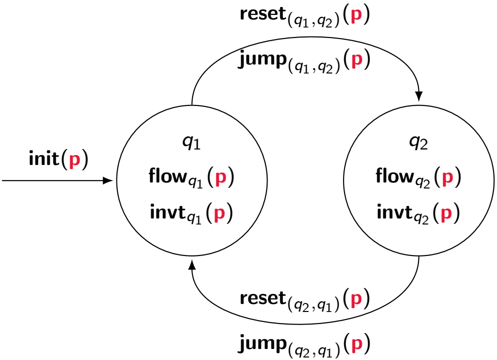
An Evaluation of Estimation Techniques for Probabilistic Verification

Mariia Vasileva and Paolo Zuliani

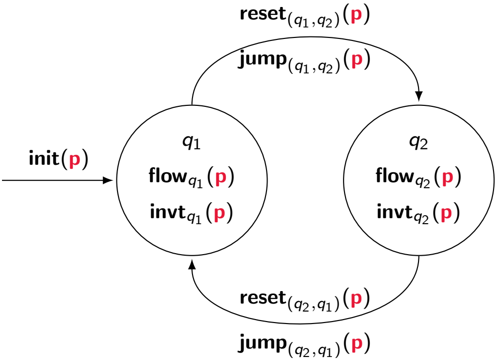
School of Computing
Newcastle University

Newcastle upon Tyne, UK

Hybrid Systems

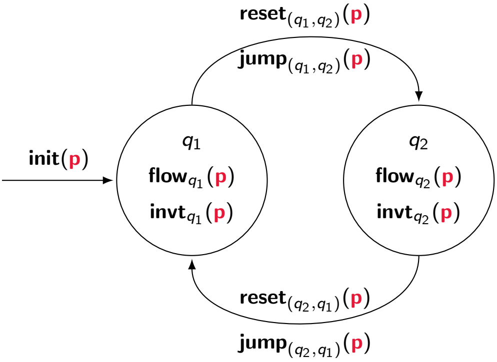


Hybrid Systems



Parametric Hybrid System (**PHS**)

- $\mathbf{p} \in P$ – parameter
- $P \neq \emptyset$ – parameter space
- $\frac{d\mathbf{p}}{dt} = 0$



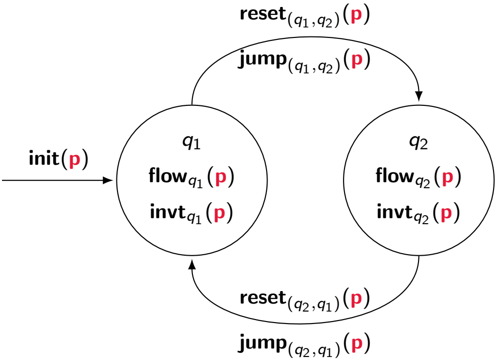
Parametric Hybrid System (**PHS**)

- $\mathbf{p} \in P$ – parameter
- $P \neq \emptyset$ – parameter space
- $\frac{d\mathbf{p}}{dt} = 0$

Stochastic PHS (**SPHS**)

- PHS with random parameters

Hybrid Systems



Parametric Hybrid System (**PHS**)

- $\mathbf{p} \in P$ – parameter
- $P \neq \emptyset$ – parameter space
- $\frac{d\mathbf{p}}{dt} = 0$

Stochastic PHS (**SPHS**)

- PHS with random parameters

- **init** and **reset** – numerically "type 2 computable functions"
- **flow** – Lipschitz-continuous ODEs
- **invt** and **jump** – Boolean logic formula $\bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{k_i} (f_{i,j}(\mathbf{x}, \mathbf{p}) \circ 0) \right)$
  - $\circ \in \{>, \geq\}$
  - $f_{i,j}$ – numerically "type 2 computable functions"

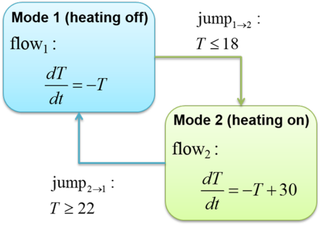
Bounded Probabilistic Reachability: Thermostat

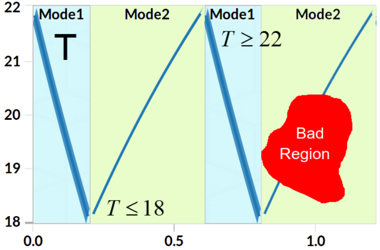
- *Bounded k-step reachability* in SPHSs aims to find the probability that for the given initial conditions, the system reaches a bad state in k discrete transitions.

Bounded Probabilistic Reachability: Thermostat

- *Bounded k-step reachability* in SPHSs aims to find the probability that for the given initial conditions, the system reaches a bad state in k discrete transitions.

What is the probability of reaching the **bad** region?

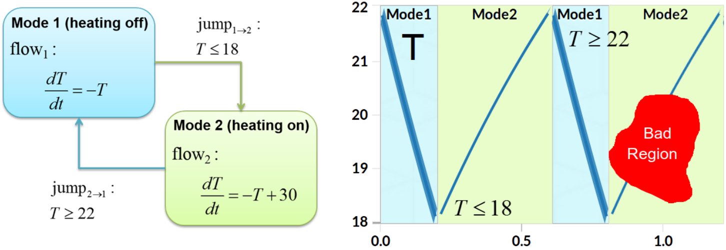




Bounded Probabilistic Reachability: Thermostat

- *Bounded k-step reachability* in SPHSs aims to find the probability that for the given initial conditions, the system reaches a bad state in k discrete transitions.

What is the probability of reaching the **bad** region?



- The probability can be computed as an *integral* of the form $\int_G d\mathbb{P}$
  - G denotes the set of all random parameter values for which the system with random parameters reaches a bad state in k steps.
  - $\mathbb{P}$ is the probability measure associated with the random parameters.

Integral Estimation Methods

- *Formal Approach* (Exhaustive Search)
  - Absolute numerical guarantees.
  - Performance crucially depends on the number of parameters.

Integral Estimation Methods

- *Formal Approach* (Exhaustive Search)
  - Absolute numerical guarantees.
  - Performance crucially depends on the number of parameters.
- *Sampling Approach* (Quasi-Monte Carlo (QMC)/ Monte Carlo methods (MC))
  - Statistical numerical guarantees (*still works OK*).
  - *Scales better* with the number of parameters.
  - We have implemented *The QMC approach in ProbReach* - a tool for probabilistic bounded reachability analysis in SPHS (Shmarov & Zuliani, HSCC 2015).

Integral Estimation Methods: QMC vs Randomised QMC

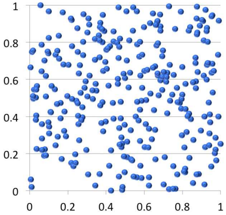
- QMC methods select the points *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967):
  - A QMC advantage with respect to MC is that its error is $O\left(\frac{1}{N}\right)$, while the MC error is $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size.
  - The terms of quasi-random sequences are *statistically dependent*, so the Central Limit Theorem (CLT) *can not* be directly used.

