is:

$$CI_{CLT} = \left(\tilde{X} - C_a\frac{\sigma}{\sqrt{n}}; \tilde{X} + C_a\frac{\sigma}{\sqrt{n}}\right). \tag{5}$$

In practice, the variance σ^2 is often unknown, but one can use the same CI by replacing σ with the sample standard deviation $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \tilde{X})^2}$. This method is widely used for estimating the distribution of binomially-distributed proportions. A number of works (*e.g.*, [8,9,10]) note that the CI_{CLT} approximation can be poor when applied to Bernoulli trials with $\hat{p}$ close to 0 or 1. Indeed, in the Bernoulli case, when $\hat{p}$ is 0 (or 1) the CLT interval (5) cannot be constructed, since $s = 0$ (recall that the sample standard deviation s substitutes σ, which is most often unknown). In order to address this problem, we simply overapproximate the sample standard deviation with $\frac{1}{n^2}$ if $\hat{p}$ is equal to 0 (or 1).

Wilson interval. It was introduced by Wilson in 1927 in his fundamental work [11] and uses the inversion of the CLT interval. The interval is:

$$CI_W = \left(\frac{n_s + \frac{C_a^2}{2}}{n + C_a} - \frac{C_a\sqrt{n}}{n + C_a^2}\sqrt{\hat{p}\hat{q} + \frac{C_a^2}{4n}}; \frac{n_s + \frac{C_a^2}{2}}{n + C_a} + \frac{C_a\sqrt{n}}{n + C_a^2}\sqrt{\hat{p}\hat{q} + \frac{C_a^2}{4n}}\right) \tag{6}$$

This interval has some obvious advantages - it can not exceed probability boundaries, and it can be easily calculated even if $\hat{p}$ is 0 or 1. At the same time, CI_W has downward spikes when $\hat{p}$ is close to 0 and 1, because it is formed by an inverted CLT approximation.

Agresti-Coull interval. This method was introduced by Agresti and Coull in 1998 [1]. One of the most interesting features of this CI is that it makes a crucial assumption about n_s and n_f. This interval formally adds two successes and two

failures to the obtained values in case of 95% confidence level and then uses the CLT method. The interval can be constructed as follows:

$$CI_{AC} = \left(\tilde{X} - \frac{1}{n + C_a^2}(n_s + \frac{1}{2}C_a^2); \tilde{X} + \frac{1}{n + C_a^2}(n_s + \frac{1}{2}C_a^2) \right) \tag{7}$$

Additionally, this interval can be modified by using the center of the Wilson interval (6) in place of $\hat{p}$:

$$CI_{AC_W} = \left(\frac{n_s + \frac{C_a^2}{2}}{n + C_a} - C_a\sqrt{\hat{p}\hat{q}(n + C_a^2)}; (\frac{n_s + \frac{C_a^2}{2}}{n + C_a} - C_a\sqrt{\hat{p}\hat{q}(n + C_a^2)} \right) . \tag{8}$$

Logit interval. The Logit interval is based on a transformation of the standard interval [10]. It uses the empirical logit transformation: $\lambda = ln(\frac{\hat{p}}{1-\hat{p}}) = ln(\frac{n_s}{n-n_s})$. The variance of λ is: $\widehat{Var}(\lambda) = \frac{n}{n_s(n-n_s)}$ and the Logit interval is estimated as:

$$CI_L = \left(\frac{e^{\lambda_L}}{1 + e^{\lambda_L}}, \frac{e^{\lambda_U}}{1 + e^{\lambda_U}} \right) \tag{9}$$

where the lower bound transformation is $\lambda_L = \lambda - C_a\sqrt{\widehat{Var}(\lambda)}$ and the upper bound transformation is $\lambda_U = \lambda + C_a\sqrt{\widehat{Var}(\lambda)}$.

Anscombe interval. This interval was proposed by Anscombe in 1956 [4] and is based on the Logit interval (9). The key difference is in λ and $\widehat{Var}(\lambda)$ estimation, where λ is defined as $\lambda = ln(\frac{n_s+\frac{1}{2}}{n-n_s+\frac{1}{2}})$ and the variance is $\widehat{Var}(\lambda) = \frac{(n+1)(n+2)}{n(n_s+1)(n-n_s+1)}$. On this basis, the Anscombe interval CI_{Anc} is estimated in the same way as Logit interval (9).

Arcsine interval. It uses a variance-stabilising transformation of $\hat{p}$. In 1948, Anscombe introduced an improvement [3] for achieving better variance stabilisation by replacing $\hat{p}$ to $p^\dagger = \frac{n_s+3/8}{n+3/4}$, obtaining

$$CI_{Arc} = \left(\sin(\arcsin(\sqrt{p^\dagger}) - \frac{C_a}{2\sqrt{n}})^2, \sin(\arcsin(\sqrt{p^\dagger}) + \frac{C_a}{2\sqrt{n}})^2 \right) . \tag{10}$$

3.2 Alternative Intervals Based on the Beta-Function

Bayesian interval. This method is based on the assumption that the (unknown) probability p to estimate is itself a random quantity [25]. The Bayesian interval is also called *credible*, because it is based on the posterior distribution of the unknown quantity computed by using its prior distribution and the Bayes theorem. The prior distribution can be constructed by means of the *Beta* distribution. If

p has a prior distribution $Beta(\alpha, \beta)$ then after n Bernoulli trials with n_s successes, p has posterior distribution $Beta(n_s + \alpha, n - n_s + \beta)$. We can construct a Bayesian equal-tailed interval by the formula:

$$CI_B = \left(Beta^{-1}(\frac{a}{2}, n_s + \alpha, n - n_s + \beta), Beta^{-1}(1 - \frac{a}{2}, n_s + \alpha, n - n_s + \beta)\right) \tag{11}$$

where, $Beta^{-1}(a, \alpha, \beta)$ is the inverse of the cumulative distribution function of $Beta(\alpha, \beta)$. The probability density function of the $Beta$ distribution is $f(x; \alpha, \beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$, where $0 \leq x \leq 1$, $\alpha, \beta > 0$ and B is the beta function. In our experiments we used $\alpha = \beta = 1$, which gives the uniform distribution.