Integral Estimation Methods: QMC vs Randomised QMC

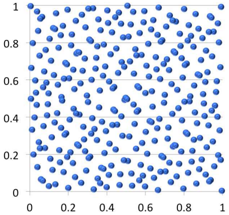
- QMC methods select the points *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967):
  - A QMC advantage with respect to MC is that its error is $O\left(\frac{1}{N}\right)$, while the MC error is $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size.
  - The terms of quasi-random sequences are *statistically dependent*, so the Central Limit Theorem (CLT) *can not* be directly used.
- We can successfully use *Randomised QMC* (RQMC) methods:
  - Suppose $\mathfrak{X} = \{x_1, ..., x_n\}$ - a low-discrepancy set: by transformation $\tilde{\mathfrak{X}} = \Gamma(\mathfrak{X}, \xi)$ a finite set $\tilde{\mathfrak{X}}$ is generated by the random variable ξ.

Integral Estimation Methods: QMC vs Randomised QMC

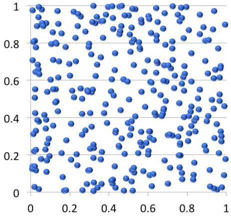
- QMC methods select the points *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967):
  - A QMC advantage with respect to MC is that its error is $O\left(\frac{1}{N}\right)$, while the MC error is $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size.
  - The terms of quasi-random sequences are *statistically dependent*, so the Central Limit Theorem (CLT) *can not* be directly used.
- We can successfully use *Randomised QMC* (RQMC) methods:
  - Suppose $\mathfrak{X} = \{x_1, ..., x_n\}$ - a low-discrepancy set: by transformation $\tilde{\mathfrak{X}} = \Gamma(\mathfrak{X}, \xi)$ a finite set $\tilde{\mathfrak{X}}$ is generated by the random variable ξ.



Pseudorandom points



Sobol sequence points



Randomised Sobol points

Intervals Based on the Beta-Function

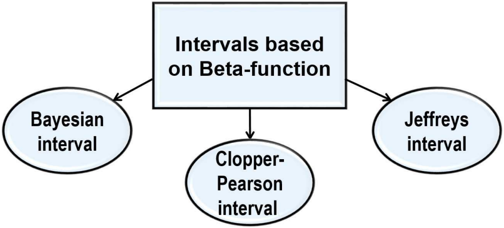


Figure 1: Alternative intervals based on the Beta-function

Intervals Based on the Beta-Function

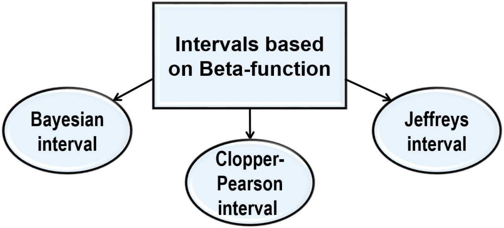


Figure 1: Alternative intervals based on the Beta-function

- These intervals compute the *posterior distribution* of the unknown quantity by using its *prior distribution.*
- The standard PDF of beta distribution is represented by the formula $Beta(\alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$, where $0 \leq x \leq 1$; $\alpha, \beta > 0$ and $B(\alpha, \beta)$ is the Beta-Function defined as $B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$.
- If a parameter value p has a prior distribution $Beta(\alpha, \beta)$ then after n Bernoulli trials with n_s successes, p has posterior distribution $Beta(n_s + \alpha, n - n_s + \beta)$.

Intervals Based on the Standard CLT Interval

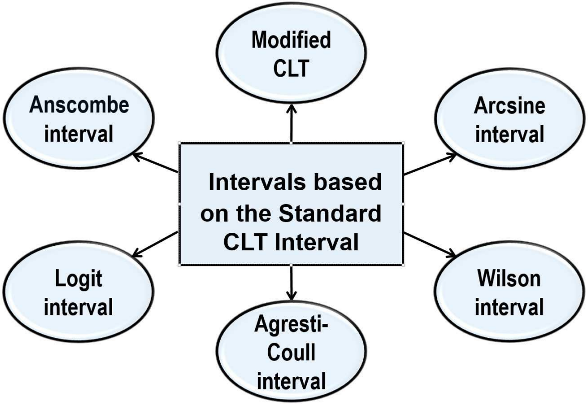


Figure 2: Intervals based on the Standard CLT interval

Intervals Based on the Standard CLT Interval

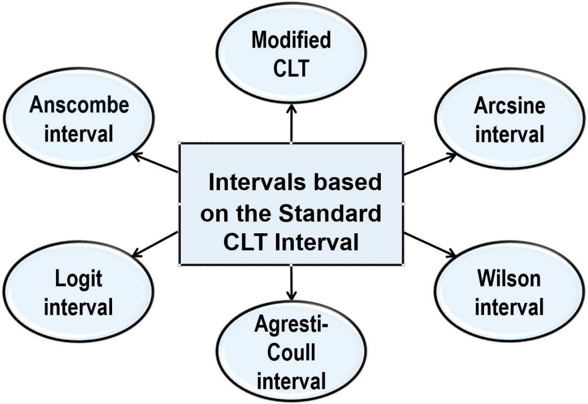


Figure 2: Intervals based on the Standard CLT interval

- We also consider *modified Qint method* presented by Ermakov Antonov, which is based on the random quadrature formulas.

Modified CLT interval

Standard CLT Confidence Interval (CI)

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

- N is the number of samples
- $\tilde{X} = \frac{1}{N} \sum_{i=1}^{N} x_i$
- parameter a defines confidence level at $c = 1 - a$.
- σ - standard deviation of the samples $x_1, ...x_N$
- $C_a = Quant(1 - \frac{a}{2})$ is the inverse CDF of a Gaussian distribution with parameters (0,1).

Modified CLT interval

Standard CLT Confidence Interval (CI)

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

- N is the number of samples
- $\tilde{X} = \frac{1}{N} \sum_{i=1}^{N} x_i$
- parameter a defines confidence level at $c = 1 - a$.
- σ - standard deviation of the samples $x_1, ...x_N$
- $C_a = Quant(1 - \frac{a}{2})$ is the inverse CDF of a Gaussian distribution with parameters (0,1).

- Unfortunately, in practice *the variance σ^2 is unknown.*

Modified CLT interval

Standard CLT Confidence Interval (CI)

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

- N is the number of samples
- $\tilde{X} = \frac{1}{N} \sum_{i=1}^{N} x_i$
- parameter a defines confidence level at $c = 1 - a$.
- σ - standard deviation of the samples $x_1, ...x_N$
- $C_a = Quant(1 - \frac{a}{2})$ is the inverse CDF of a Gaussian distribution with parameters (0,1).

- Unfortunately, in practice *the variance σ^2 is unknown.*

Modified CLT Interval

We can use CI_{CLT} by replacing σ to *sample standard deviation $s = \frac{1}{N^2}$ at the initial stages of the computation if $\tilde{X}$ is equal to 0 (or 1).*

Results: MC and QMC Error Comparison

Absolute error with respect to the number of samples.

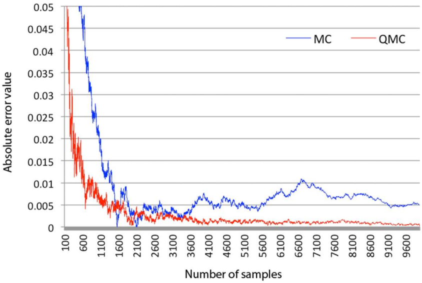


Figure 3: Model: Collision, type-max.