Jeffreys interval. The Jeffreys interval is a Bayesian interval and uses the Jeffreys prior [15], which is a non-informative prior given by the $Beta$ distribution with parameters $(\frac{1}{2}, \frac{1}{2})$. We can form Jeffreys' equal-tailed interval by (11) with parameters $(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$.

Clopper-Pearson interval. This method was introduced by Clopper and Pearson in 1934 [8] and is based on the inversion of the binomial test, rather than on approximations. The Clopper-Pearson interval is:

$$CI_{CP} = \left(Beta^{-1}(\frac{a}{2}, n_s, n - n_s + 1), Beta^{-1}(1 - \frac{a}{2}, n_s + 1, n - n_s)\right) \ . \tag{12}$$

The interval states that the computed coverage probability is always above or equal to the $1 - a$ confidence level. In practice, it can be achieved in cases when n is large enough, while in general the actual coverage can exceed $1 - a$. We can conclude from Eq. (12) that due to the absence of the α and β parameters, a tighter CI can be achieved only by increasing the number of trials. We report this interval only for completeness, although we will not use it in our experiments as it is similar to the Bayesian and Jeffreys intervals.

4 Results

We evaluate confidence interval (CI) estimation methods based on the CLT interval with the RQMC and MC techniques and the Bayesian CI estimation method with the MC technique. In the RQMC case $r = 10$ quasi-random sequences were obtained by changing the pseudo-random points ϵ of the transformation $\Gamma = (\mathfrak{X} + \epsilon) \mod 1$, while the Sobol sequence points $\mathfrak{X}$ remained the same. In the MC case we used the same 10 pseudo-random points sequences that were used for RQMC calculations. (The high confidence levels used (up to 0.99999) motivates our choice of ten repetitions.) The samples used in Section 4.1 and 4.2 were obtained by sampling Bernoulli's, *i.e.*, no model simulation was performed.

4.1 Border Probability Cases

Intervals based on CLT and Bayesian interval. The comparison of the different CI estimation techniques for low probability cases is presented in Figure 2. It

shows that all intervals except the Arcsin interval (10) (see plot $c = 0.99$ of Figure 2 for probability=0.001) contain the true probability value. The Bayesian method tends to overestimate the true probability values as they increase while CI_{CLT} tends to underestimate them. Also, it is interesting to note that the most accurate center value is returned by the Agresti-Coull interval. The reason why CI_{CLT} tends to include the true probability value near the upper bound of the interval is directly related to the number of samples. As shown in Figure 2, for true probability values between 0.007 - 0.01, the CI_{CLT} center is moving up evenly to the true probability value with the increase of the confidence value. For the other true probability values (0.001-0.006) this drift remains, but it can not be seen from the Figure because of the small difference in the number of samples for all confidence levels, which causes the CI center to move wave-like.

In Figure 3 we plot the number of samples that the CI estimation techniques used to return intervals for four confidence levels. It can be clearly seen that when increasing the confidence level the CIs based on the CLT interval outperform the Bayesian CI. The plot with $c = 0.99999$ in Figure 3 illustrates that the best techniques in the number of samples are (best to worst): CI_{CLT}, *Qint*, CI_{Arc}, CI_W, CI_L, CI_{Ans}, CI_{AC_W} and CI_B. The CI_L and CI_{Anc} techniques always show almost the same results near the bounds, because of the modification of the CI_L. Initially, CI_L is not able to deal with probability values near the bounds according to its λ formula (see Section 3.1). It has been modified to use the Anscombe estimation formula in cases when $\hat{p} = 0$ or $\hat{p} = 1$. It is also important to note that the difference in the number of samples between CI_{CLT}, CI_{Arc} and CI_B for extreme probability cases is relevant. For example in the plot with $c = 0.9999$ of Figure 3 the number of samples used to obtain interval for $p = 0.005$ equals to 1,078 for CI_{CLT}, 2,662 for the CI_{Arc} and 4,440 for CI_B.

Summarising, for probability values near the bounds (0 or 1) our modified CLT method achieves better results with fewer samples in comparison with the other techniques. For probability values away from the bounds, the CLT, Wilson, Agresti–Coull, Logit and Anscombe methods are all very similar, and so for such probabilities we come to the conclusion that the CLT interval should be recommended, due to its simplest form. Meanwhile for smaller sample sizes, the CI_{CLT} is strongly preferable to the others and so might be the choice where sampling cost is paramount.

Qint method results. In Figure 2 and Figure 3 we also plotted the results of the *Qint* algorithm (see Section 2). In our research we used *Qint* with $n = k \times 2^s$, where $k = 2$. These parameters were used to form n points of the Sobol sequence x_i with numbers $i \in I_{k,s} = \{1, 2, ..., k \times 2^s\}$. These parameters were chosen on the basis of the original study of the *Qint* method [5]. As mentioned in Section 2, *Qint* uses a cubature randomization method and provides an integral estimation variance (see Appendix A.3) that we used to obtain a CI by our modified CLT interval (5).

In Figure 2 we display the *Qint* intervals for border probability values. We can see from the plots that the *Qint* CI always contains the true probability value. At the same time for all confidence levels from 0.99 to 0.99999 and for

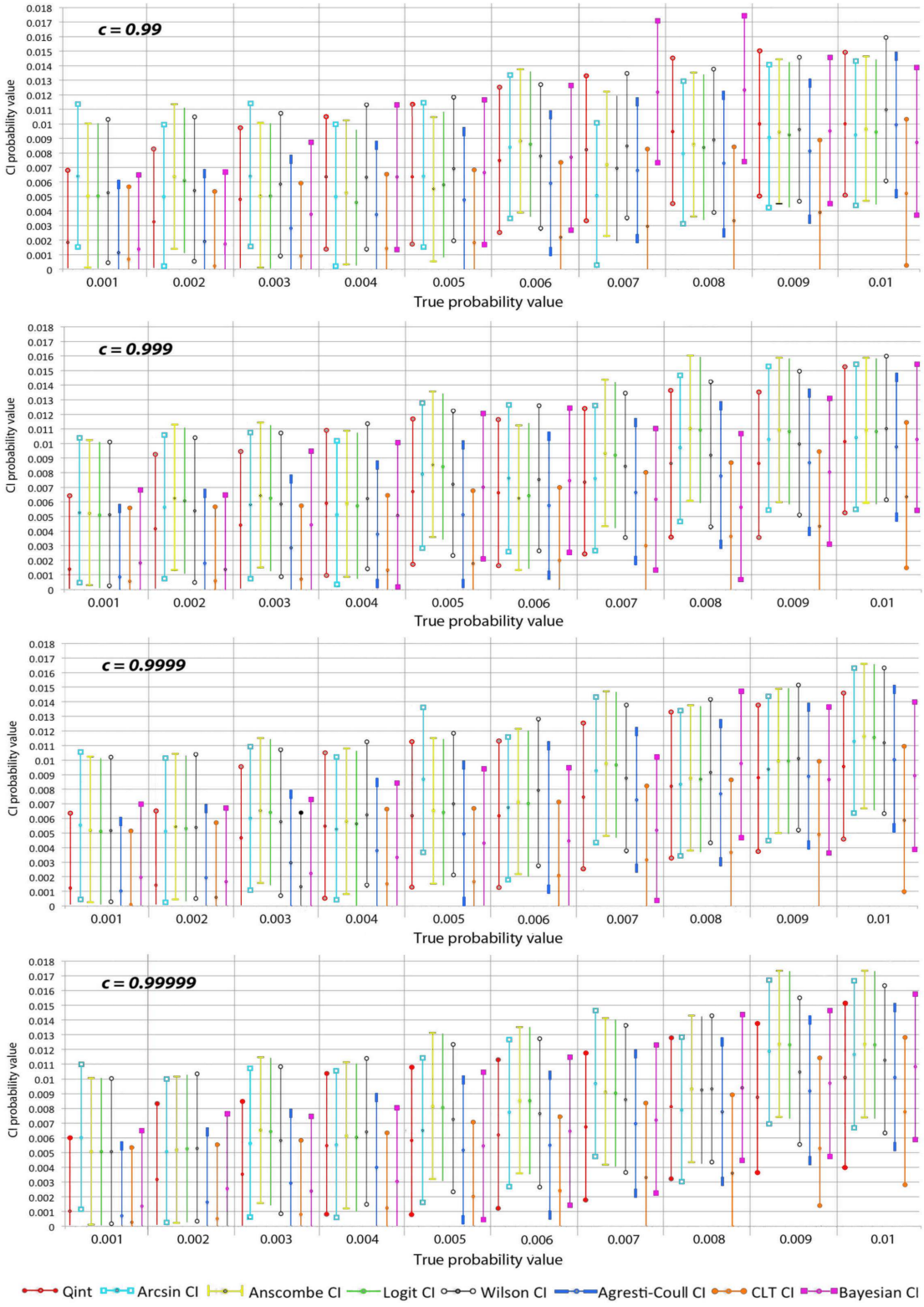


Fig. 2: Comparison of confidence intervals for probability values near 0, interval size equal to 10^{-2} and $\boldsymbol{c}$ - confidence level.

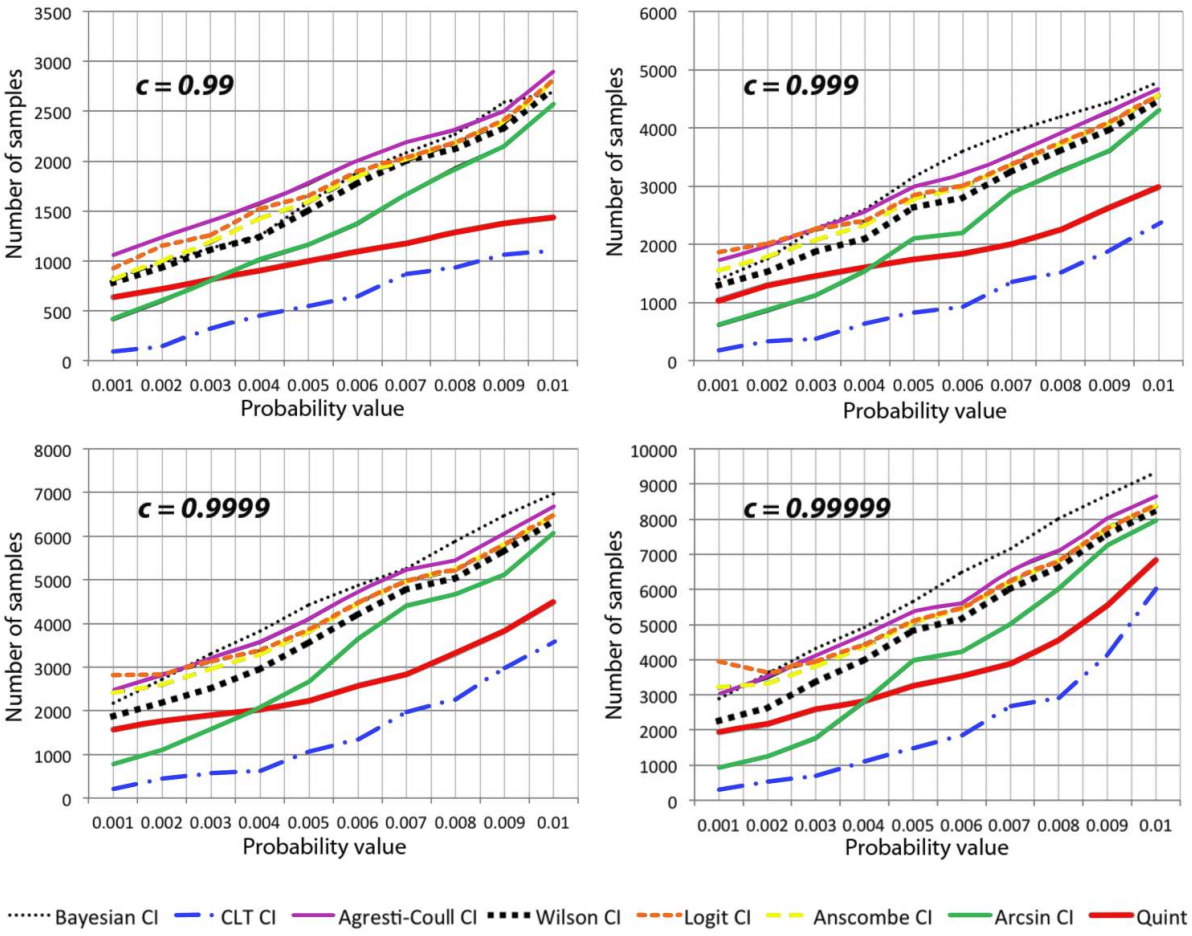
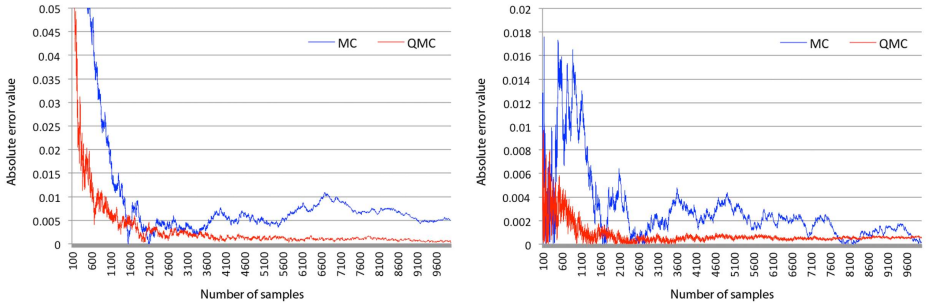