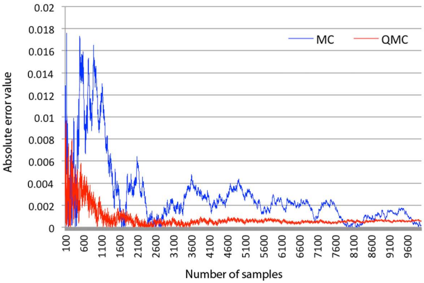


Figure 4: Model: Bad, type-max2.

Results: MC and QMC Error Comparison

Absolute error with respect to the number of samples.

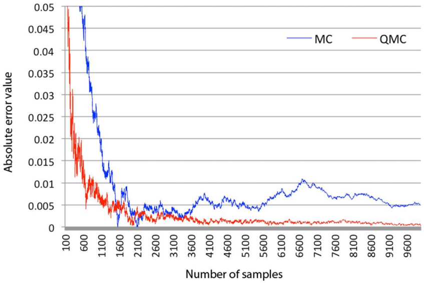


Figure 3: Model: Collision, type-max.

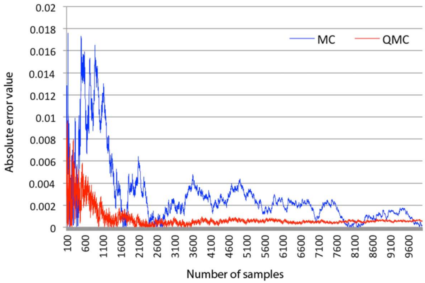


Figure 4: Model: Bad, type-max2.

- QMC advantage in the error size holds for all tested models
- We also note “unlucky” pairs (p,n) such that the corresponding MC error is much bigger than the QMC method error.

Results: Border Probability Interval Coverage

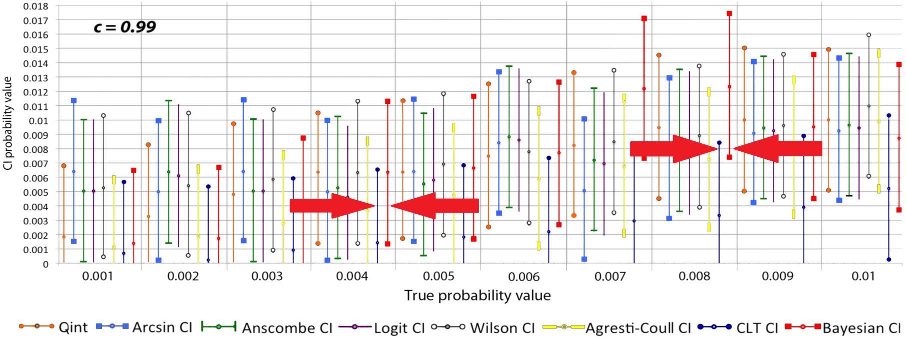


Figure 5: Comparison of confidence interval distribution for probability values near 0, interval size equal to 10^{-2} and **c** - confidence level.

Results: Border Probability Interval Coverage

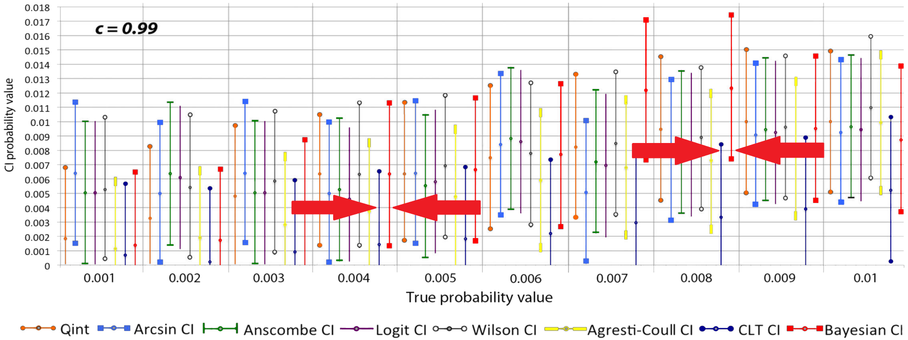


Figure 5: Comparison of confidence interval distribution for probability values near 0, interval size equal to 10^{-2} and **c** - confidence level.

- *The Bayesian method tends to overestimate* the true probability values while *CLT tends to underestimate* them.

Results: Border Probability Interval Coverage

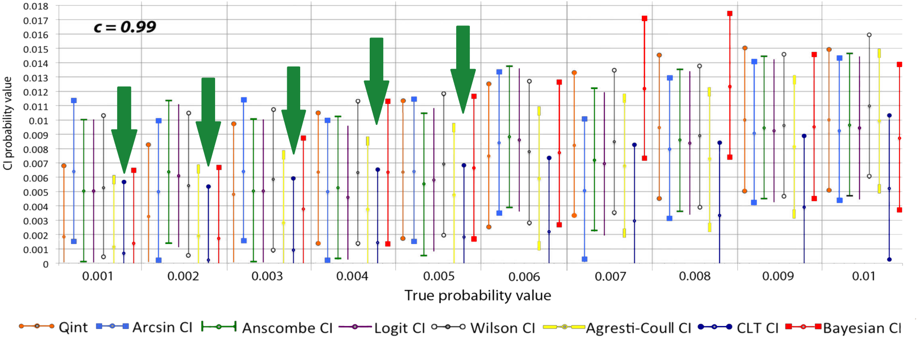


Figure 6: Comparison of confidence interval distribution for probability values near 0, interval size equal to 10^{-2} and **c** - confidence level.

Results: Border Probability Interval Coverage

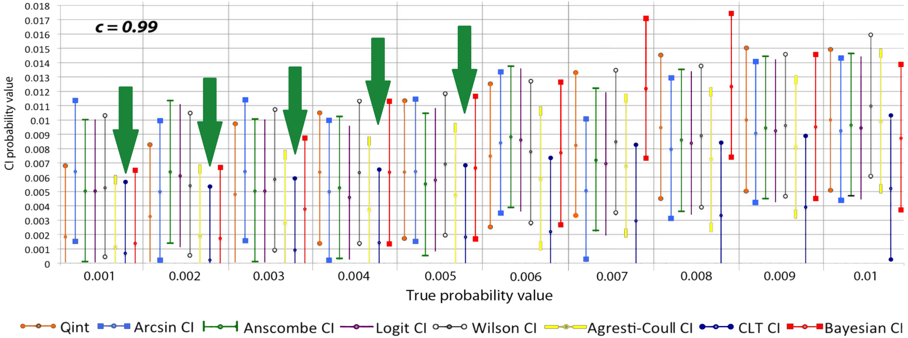


Figure 6: Comparison of confidence interval distribution for probability values near 0, interval size equal to 10^{-2} and **c** - confidence level.

- The *size* of the Bayesian, CLT and Agresti-Coull intervals *decreases* significantly as the true probability moves toward 0.