Fig. 3: Comparison of sample size for probability values near 0, interval size equal to 10^{-2} and $\boldsymbol{c}$ - confidence level.

true probability values 0.006-0.01, *Qint* shows better centration than CI_B and CI_{CLT}. For example, the greatest difference between the *Qint* CLT center result and the true probability values is 0.00245 for $c = 0.99$ (p=0.004), while this difference for CI_B reaches 0.00518 for $c = 0.99$ (p=0.007). We can see in Figure 3 that, as expected, *Qint* uses fewer samples than other CIs but CI_{CLT}. Our modification allows the *Qint* algorithm to return intervals even if $n_s = 0$, which significantly decreases the final sample size.

The fact that with the chosen parameters *Qint* can not outperform our modified CI_{CLT} leads us to the conclusion that our use of the standard deviation formula with $\frac{1}{n^2}$ lower bound is a rather effective and simple solution. However,



(a) Model: Collision advanced, type - max.

(b) Model: Bad, type - max2.

Fig. 4: MC (blue line) and QMC (red line) absolute error with respect to the number of samples.

the deep range of the possible parameters variation of the *Qint* algorithm lead us to believe that further research towards their comparison is needed.

4.2 MC and QMC Error Comparison

As mentioned in Section 2, the aymptotic QMC advantage over MC on the integration error holds in general. When the true probability value is extremely close to 0 (*i.e.*, $n_s = 0$ is obtained), we have that both the MC and QMC produce a 0 estimate and hence their error equals the true probability value. Also, the chaotic coverage properties of the MC method are far more persistent than they are appreciated. The chaotic behaviour does not disappear even when n is quite large and the true probability p is not near the boundaries. For instance, in Figure 4 (a) it is visible that even when n is quite large (*i.e.*, tends to 10,000 samples) the actual error value of the MC method reaches 0.005. Hence we conclude that MC-based CI estimation techniques can be misleading and defective in several respects and their *point estimates* should be used with care [9].

A notable phenomenon, which was noticed for both the MC and QMC methods is that the actual error contains non-negligible oscillations as both p and n vary. There exist some "unlucky" pairs (p, n) such that the corresponding absolute error is much greater than the results for smaller n. The phenomenon of oscillation is both in n, for fixed p, and in p, for fixed n. Furthermore, drastic changes in coverage probability can occur in nearby p for fixed n and in nearby n for fixed p [9]. We can see it on the simple example in Figure 4 (b). However, the same figure shows that error of QMC is more "stable" than the MC error.

4.3 Hybrid Systems Results

The results in this Section have been obtained via the ProbReach tool [18] for computing bounded reachability in stochastic parametric hybrid systems. Five

models[1], including nonlinear systems, were chosen for our experiments to give use cases representative of real applications. Based on our model set, we provide in Table 1 a comparison of the CIs described in Section 3. The true probability value $\boldsymbol{P}$ is either an analytically computed single probability value or a rigorously computed absolute (non-statistical) interval [18].

As it can be seen in Table 1, all the intervals computed overlap with each other. The key difference in the interval size can be found in the results of the Bad model Type min and the Collision (Basic) model Type min. From the results we can conclude that the true probability value is very close to 0. This allows the Bayesian, CLT and Agresti-Coull methods to form intervals that are actually half of the required width 10^{-2}, while the other techniques return fully sized intervals. That happens because CI_B is using the posterior distribution to form the interval, which is always defined on the whole [0,1] interval. At the same time, the CI_{CLT} and CI_{AC_W} calculations of the mean value are quite close to zero, thus cutting out the negative part of the interval. This trend holds for all probability values within $[0, 0.001]$. Table 1 also shows that with the increase of the confidence level the interval's precision grows, which in turn is directly related to the usage of the inverse cumulative distribution function for normal random variable with given confidence level in formulas for CI_{CLT} (5), CI_W (6), CI_{AC_W} (8) and CI_{Arc} (10). It also results in the increase of the sample size for CI_L and CI_{Anc}.

The comparison of the obtained intervals (see Table 1) with the true probability value or interval $\boldsymbol{P}$ shows that all CIs contain the single probability values but CI_{Acr} (see Bad type min model of Table 1), and all CIs overlap with the true probability intervals. We also note that the true probability intervals of the Collision Extended, Collision Advanced, and Anesthesia models contain all confidence intervals for all confidence levels. The Collision Basic and Deceleration models' true probability intervals do not contain CIs due to their size (< 0.01). The original *Qint* algorithm was not able to provide results for the Bad type min and Collision Basic type min models, because of the very small probability involved (4×10^{-7} and [0, 0.00201]) it could not detect $n_s > 0$ for the chosen confidence level and interval size. Therefore, *Qint* was used in conjunction with our CLT method described in Section 3.1. In conclusion, all the CI techniques examined returned reliable intervals.

Table 2 reports the number of samples required to compute the CIs obtained via ProbReach (for confidence 0.99999 — results for other confidence levels can be found in Appendix A.4). As it was noted earlier for Figure 4, the number of samples needed for CI computation grows from the bounds to the center of the [0,1] interval (this is because the variance of a Bernoulli is largest at $p = 0.5$). The most important outcome is that all CIs (except CI_{Arc}) show better result in number of samples with respect to CI_B. Overall, *Qint* showed the best result for every model, closely followed by CI_{CLT}: from Table 2 we see that *Qint* used on average between 1,850 and 24,802 fewer samples than other CI techniques.

[1] Available at https://github.com/dreal/probreach/tree/master/model

Confidence level c=0.99										
Model	**Type**	P	CI_B	CI_{CLT}	CI_{AC_W}	CI_W	CI_L	CI_{Ans}	CI_{Arc}	$Qint$
Good	max	0.1	[0.09671, 0.10671]	[0.09564, 0.10564]	[0.09632, 0.10632]	[0.09574, 0.10574]	[0.09575, 0.10575]	[0.09577, 0.10577]	[0.09559, 0.10559]	[0.09147, 0.10147]
	min	0.1	[0.09529, 0.10529]	[0.0956, 0.1056]	[0.09666, 0.10666]	[0.09679, 0.10679]	[0.09678, 0.10678]	[0.0968, 0.1068]	[0.09639, 0.10639]	[0.09164, 0.10164]
Bad	max	0.95001	[0.94416, 0.95416]	[0.94495, 0.95493]	[0.94422, 0.95422]	[0.94397, 0.95397]	[0.94396, 0.95396]	[0.94392, 0.95392]	[0.94735, 0.95735]	[0.94459, 0.95459]
	max2	0.88747	[0.8825, 0.8925]	[0.88028, 0.89028]	[0.88031, 0.88031]	[0.88019, 0.89019]	[0.8803, 0.8902]	[0.88019, 0.89019]	[0.88325, 0.89325]	[0.88136, 0.89136]
	min	4×10^{-7}	[0, 0.00525]	[0, 0.005]	[0, 0.00483]	[0, 0.00955]	[0.00005, 0.00959]	[0.00005, 0.00959]	[0.00131, 0.00959]	[0,0.005]
Deceleration	max	[0.08404, 0.08881]	[0.08471, 0.09471]	[0.08802, 0.09802]	[0.08817, 0.09817]	[0.08685, 0.09685]	[0.08614, 0.09614]	[0.0863, 0.0963]	[0.08963, 0.09932]	[0.08852, 0.09852]
	min	[0.04085, 0.04275]	[0.03835, 0.04835]	[0.03861, 0.04861]	[0.03854, 0.04854]	[0.03884, 0.04884]	[0.03886, 0.04886]	[0.0389, 0.0489]	[0.03873, 0.04873]	[0.03337, 0.04337]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96371, 0.97381]	[0.96873, 0.97873]	[0.9684, 0.9784]	[0.96851, 0.97851]	[0.96875, 0.97875]	[0.96853, 0.97853]	[0.96851, 0.97851]	[0.96301, 0.97301]
	min	[0, 0.00201]	[0, 0.00525]	[0, 0.005]	[0, 0.00483]	[0, 0.00955]	[0.00005, 0.00959]	[0.00005, 0.00959]	[0.00131, 0.00959]	[0,0.005]
Collision (Extended)	max	[0.35751, 0.49961]	[0.42267, 0.43675]	[0.42418, 0.4342]	[0.42187, 0.43187]	[0.42345, 0.43345]	[0.42463, 0.43463]	[0.42457, 0.43457]	[0.42385, 0.43385]	[0.42342, 0.43342]
	min	[0.04296, 0.06311]	[0.0482, 0.0582]	[0.04772, 0.05772]	[0.04785, 0.05785]	[0.04823, 0.05823]	[0.04812, 0.05812]	[0.0481, 0.0581]	[0.04757, 0.05772]	[0.04618, 0.05618]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.2072, 0.2172]	[0.20873, 0.21872]	[0.21872, 0.2185]	[0.20854, 0.21854]	[0.20854, 0.21854]	[0.20855, 0.21855]	[0.20111, 0.21111]	[0.20167, 0.21166]
	min	[0.02471, 0.05191]	[0.02631, 0.03631]	[0.03045, 0.04045]	[0.03016, 0.04016]	[0.03001, 0.04]	[0.03001, 0.04]	[0.03016, 0.04016]	[0.03164, 0.04164]	[0.0304, 0.0404]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01361, 0.02361]	[0.01339, 0.02332]	[0.01374, 0.02374]	[0.01373, 0.02373]	[0.01318, 0.02318]	[0.01311, 0.02311]	[0.01592, 0.02592]	[0.01815, 0.02815]
Confidence level c=0.99999										
Model	**Type**	P	CI_B	CI_{CLT}	CI_{AC_W}	CI_W	CI_L	CI_{Ans}	CI_{Arc}	$Qint$
Good	max	0.1	[0.09499, 0.10499]	[0.09378, 0.10378]	[0.09386, 0.10386]	[0.09389, 0.10389]	[0.09391, 0.10391]	[0.09392, 0.10392]	[0.09405, 0.10405]	[0.09512, 0.10512]
	min	0.1	[0.09419, 0.10419]	[0.09667, 0.10667]	[0.09668, 0.10668]	[0.09677, 0.10677]	[0.09671, 0.10671]	[0.09679, 0.10679]	[0.09675, 0.10675]	[0.09525, 0.10525]
Bad	max	0.95001	[0.94525, 0.95525]	[0.94579, 0.95579]	[0.94564, 0.95564]	[0.94548, 0.95548]	[0.94545, 0.95545]	[0.94543, 0.95543]	[0.94735, 0.95735]	[0.94543, 0.95543]
	max2	0.88747	[0.88215, 0.89215]	[0.88055, 0.89055]	[0.88057, 0.89057]	[0.88046, 0.89046]	[0.88046, 0.89046]	[0.88046, 0.89046]	[0.88325, 0.89325]	[0.88052, 0.89052]
	min	4×10^{-7}	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]	[0, 0.00992]	[0, 0.00992]	[0.00445, 0.0139]	[0,0.005]
Deceleration	max	[0.08404, 0.08881]	[0.08613, 0.09613]	[0.08624, 0.09624]	[0.08312, 0.09312]	[0.08725, 0.09725]	[0.08725, 0.09725]	[0.08726, 0.09726]	[0.08746, 0.09746]	[0.08737, 0.09735]
	min	[0.04085, 0.04275]	[0.03514, 0.04514]	[0.03919, 0.04919]	[0.03918, 0.04918]	[0.03942, 0.04942]	[0.03943, 0.04943]	[0.03944, 0.04944]	[0.039, 0.049]	[0.03377, 0.04377]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96359, 0.97359]	[0.96241, 0.97241]	[0.96767, 0.9767]	[0.96892, 0.96892]	[0.96689, 0.97589]	[0.96683, 0.97583]	[0.96863, 0.97863]	[0.96462, 0.97462]
	min	[0 , 0.00201]	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]	[0, 0.00992]	[0, 0.00992]	[0.00445, 0.0139]	[0,0.005]
Collision (Extended)	max	[0.35751, 0.49961]	[0.42651, 0.43652]	[0.42719, 0.43724]	[0.42757, 0.43757]	[0.42656, 0.43656]	[0.41774, 0.42774]	[0.41779, 0.42779]	[0.42745, 0.43745]	[0.42875, 0.43875]
	min	[0.04296, 0.06311]	[0.04979, 0.05979]	[0.04766, 0.05766]	[0.04764, 0.05764]	[0.04748, 0.05748]	[0.04745, 0.05745]	[0.04776, 0.05776]	[0.05776, 0.05673]	[0.04576, 0.05576]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.20515, 0.21519]	[0.20558, 0.21563]	[0.20533, 0.21533]	[0.20531, 0.21531]	[0.20547, 0.21547]	[0.20547 0.21547]	[0.20385, 0.21385]	[0.20453, 0.21453]
	min	[0.02471, 0.05191]	[0.03011, 0.04015]	[0.02902, 0.03902]	[0.02954, 0.03945]	[0.03956, 0.04956]	[0.03861, 0.04861]	[0.03887, 0.04887]	[0.0363, 0.04363]	[0.03031, 0.04031]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01284, 0.02284]	[0.01513, 0.02511]	[0.01623, 0.02623]	[0.01545, 0.02545]	[0.01557, 0.02557]	[0.01562, 0.02562]	[0.01385, 0.02385]	[0.01852, 0.02852]