Results: Border Probability Sample Sizes

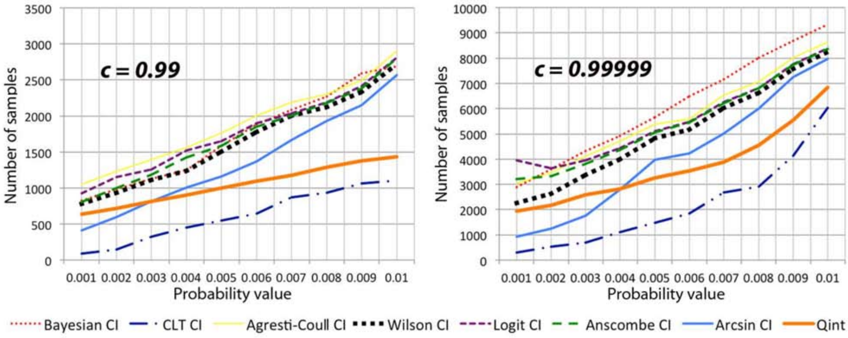


Figure 7: Comparison of sample size for probability values near 0, interval size equal to 10^{-2} and **c** - confidence level.

Results: Border Probability Sample Sizes

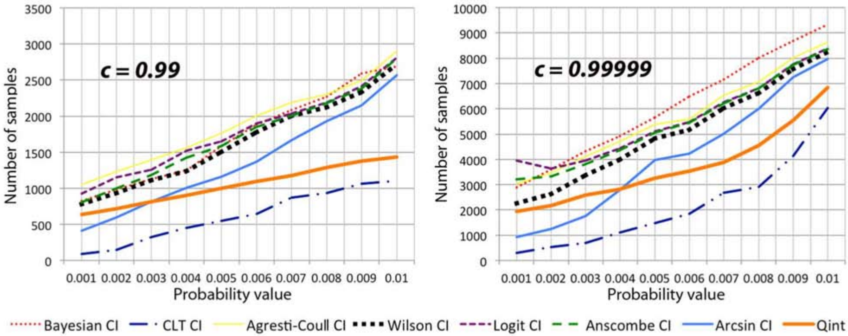


Figure 7: Comparison of sample size for probability values near 0, interval size equal to 10^{-2} and **c** - confidence level.

- Our modified CLT technique shows the best result across all confidence levels.
- When increasing the confidence all CIs based on the standard interval outperform the Bayesian CI.

Results: Tested Models Interval Coverage

Model	Type	P	CI_B	CI_{CLT}	CI_{AC_W}	CI_W
Good	max	0.1	[0.09499, 0.10499]	[0.09378, 0.10378]	[0.09386, 0.10386]	[0.09389, 0.10389]
	min	0.1	[0.09419, 0.10419]	[0.09667, 0.10667]	[0.09668, 0.10668]	[0.09677, 0.10677]
Bad	max	0.95001	[0.94525, 0.95525]	[0.94579, 0.95579]	[0.94564, 0.95564]	[0.94548, 0.95548]
	max2	0.88747	[0.88215, 0.89215]	[0.88055, 0.89055]	[0.88057, 0.89057]	[0.88046, 0.89046]
	min	4×10^{-7}	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Deceleration	max	[0.08404, 0.08881]	[0.08613, 0.09613]	[0.08624, 0.09624]	[0.08312, 0.09312]	[0.08725, 0.09725]
	min	[0.04085, 0.04275]	[0.03514, 0.04514]	[0.03919, 0.04919]	[0.03918, 0.04918]	[0.03942, 0.04942]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96359, 0.97359]	[0.96241, 0.97241]	[0.96767, 0.9767]	[0.96892, 0.96892]
	min	[0 , 0.00201]	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Collision (Extended)	max	[0.35751, 0.49961]	[0.42651, 0.43652]	[0.42719, 0.43724]	[0.42757, 0.43757]	[0.42656, 0.43656]
	min	[0.04296, 0.06311]	[0.04979, 0.05979]	[0.04766, 0.05766]	[0.04764, 0.05764]	[0.04748, 0.05748]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.20515, 0.21519]	[0.20558, 0.21563]	[0.20533, 0.21533]	[0.20531, 0.21531]
	min	[0.02471, 0.05191]	[0.03011, 0.04015]	[0.02902, 0.03902]	[0.02954, 0.03945]	[0.03956, 0.04956]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01284, 0.02284]	[0.01513, 0.02511]	[0.01623, 0.02623]	[0.01545, 0.02545]

Table 1: Confidence interval computation obtained via ProbReach, with solver precision $\delta = 10^{-3}$ and interval size equal to 10^{-2}, **Type** - extremum type and **P** - true probability value, where single point values were analytically computed and interval values are numerically guaranteed enclosures (computed by ProbReach); confidence level $= 0.99999$.

Results: Tested Models Interval Coverage

Model	Type	P	CI_B	CI_{CLT}	CI_{AC_W}	CI_W
Good	max	0.1	[0.09499, 0.10499]	[0.09378, 0.10378]	[0.09386, 0.10386]	[0.09389, 0.10389]
	min	0.1	[0.09419, 0.10419]	[0.09667, 0.10667]	[0.09668, 0.10668]	[0.09677, 0.10677]
Bad	max	0.95001	[0.94525, 0.95525]	[0.94579, 0.95579]	[0.94564, 0.95564]	[0.94548, 0.95548]
	max2	0.88747	[0.88215, 0.89215]	[0.88055, 0.89055]	[0.88057, 0.89057]	[0.88046, 0.89046]
	min	4×10^{-7}	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Deceleration	max	[0.08404, 0.08881]	[0.08613, 0.09613]	[0.08624, 0.09624]	[0.08312, 0.09312]	[0.08725, 0.09725]
	min	[0.04085, 0.04275]	[0.03514, 0.04514]	[0.03919, 0.04919]	[0.03918, 0.04918]	[0.03942, 0.04942]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96359, 0.97359]	[0.96241, 0.97241]	[0.96767, 0.9767]	[0.96892, 0.96892]
	min	[0 , 0.00201]	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Collision (Extended)	max	[0.35751, 0.49961]	[0.42651, 0.43652]	[0.42719, 0.43724]	[0.42757, 0.43757]	[0.42656, 0.43656]
	min	[0.04296, 0.06311]	[0.04979, 0.05979]	[0.04766, 0.05766]	[0.04764, 0.05764]	[0.04748, 0.05748]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.20515, 0.21519]	[0.20558, 0.21563]	[0.20533, 0.21533]	[0.20531, 0.21531]
	min	[0.02471, 0.05191]	[0.03011, 0.04015]	[0.02902, 0.03902]	[0.02954, 0.03945]	[0.03956, 0.04956]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01284, 0.02284]	[0.01513, 0.02511]	[0.01623, 0.02623]	[0.01545, 0.02545]

Table 1: Confidence interval computation obtained via ProbReach, with solver precision $\delta=10^{-3}$ and interval size equal to 10^{-2}, **Type** - extremum type and **P** - true probability value, where single point values were analytically computed and interval values are numerically guaranteed enclosures (computed by ProbReach); confidence level = 0.99999.

- *The true probability value* of the “Bad” model Type min and the Collision (Basic) model Type min is *very close to 0*.
- It allows the *Bayesian, CLT and Agresti-Coull methods* to form intervals, which *represent a half of the proposed interval size* 10^{-2}.