Table 1: Confidence interval computation obtained via ProbReach, with solver precision δ=10^{-3} and interval size equal to 10^{-2}, **Type** - extremum type and $\boldsymbol{P}$ - true probability value, where single point values were analytically computed and interval values are numerically guaranteed enclosures (computed by ProbReach).

Model	Type	CI_B	CI_{CLT}	CI_{AC_W}	CI_W	CI_L	CI_{Ans}	CI_{Arc}	$Qint$
Good	max	70422	69484	69582	69496	69530	69529	77262	68456
	min	71898	71286	71339	71293	71321	71321	79369	68994
Bad	max	37388	36518	36771	36629	36687	36868	60006	36164
	max2	79306	79097	79125	79101	79118	79118	96442	77892
	min	5797	124	2766	1963	4136	4136	572	94
Deceleration	max	65248	65233	65330	65299	65320	65319	72114	59882
	min	33147	32969	33133	33018	33060	33060	34231	29096
Collision (Basic)	max	25279	24711	24834	24789	24934	24933	26045	23016
	min	5797	124	2766	1963	4136	4136	572	94
Collision (Extended)	max	191466	190776	191253	190894	191485	191472	376294	185456
	min	41153	38942	39745	39473	39537	39541	47923	37608
Collision (Advanced)	max	131517	129746	131185	129845	129934	129933	183405	127486
	min	27305	25657	25835	25736	25792	25791	29362	24569
Anesthesia	n/a	16197	15453	15834	15634	15734	15733	17845	15314

Table 2: Sample size comparison for confidence interval computation obtained via ProbReach, with solver δ precision equal to 10^{-3} and interval size equal to 10^{-2}, **Type** - extremum type; confidence level = 0.99999.

5 Conclusions

In this paper, we have provided a comprehensive evaluation of confidence interval calculation techniques for Monte Carlo (MC) and Quasi-Monte Carlo (QMC) methods. We have shown that:

- the Central Limit Theorem (CLT) interval generally performs best, in particular for small sample sizes;
- when estimating probabilities near the borders (*i.e.*, close to 0 or 1), our simple CLT modification has proved to be very effective, while other techniques cannot form intervals;
- QMC methods are more efficient than MC methods by providing precise estimates with fewer samples.

Based on our analysis, we suggest that our results can be used as guidelines for statistical model checking of time-bounded properties beyond reachability.