Results: Tested Models Interval Coverage

Model	Type	P	CI_B	CI_{CLT}	CI_{AC_W}	CI_W
Good	max	0.1	[0.09499, 0.10499]	[0.09378, 0.10378]	[0.09386, 0.10386]	[0.09389, 0.10389]
	min	0.1	[0.09419, 0.10419]	[0.09667, 0.10667]	[0.09668, 0.10668]	[0.09677, 0.10677]
Bad	max	0.95001	[0.94525, 0.95525]	[0.94579, 0.95579]	[0.94564, 0.95564]	[0.94548, 0.95548]
	max2	0.88747	[0.88215, 0.89215]	[0.88055, 0.89055]	[0.88057, 0.89057]	[0.88046, 0.89046]
	min	4×10^{-7}	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Deceleration	max	[0.08404, 0.08881]	[0.08613, 0.09613]	[0.08624, 0.09624]	[0.08312, 0.09312]	[0.08725, 0.09725]
	min	[0.04085, 0.04275]	[0.03514, 0.04514]	[0.03919, 0.04919]	[0.03918, 0.04918]	[0.03942, 0.04942]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96359, 0.97359]	[0.96241, 0.97241]	[0.96767, 0.9767]	[0.96892, 0.96892]
	min	[0 , 0.00201]	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Collision (Extended)	max	[0.35751, 0.49961]	[0.42651, 0.43652]	[0.42719, 0.43724]	[0.42757, 0.43757]	[0.42656, 0.43656]
	min	[0.04296, 0.06311]	[0.04979, 0.05979]	[0.04766, 0.05766]	[0.04764, 0.05764]	[0.04748, 0.05748]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.20515, 0.21519]	[0.20558, 0.21563]	[0.20533, 0.21533]	[0.20531, 0.21531]
	min	[0.02471, 0.05191]	[0.03011, 0.04015]	[0.02902, 0.03902]	[0.02954, 0.03945]	[0.03956, 0.04956]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01284, 0.02284]	[0.01513, 0.02511]	[0.01623, 0.02623]	[0.01545, 0.02545]

Table 2: Confidence interval computation obtained via ProbReach, with solver precision $\delta=10^{-3}$ and interval size equal to 10^{-2}, **Type** - extremum type and **P** - true probability value, where single point values were analytically computed and interval values are numerically guaranteed enclosures (computed by ProbReach); confidence level = 0.99999.

Results: Tested Models Interval Coverage

Model	Type	P	CI_B	CI_{CLT}	CI_{AC_W}	CI_W
Good	max	0.1	[0.09499, 0.10499]	[0.09378, 0.10378]	[0.09386, 0.10386]	[0.09389, 0.10389]
	min	0.1	[0.09419, 0.10419]	[0.09667, 0.10667]	[0.09668, 0.10668]	[0.09677, 0.10677]
Bad	max	0.95001	[0.94525, 0.95525]	[0.94579, 0.95579]	[0.94564, 0.95564]	[0.94548, 0.95548]
	max2	0.88747	[0.88215, 0.89215]	[0.88055, 0.89055]	[0.88057, 0.89057]	[0.88046, 0.89046]
	min	4×10^{-7}	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Deceleration	max	[0.08404, 0.08881]	[0.08613, 0.09613]	[0.08624, 0.09624]	[0.08312, 0.09312]	[0.08725, 0.09725]
	min	[0.04085, 0.04275]	[0.03514, 0.04514]	[0.03919, 0.04919]	[0.03918, 0.04918]	[0.03942, 0.04942]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96359, 0.97359]	[0.96241, 0.97241]	[0.96767, 0.9767]	[0.96892, 0.96892]
	min	[0 , 0.00201]	[0, 0.00517]	[0, 0.00319]	[0, 0.00494]	[0, 0.00984]
Collision (Extended)	max	[0.35751, 0.49961]	[0.42651, 0.43652]	[0.42719, 0.43724]	[0.42757, 0.43757]	[0.42656, 0.43656]
	min	[0.04296, 0.06311]	[0.04979, 0.05979]	[0.04766, 0.05766]	[0.04764, 0.05764]	[0.04748, 0.05748]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.20515, 0.21519]	[0.20558, 0.21563]	[0.20533, 0.21533]	[0.20531, 0.21531]
	min	[0.02471, 0.05191]	[0.03011, 0.04015]	[0.02902, 0.03902]	[0.02954, 0.03945]	[0.03956, 0.04956]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01284, 0.02284]	[0.01513, 0.02511]	[0.01623, 0.02623]	[0.01545, 0.02545]

Table 2: Confidence interval computation obtained via ProbReach, with solver precision $\delta=10^{-3}$ and interval size equal to 10^{-2}, **Type** - extremum type and **P** - true probability value, where single point values were analytically computed and interval values are numerically guaranteed enclosures (computed by ProbReach); confidence level = 0.99999.

- *The true probability intervals of the Collision Extended, Collision Advanced, and Anesthesia models contain all confidence intervals*.

Results: Tested Models Interval Coverage

Model	Type	P	CI_L	CI_{Ans}	CI_{Arc}	$Qint$
Good	max	0.1	[0.09391, 0.10391]	[0.09392, 0.10392]	[0.09405, 0.10405]	[0.09512, 0.10512]
	min	0.1	[0.09671, 0.10671]	[0.09679, 0.10679]	[0.09675, 0.10675]	[0.09525, 0.10525]
Bad	max	0.95001	[0.94545, 0.95545]	[0.94543, 0.95543]	[0.94735, 0.95735]	[0.94543, 0.95543]
	max2	0.88747	[0.88046, 0.89046]	[0.88046, 0.89046]	[0.88325, 0.89325]	[0.88052, 0.89052]
	min	4×10^{-7}	[0, 0.00992]	[0, 0.00992]	[0.00445, 0.0139]	[0,0.005]
Deceleration	max	[0.08404, 0.08881]	[0.08725, 0.09725]	[0.08726, 0.09726]	[0.08746, 0.09746]	[0.08737, 0.09735]
	min	[0.04085, 0.04275]	[0.03943, 0.04943]	[0.03944, 0.04944]	[0.039, 0.049]	[0.03377, 0.04377]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96689, 0.97589]	[0.96683, 0.97583]	[0.96863, 0.97863]	[0.96462, 0.97462]
	min	[0 , 0.00201]	[0, 0.00992]	[0, 0.00992]	[0.00445, 0.0139]	[0,0.005]
Collision (Extended)	max	[0.35751, 0.49961]	[0.41774, 0.42774]	[0.41779, 0.42779]	[0.42745, 0.43745]	[0.42875, 0.43875]
	min	[0.04296, 0.06311]	[0.04745, 0.05745]	[0.04776, 0.05776]	[0.05776, 0.05673]	[0.04576, 0.05576]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.20547, 0.21547]	[0.20547, 0.21547]	[0.20385, 0.21385]	[0.20453, 0.21453]
	min	[0.02471, 0.05191]	[0.03861, 0.04861]	[0.03887, 0.04887]	[0.0363, 0.04363]	[0.03031, 0.04031]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01557, 0.02557]	[0.01562, 0.02562]	[0.01385, 0.02385]	[0.01852, 0.02852]

Table 3: Confidence interval computation obtained via ProbReach, with solver precision $\delta=10^{-3}$ and interval size equal to 10^{-2}, **Type** - extremum type and **P** - true probability value, where single point values were analytically computed and interval values are numerically guaranteed enclosures (computed by ProbReach); confidence level = 0.99999.