References

1. Agresti, A., Coull, B.A.: Approximate is better than "exact" for interval estimation of binomial proportions. The American Statistician **52**(2), 119–126 (May 1998)
2. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.H.: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In: Hybrid Systems. LNCS, vol. 736, pp. 209–229 (1992)
3. Anscombe, F.J.: The transformation of poisson, binomial and negative-binomial data. Biometrika **35**(3/4), 246–254 (December 1948)
4. Anscombe, F.J.: On estimating binomial response relations. Biometrika **43**(3/4), 461–464 (December 1956)
5. Antonov, A.A., Ermakov, S.M.: Empirically estimating error of integration by quasi-monte carlo method. Vestnik St. Petersburg University: Mathematics **47**(1), 1–8 (2015)

6. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.: Model-checking algorithms for continuous-time markov chains. IEEE Trans. Software Eng. **29**(6), 524–541 (2003)
7. Bianco, A., de Alfaro, L.: Model checking of probabalistic and nondeterministic systems. In: FSTTCS. LNCS, vol. 1026, pp. 499–513 (1995)
8. Clopper, C.J., Pearson, E.S.: The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika **26**(4), 404–413 (1934)
9. D., B.L., Tony, C.T., Anirban, D.: Interval estimation for a binomial proportion. Statistical Science **16**(2), 128–133 (2001)
10. Dean, N., Pagano, M.: Evaluating confidence interval methods for binomial proportions in clustered surveys. Journal of Survey Statistics and Methodology **3**(4), 484–503 (December 2015)
11. Edwin, W.B.: Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association **22**(158), 209–212 (1927)
12. Fränzle, M., Chen, M., Kröger, P.: In memory of Oded Maler: Automatic reachability analysis of hybrid-state automata. ACM SIGLOG News **6**(1), 19–39 (2019)
13. Gao, S., Avigad, J., Clarke, E.M.: Delta-decidability over the reals. In: LICS. pp. 305–314 (2012)
14. Gnewuch, M., Srivastav, A., Winzen, C.: Finding optimal volume subintervals with k points and calculating the star discrepancy are np-hard problems. J. Complexity **25**(2), 115–127 (2009)
15. Mahajan, K.K., Arora, S., Kaur, K.: Bayesian estimation for gini index and a poverty measure in case of pareto distribution using Jeffreys' prior. MASA **10**(1), 63–72 (2015)
16. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM (1992)
17. Pradhan, V., Banerjee, T.: Confidence interval of the difference of two independent binomial proportions using weighted profile likelihood. Communications in Statistics - Simulation and Computation **37**(4), 645–659 (2008)
18. Shmarov, F., Zuliani, P.: ProbReach: Verified probabilistic δ-reachability for stochastic hybrid systems. In: HSCC. pp. 134–139. ACM (2015)
19. Shmarov, F., Zuliani, P.: Probabilistic hybrid systems verification via SMT and Monte Carlo techniques. In: HVC. LNCS, vol. 10028, pp. 152–168 (2016)
20. Sobol', I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput. Math. and Math. Phys. **7**(4), 86 – 112 (1967)
21. Traub, J.F., Wasilkowski, G.W., Woźniakowski, H.: Information-based Complexity. Academic Press (1988)
22. Tuffin, B.: Randomization of quasi-monte carlo methods for error estimation: Survey and normal approximation. Monte Carlo Meth. and Appl. **10**(3-4), 617–628 (2004)
23. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. In: FOCS. pp. 327–338 (1985)
24. Younes, H.L.S., Simmons, R.G.: Statistical probabilistic model checking with a focus on time-bounded properties. Inf. Comput. **204**(9), 1368–1409 (2006)
25. Zuliani, P., Platzer, A., Clarke, E.M.: Bayesian statistical model checking with application to Stateflow/Simulink verification. Formal Methods in System Design **43**(2), 338–367 (2013)

A Appendix

A.1 Hybrid System Definition

A Parametric Hybrid System [19] is a tuple

$$H =< Q, \Upsilon, X, P, Y, R, \text{jump, goal} >$$

where

- $Q = \{q_0, \cdots, q_m\}$ a set of modes (discrete components of the system),
- $\Upsilon = \{(q, q') : q, q' \in Q\}$ a set of transitions between modes,
- $X = [u_1, v_1] \times \cdots \times [u_n, v_n] \subset \mathbb{R}^n$ a domain of continuous variables,
- $P = [a_1, b_1] \times \cdots \times [a_k, b_k] \subset \mathbb{R}^k$ the parameter space of the system,
- $Y = \{\mathbf{y}_q(\mathbf{p}, t) : q \in Q, \mathbf{p} \in X \times P, t \in [0, T]\}$ the continuous system dynamics where $\mathbf{y}_q : X \times P \times [0, T] \to X$,
- $R = \{\mathbf{g}_{(q,q')}(\mathbf{p}, t) : (q, q') \in \Upsilon, \mathbf{p} \in X \times P, t \in [0, T]\}$ 'reset' functions $\mathbf{g}_{(q,q')} : X \times P \times [0, T] \to X$ defining the continuous state at time $t = 0$ in mode q' after taking the transition from mode q.

and predicates (or relations)

- $\text{jump}_{(q,q')}(\mathbf{x})$ defines a discrete transition $(q, q') \in \Upsilon$ which may (but does not have to) occur upon reaching the jump condition in state $(\mathbf{x}, q) \in X \times P \times Q$,
- $\text{goal}_q(\mathbf{x})$ defines the goal state $\mathbf{x}$ in mode q.

A.2 Koksma-Hlawka inequality

The well-known Koksma-Hlawka inequality [16] provides an upper bound for the integral estimation error with QMC methods. Suppose we want to compute $I = \int_{U_d} f(x)dx$, where U_d is the hypercube over $[0, 1]^d$. Let $\{u_1, ..., u_n\}$ be a set in U_d. Then the Koksma-Hlawka inequality is:

$$\left| I - \frac{1}{n} \sum_{i=1}^{n} f(u_i) \right| \leqslant V(f) D_n^* \{u_1, ..., u_n\}, \tag{13}$$

where $V(f)$ is the bounded variation in the sense of Hardy and Krause:

$$V(f) = \sum_{k=1}^{d} \sum_{1 < i_1 < ... < i_k < d} V_{V_{it}}^k (f; i_1, ..., i_k),$$

where $V_{V_{it}}^k (f; i_1, ..., i_k)$ is the variation in sense of Vitali [22], applied to the restriction of f to the space dimension $k\{(u_1, ..., u_d) \in [0, 1]^d : u_j = 1 \text{ for } j \neq i_1, ..., i_k\}$. If $k = d$ we obtain an empty set, which can not be calculated.

The star-discrepancy D_n^* is defined as follows:

$$\mathbb{D}_n^* \{u_1, ..., u_n\} = \sup_{B \in W^*} \left| \frac{\#\{u_i : u_i \in B\}}{n} - \lambda_d(B) \right|,$$

where $\#\{u_i : u_i \in B\}$ are points from the set B and W^* is defined as the set of the form:

$$\prod_{k=1}^{d}[0, c_k) = \{y \in U_d : 0 \leqslant y_k < c_k\}$$

Unfortunately inequality (13) can not serve as a basis for a constructive evaluation of the integration error in practical applications. In particular, computing the star-discrepancy of an arbitrary set is an NP-hard problem [14]. Also, estimating the Hardy-Krause variation is a computationally heavy problem.

A.3 Qint algorithm

Consider an arbitrary probability space $(\mathcal{U}, \mathcal{B}, \mu)$, where $\mathcal{U}$ is non-empty set, $\mathcal{B}$ is σ-algebra for subsets of $\mathcal{U}$ with probability measure μ. We choose some number s so that $N = 2^s$, *i.e.*we need to split $\mathcal{U}$ on N disjunctive parts of equal measure $\mathcal{U}_1, \mathcal{U}_2 ... \mathcal{U}_\mathcal{N}$, which fully cover $\mathcal{U}$. Then we need to construct a system of N Haar functions derived from $\mathcal{U}_1, \mathcal{U}_2 ... \mathcal{U}_\mathcal{N}$ and orthonormal in $L^2(d\mu)$.

To construct an estimate of the integral I they use a set of random quadrature formulas introduced by the Ermakov-Granovsky theorem, which allows us to construct N-point formulas with two important properties: the unbiasedness property for integral I and the accuracy property for the considered Haar system. The nodes of the formula are random variables with distribution density:

$$\phi(u_1, u_2, ..., u_N) = \begin{cases} \frac{N^N}{N!} & \text{if } (u_1, u_2, ..., u_N) \in Lat(i_1, i_2, ..., i_N) \\ 0 & \text{otherwise} \end{cases}$$

where $Lat(i_1, i_2, ..., i_N)$ is a Latin set of the permutation $(i_1, i_2, ..., i_N)$, defined by the condition: $(u_1, u_2, ..., u_N) \in Lat(i_1, i_2, ..., i_N) \Leftrightarrow \forall j \in \{1, 2, ..., N\} u_j \in \mathcal{U}_{i_j}$, where $\mathcal{U}_{i_j}$ is a set of permuted orthonormal Haar functions [5].

The variance of the constructed cubature formula $Cub[f] = \frac{1}{N}\sum_{i=1}^{N} f(u_i)$ can be calculated as:

$$\mathbb{D}Cub[f] = \int_{\mathcal{U}^\mathcal{N}} Cub[f]^2 d\phi - \left(\int_{\mathcal{U}^\mathcal{N}} Cub[f] d\phi \right)^2 =$$

$$= \mathbb{D}MC[f] + \frac{1}{N}(a_1 + a_2 + ... + a_N)^2 - a_1^2 - a_2^2 - ... - a_N^2 = \mathbb{D}_{MC}[f] - \frac{1}{N}\sum_{i<j}(a_i - a_j)^2,$$

where $\mathbb{D}MC$ is the variance of MC method (2) and $a_i = \int_{\mathcal{U}_i} f(u)\mu(du)$ for $i = 1, 2, ..., N$. We can then redefine the integral estimation variance as:

$$Var(QMC) = Var(MC) - \frac{1}{N}\sum_{i<j}(a_i - a_j)^2 \ . \tag{14}$$

A.4 Further Results

Model	**Type**	c	CI_B	CI_{CLT}	CI_{AC_W}	CI_W	CI_L	CI_{Ans}	CI_{Arc}	$Qint$
Good	max	0.999	39211	39187	39215	39196	39210	39200	43407	38094
	min	0.999	39650	39364	39401	39368	39373	39373	43848	38204
Bad	max	0.999	20717	20401	20550	20497	20527	20562	32006	20322
	max2	0.999	44557	43848	43863	43848	43855	43855	56442	42888
	min	0.999	3950	107	1549	1103	1362	1362	434	n/a
Deceleration	max	0.999	36609	36039	36061	36044	36132	36130	39524	33068
	min	0.999	18727	18629	18709	18671	18628	18682	19438	16618
Collision (Basic)	max	0.999	13795	13222	13341	13286	13311	13397	15385	13098
	min	0.999	3950	107	1549	1103	1362	1362	434	n/a
Collision (Extended)	max	0.999	106252	106099	106243	106147	106224	106224	166345	104531
	min	0.999	22887	21860	22196	21935	22041	22038	24742	20862
Collision (Advanced)	max	0.999	71746	70435	70646	70636	70642	70640	143390	69642
	min	0.999	15833	15679	15746	15723	15748	15746	18354	15086
Anesthesia	n/a	0.999	9017	8516	8827	8628	8593	8592	9284	8430
Good	max	0.9999	55187	54327	54361	54347	54355	54362	60104	52990
	min	0.9999	55885	55281	55231	55286	55307	55307	61631	53411
Bad	max	0.9999	29147	28240	28339	28276	28289	28289	42463	27944
	max2	0.9999	62735	61139	61364	61152	61359	61358	86442	59012
	min	0.9999	4849	116	2153	1530	2458	2458	508	n/a
Deceleration	max	0.9999	50476	50243	50277	50250	50269	50268	55495	46084
	min	0.9999	25741	25695	25817	25779	25794	25794	26790	22466
Collision (Basic)	max	0.9999	19476	18907	19084	18984	19035	19032	21537	18128
	min	0.9999	4849	116	2153	1530	2458	2458	508	n/a
Collision (Extended)	max	0.9999	148388	147675	147834	147746	147786	147635	236423	145974
	min	0.9999	31528	29894	30420	30023	30423	30420	34736	28588
Collision (Advanced)	max	0.9999	100592	100143	100275	100174	100196	100195	168345	99456
	min	0.9999	20497	20130	20412	20312	20384	20383	23864	19788
Anesthesia	n/a	0.9999	13131	11462	11683	11658	11724	11722	13948	11288

Table 3: Samples size comparison for confidence interval computation obtained via ProbReach, with solver δ precision equal to 10^{-3} and interval size equal to 10^{-2}, **Type** - extremum type and $\boldsymbol{c}$ - confidence level.