Results: Tested Models Interval Coverage

Model	Type	P	CI_L	CI_{Ans}	CI_{Arc}	$Qint$
Good	max	0.1	[0.09391, 0.10391]	[0.09392, 0.10392]	[0.09405, 0.10405]	[0.09512, 0.10512]
	min	0.1	[0.09671, 0.10671]	[0.09679, 0.10679]	[0.09675, 0.10675]	[0.09525, 0.10525]
Bad	max	0.95001	[0.94545, 0.95545]	[0.94543, 0.95543]	[0.94735, 0.95735]	[0.94543, 0.95543]
	max2	0.88747	[0.88046, 0.89046]	[0.88046, 0.89046]	[0.88325, 0.89325]	[0.88052, 0.89052]
	min	4×10^{-7}	[0, 0.00992]	[0, 0.00992]	[0.00445, 0.0139]	[0,0.005]
Deceleration	max	[0.08404, 0.08881]	[0.08725, 0.09725]	[0.08726, 0.09726]	[0.08746, 0.09746]	[0.08737, 0.09735]
	min	[0.04085, 0.04275]	[0.03943, 0.04943]	[0.03944, 0.04944]	[0.039, 0.049]	[0.03377, 0.04377]
Collision (Basic)	max	[0.96567, 0.97254]	[0.96689, 0.97589]	[0.96683, 0.97583]	[0.96863, 0.97863]	[0.96462, 0.97462]
	min	[0 , 0.00201]	[0, 0.00992]	[0, 0.00992]	[0.00445, 0.0139]	[0,0.005]
Collision (Extended)	max	[0.35751, 0.49961]	[0.41774, 0.42774]	[0.41779, 0.42779]	[0.42745, 0.43745]	[0.42875, 0.43875]
	min	[0.04296, 0.06311]	[0.04745, 0.05745]	[0.04776, 0.05776]	[0.05776, 0.05673]	[0.04576, 0.05576]
Collision (Advanced)	max	[0.14807, 0.31121]	[0.20547, 0.21547]	[0.20547, 0.21547]	[0.20385, 0.21385]	[0.20453, 0.21453]
	min	[0.02471, 0.05191]	[0.03861, 0.04861]	[0.03887, 0.04887]	[0.0363, 0.04363]	[0.03031, 0.04031]
Anesthesia	n/a	[0.00916, 0.04222]	[0.01557, 0.02557]	[0.01562, 0.02562]	[0.01385, 0.02385]	[0.01852, 0.02852]

Table 3: Confidence interval computation obtained via ProbReach, with solver precision $\delta = 10^{-3}$ and interval size equal to 10^{-2}, **Type** - extremum type and **P** - true probability value, where single point values were analytically computed and interval values are numerically guaranteed enclosures (computed by ProbReach); confidence level = 0.99999.

- *The Arcsin interval does not contain single probability value* for "Bad" model Type min, while all other CIs contain single probability values
- All CIs overlap with the true probability intervals.

Results: Tested Models Sample Size

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 4: 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 $= 0.99999$.

Results: Tested Models Sample Size

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 4: 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 $= 0.99999$.

- *All CIs except Arcsin CI show better result in number of points with respect to Bayesian CI.*

Results: Tested Models Sample Size

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 5: 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 $= 0.99999$.

Results: Tested Models Sample Size

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 5: 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 $= 0.99999$.

- *The best results was shown by modified CLT and Qint CIs*
- Our proposed CLT modification can provide reasonable results for RQMC calculation in comparison with the Bayesian MC method.

Conclusion

We provided a comprehensive evaluation of CIs calculation techniques based on MC and QMC techniques and showed that:

Conclusion

We provided a comprehensive evaluation of CIs calculation techniques based on MC and QMC techniques and showed that:

- 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

Conclusion

We provided a comprehensive evaluation of CIs calculation techniques based on MC and QMC techniques and showed that:

- 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-based techniques have excellent convergence and efficiency* especially when the number of samples is small

Conclusion

We provided a comprehensive evaluation of CIs calculation techniques based on MC and QMC techniques and showed that:

- 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-based techniques have excellent convergence and efficiency* especially when the number of samples is small
- *QMC methods are more efficient than MC methods* by providing precise estimates with fewer samples.

References I

Anton A. Antonov and Sergei M. Ermakov, *Empirically estimating error of integration by quasi-monte carlo method*, Vestnik Journal: Mathematics **47** (2015), no. 1, 1–8.

Sicun Gao, Jeremy Avigad, and Edmund M. Clarke, *Delta-decidability over the reals*, LICS, 2012, pp. 305–314.

Fedor Shmarov and Paolo Zuliani, *ProbReach: Verified probabilistic δ-reachability for stochastic hybrid systems*, HSCC, ACM, 2015, pp. 134–139.

Appendix

Appendix: δ-satisfiability

- Given an arbitrary bounded first-order formula:

$$\phi = \exists^{I_1} x_1, ..., \exists^{I_n} x_n : \bigwedge_{i=1}^{m} (\bigvee_{j=1}^{k_i} f_{ij}(x_1, ..., x_n) = 0)$$

where each f_{ij} is a Type 2 computable function (i.e., "numerically computable").

Appendix: δ-satisfiability

- Given an arbitrary bounded first-order formula:

$$\phi = \exists^{I_1} x_1, ..., \exists^{I_n} x_n : \bigwedge_{i=1}^{m} (\bigvee_{j=1}^{k_i} f_{ij}(x_1, ..., x_n) = 0)$$

  where each f_{ij} is a Type 2 computable function (i.e., "numerically computable").
- For $\delta \in \mathbb{Q}^+$, define the δ-**weakening** of ϕ:

$$\phi^\delta = \exists^{I_1} x_1, ..., \exists^{I_n} x_n : \bigwedge_{i=1}^{m} (\bigvee_{j=1}^{k_i} |f_{ij}(x_1, ..., x_n)| \leq \delta)$$

Appendix: δ-satisfiability

- Given an arbitrary bounded first-order formula:

$$\phi = \exists^{I_1} x_1, ..., \exists^{I_n} x_n : \bigwedge_{i=1}^{m} (\bigvee_{j=1}^{k_i} f_{ij}(x_1, ..., x_n) = 0)$$

  where each f_{ij} is a Type 2 computable function (i.e., "numerically computable").
- For $\delta \in \mathbb{Q}^+$, define the δ-**weakening** of ϕ:

$$\phi^\delta = \exists^{I_1} x_1, ..., \exists^{I_n} x_n : \bigwedge_{i=1}^{m} (\bigvee_{j=1}^{k_i} |f_{ij}(x_1, ..., x_n)| \leq \delta)$$

- Given ϕ and $\delta \in \mathbb{Q}^+$, a δ-complete decision procedure (Gao *et al.*, LICS 2012) correctly returns one of the following:
  - δ-**sat** - if ϕ^δ is true (but ϕ might not be),
  - **unsat** - if ϕ is false (can be trusted).
- δ-**sat** answer does not imply satisfiability of the (original) formula.

Appendix: Monte Carlo

- Compute MC integral estimation:

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

where N points u_i are uniformly distributed on $[a, b]$

Appendix: Monte Carlo

- Compute MC integral estimation:

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

  where N points u_i are uniformly distributed on $[a,b]$

- The variance of the MC estimator is:

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

Appendix: Monte Carlo

- Compute MC integral estimation:

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

  where N points u_i are uniformly distributed on $[a, b]$

- The variance of the MC estimator is:

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

- The MC integration error mean is $\frac{\sigma_f}{\sqrt{N}}$.

Appendix: Monte Carlo

- Compute MC integral estimation:

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

where N points u_i are uniformly distributed on $[a, b]$

- The variance of the MC estimator is:

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

- The MC integration error mean is $\frac{\sigma_f}{\sqrt{N}}$.
- In practice, the integrand variance σ_f^2 is often unknown. That is why the next estimation is instead used:

$$\widehat{\sigma}_f^2 \approx \frac{1}{N-1}\sum_{i=1}^{N}\left(f(u_i) - \frac{1}{N}\sum_{i=1}^{N} f(u_i)\right)^2$$

Appendix: Quasi-Monte Carlo

- QMC methods select the points u_i *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967).

Appendix: Quasi-Monte Carlo

- QMC methods select the points u_i *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967).

Remark

A QMC advantage with respect to MC is that its error is $O\left(\frac{1}{N}\right)$, while the MC error is $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size.

Appendix: Quasi-Monte Carlo

- QMC methods select the points u_i *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967).

Remark

A QMC advantage with respect to MC is that its error is $O\left(\frac{1}{N}\right)$, while the MC error is $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size.

- *The Koksma-Hlawka inequality* that aims to bound the QMC estimation error, but *is not useful in practice*

Appendix: Quasi-Monte Carlo

- QMC methods select the points u_i *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967).

Remark

A QMC advantage with respect to MC is that its error is $O\left(\frac{1}{N}\right)$, while the MC error is $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size.

- *The Koksma-Hlawka inequality* that aims to bound the QMC estimation error, but *is not useful in practice*
- The terms of quasi-random sequences are *statistically dependent*, so the Central Limit Theorem (CLT) *can not* be directly used for estimating the integration error

Appendix: Quasi-Monte Carlo

- QMC methods select the points u_i *deterministically* using low-discrepancy sequences, *e.g.* Sobol sequence (Sobol, 1967).

Remark

A QMC advantage with respect to MC is that its error is $O\left(\frac{1}{N}\right)$, while the MC error is $O\left(\frac{1}{\sqrt{N}}\right)$, where N is the sample size.

- *The Koksma-Hlawka inequality* that aims to bound the QMC estimation error, but *is not useful in practice*
- The terms of quasi-random sequences are *statistically dependent*, so the Central Limit Theorem (CLT) *can not* be directly used for estimating the integration error
- However, we can successfully use the CLT for estimating the error of *Randomised Quasi-Monte Carlo* (RQMC) methods.

Appendix: Randomised Quasi-Monte Carlo

- Suppose $\mathfrak{X} = \{x_1, ..., x_n\}$ - a deterministic low-discrepancy set

Appendix: Randomised Quasi-Monte Carlo

- Suppose $\mathfrak{X} = \{x_1, ..., x_n\}$ - a deterministic low-discrepancy set
- By transformation $\tilde{\mathfrak{X}} = \Gamma(\mathfrak{X}, \xi)$ a finite set $\tilde{\mathfrak{X}}$ is generated by the random variable ξ

Appendix: Randomised Quasi-Monte Carlo

- Suppose $\mathfrak{X} = \{x_1, ..., x_n\}$ - a deterministic low-discrepancy set
- By transformation $\tilde{\mathfrak{X}} = \Gamma(\mathfrak{X}, \xi)$ a finite set $\tilde{\mathfrak{X}}$ is generated by the random variable ξ

Appendix: Randomised Quasi-Monte Carlo

- Suppose $\mathfrak{X} = \{x_1, ..., x_n\}$ - a deterministic low-discrepancy set
- By transformation $\tilde{\mathfrak{X}} = \Gamma(\mathfrak{X}, \xi)$ a finite set $\tilde{\mathfrak{X}}$ is generated by the random variable ξ

Example

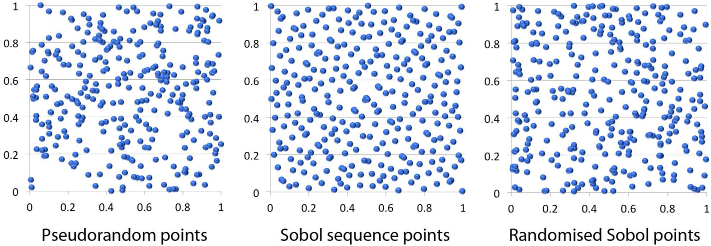
Transformation $\Gamma = (\mathfrak{X} + \xi)$ *mod* 1, where ξ is a random sample from MC sequence and $\mathfrak{X}$ is low-discrepancy sample from Sobol sequence

Appendix: Randomised Quasi-Monte Carlo

- Suppose $\mathfrak{X} = \{x_1, ..., x_n\}$ - a deterministic low-discrepancy set
- By transformation $\tilde{\mathfrak{X}} = \Gamma(\mathfrak{X}, \xi)$ a finite set $\tilde{\mathfrak{X}}$ is generated by the random variable ξ

Example

Transformation $\Gamma = (\mathfrak{X} + \xi)$ *mod* 1, where ξ is a random sample from MC sequence and $\mathfrak{X}$ is low-discrepancy sample from Sobol sequence



Appendix: Randomized Quasi-Monte Carlo

- For a randomised set $\tilde{\mathfrak{X}}_i$ we construct a RQMC estimate:

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

for $0 < j \leqslant r$, where i is a Sobol sample, j is a random sample and r is the total number of different pseudorandom sequences.

Appendix: Randomized Quasi-Monte Carlo

- For a randomised set $\tilde{\mathfrak{X}}_i$ we construct a RQMC estimate:

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

  for $0 < j \leqslant r$, where i is a Sobol sample, j is a random sample and r is the total number of different pseudorandom sequences.

- Then, we take their average for *overall RQMC estimation*:

$$RQMC_n = \frac{1}{r} \sum_{j=1}^{r} RQMC_{j,n}$$

Appendix: Randomized Quasi-Monte Carlo

- By independence of the samples we have that for all $0 < j \leqslant r$::

$$Var(RQMC_n) = \frac{Var(RQMC_{j,n})}{r}.$$

Appendix: Randomized Quasi-Monte Carlo

- By independence of the samples we have that for all $0 < j \leqslant r$::

$$Var(RQMC_n) = \frac{Var(RQMC_{j,n})}{r}.$$

- Thus, we have the following *variance estimation*:

$$\widehat{Var}(RQMC_n) = \frac{1}{r(r-1)} \sum_{j=1}^{r} \left(RQMC_{j,n} - RQMC_n \right)^2 .$$

Appendix: Qint (Ermakov & Antonov)

Consider

- A set of random cubature formulas, which were introduced in Ermakov-Granovsky theorem (Ermakov, 1975):

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

Appendix: Qint (Ermakov & Antonov)

Consider

- A set of random cubature formulas, which were introduced in Ermakov-Granovsky theorem (Ermakov, 1975):

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

- The variance of the constructed quadrature formula (Antonov & Ermakov, Vestnik StPU 2015) is:

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

  - Var(MC) is the variance of MC method
  - $a_i = \int_{\mathfrak{X}_i} f(u)\mu(du)$ for $i = 1, 2, ..., N$, where $\mathfrak{X}_i$ is a Haar function set.

Appendix: Qint (Ermakov & Antonov)

- $\mathfrak{X} = [0, 1]^s$. We are using the Sobol sequence (base $b = 2$), so $N = 2^{\Omega}$, where $\Omega \in \mathbb{N}$;

Appendix: Qint (Ermakov & Antonov)

- $\mathfrak{X} = [0,1]^s$. We are using the Sobol sequence (base $b = 2$), so $N = 2^{\Omega}$, where $\Omega \in \mathbb{N}$;
- We choose a splitting $\mathfrak{X}_1, \mathfrak{X}_2, ..., \mathfrak{X}_N$ based on elementary subsets, so that any N-point segment in the form $\{x_{T \cdot N+1}, ..., x_{(T+1) \cdot N}\}$, $T \geq 0$, is guaranteed to be in $\mathfrak{Lat}(i_1, ..., i_N)$.

Appendix: Qint (Ermakov & Antonov)

- $\mathfrak{X} = [0,1]^s$. We are using the Sobol sequence (base $b = 2$), so $N = 2^\Omega$, where $\Omega \in \mathbb{N}$;
- We choose a splitting $\mathfrak{X}_1, \mathfrak{X}_2, ..., \mathfrak{X}_N$ based on elementary subsets, so that any N-point segment in the form $\{x_{T \cdot N+1}, ..., x_{(T+1) \cdot N}\}$, $T \geq 0$, is guaranteed to be in $\mathfrak{Lat}(i_1, ..., i_N)$.
- Some randomization is required: we apply the simplest (subset-preserving) shift $x \longrightarrow x + \xi$. $\xi \in U([0,1]^s)$, where ξ is the same for the whole point set.

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;
2. Choose repetition parameter $R \in \mathbb{N}$ (total number of function evaluations will be $R \cdot 2^{\Omega}$);

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;
2. Choose repetition parameter $R \in \mathbb{N}$ (total number of function evaluations will be $R \cdot 2^{\Omega}$);
3. Split $[0,1]^s$ into $N = 2^{\Omega}$ subsets of equal volume, based on elementary subsets in base 2;

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;
2. Choose repetition parameter $R \in \mathbb{N}$ (total number of function evaluations will be $R \cdot 2^{\Omega}$);
3. Split $[0,1]^s$ into $N = 2^{\Omega}$ subsets of equal volume, based on elementary subsets in base 2;
4. For $r = 1, 2, ..., R$ repeat the following computational cycle:

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;
2. Choose repetition parameter $R \in \mathbb{N}$ (total number of function evaluations will be $R \cdot 2^{\Omega}$);
3. Split $[0, 1]^s$ into $N = 2^{\Omega}$ subsets of equal volume, based on elementary subsets in base 2;
4. For $r = 1, 2, ..., R$ repeat the following computational cycle:
   - Take N next Sobol points $\{x_{(r-1)\cdot N+1}, ...x_{r \cdot N}\}$ and shift them by ξ, evaluate f;

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;
2. Choose repetition parameter $R \in \mathbb{N}$ (total number of function evaluations will be $R \cdot 2^{\Omega}$);
3. Split $[0,1]^s$ into $N = 2^{\Omega}$ subsets of equal volume, based on elementary subsets in base 2;
4. For $r = 1, 2, ..., R$ repeat the following computational cycle:
   - Take N next Sobol points $\{x_{(r-1)\cdot N+1}, ...x_{r\cdot N}\}$ and shift them by ξ, evaluate f;
   - Update the integral estimate $\hat{I} = \frac{1}{rN} \sum_{i=1}^{rN} f(x_i)$;

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;
2. Choose repetition parameter $R \in \mathbb{N}$ (total number of function evaluations will be $R \cdot 2^{\Omega}$);
3. Split $[0,1]^s$ into $N = 2^{\Omega}$ subsets of equal volume, based on elementary subsets in base 2;
4. For $r = 1, 2, ..., R$ repeat the following computational cycle:
   - Take N next Sobol points $\{x_{(r-1)\cdot N+1}, ...x_{r\cdot N}\}$ and shift them by ξ, evaluate f;
   - Update the integral estimate $\hat{I} = \frac{1}{rN}\sum_{i=1}^{rN} f(x_i)$;
   - Update the variance estimate according to the aforecited theorem, where a_j are estimated by an average of r points each: $\hat{a}_j = \frac{1}{r}\sum_{i\in\mathfrak{X}_j} f(x_i)$;

Appendix: Qint (Ermakov & Antonov)

1. Choose partition parameter $\Omega \in \mathbb{N}$;
2. Choose repetition parameter $R \in \mathbb{N}$ (total number of function evaluations will be $R \cdot 2^{\Omega}$);
3. Split $[0,1]^s$ into $N = 2^{\Omega}$ subsets of equal volume, based on elementary subsets in base 2;
4. For $r = 1, 2, ..., R$ repeat the following computational cycle:
   - Take N next Sobol points $\{x_{(r-1)\cdot N+1}, ...x_{r\cdot N}\}$ and shift them by ξ, evaluate f;
   - Update the integral estimate $\hat{I} = \frac{1}{rN}\sum_{i=1}^{rN} f(x_i)$;
   - Update the variance estimate according to the aforecited theorem, where a_j are estimated by an average of r points each: $\hat{a}_j = \frac{1}{r}\sum_{i\in\mathfrak{X}_j} f(x_i)$;
   - For rN function evaluations, build the confidence interval.

Appendix: The Expectation of Monte Carlo Method

Consider the integral $I = \int_a^b f(y)dy$, and a random variable U on $[a, b]$. The expectation of $f(U)$ is:

$$\mathbb{E}[f(U)] = \int_a^b f(y)\varphi(y)dy$$

where φ is the density of U. If U is uniformly distributed on $[a, b]$, then the integral becomes:

$$I = \int_a^b f(y)dy = (b - a)\mathbb{E}[f(U)]$$

Appendix: The Variance of Monte Carlo Method

The variance of the MC estimator is:

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

In practice, the integrand variance σ_f^2 is often unknown. That is why the next estimation for the CI is instead used:

$$\widehat{\sigma}_f^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(f(u_i) - \frac{1}{N} \sum_{i=1}^{N} f(u_i) \right)^2$$

Appendix: The Variance of Randomised Quasi-Monte Carlo Method

By independence of the samples we have that:

$$Var(RQMC_n) = \frac{Var(RQMC_{j,n})}{r}.$$

Thus, we have the following variance estimation:

$$\widehat{Var}(RQMC_n) = \frac{1}{r(r-1)} \sum_{j=1}^{r} \left(RQMC_{j,n} - RQMC_n \right)^2 .$$