Adaptive Discretization for Probabilistic Safety Cost Functions Evaluation – Appendix

Christoph Zimmer(1), Danny Driess(2), Mona Meister(1) and Duy Nguyen-Tuong(1)

[1] Bosch Center for Artificial Intelligence, Robert Bosch GmbH, Renningen, Germany [2] MLR Laboratory, University of Stuttgart, Stuttgart, Germany

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy

A.1 Remaining Proofs

We prove the remaining statements in this subsection.

Proof of Theorem 1. We can show for the differences $\Delta_m := \xi_m - \xi_{m-1}$:

$$\begin{aligned}
\Delta_m &= P(z_1 > 0, \dots, z_m > 0) - P(z_1 > 0, \dots, z_{m-1} > 0) \\
&= P(z_1 > 0, \dots, z_m > 0) - \\
&\quad \Big(P(z_m > 0 | z_1 > 0, \dots, z_{m-1} > 0) + P(z_m \leq 0 | z_1 > 0, \dots, z_{m-1} > 0))\Big) \\
&\quad P(z_1 > 0, \dots, z_{m-1} > 0) \\
&= -P(z_m \leq 0 | z_1 > 0, \dots, z_{m-1} > 0) P(z_1 > 0, \dots, z_{m-1} > 0) \\
&\leq 0.
\end{aligned}$$

Therefore, ξ_m is monotonously decreasing in m. Furthermore, it is bounded by 0 from below as it is a probability. Therefore, it is convergent. ■

Proof of Lemma 1.

$$\begin{aligned}
&P(z_t \leq 0 | z_1 > 0, \dots, z_m > 0) P(z_1 > 0, \dots, z_m > 0) \\
&\quad \geq P(z_t \leq 0 \wedge z_{m+1} > 0 | z_1 > 0, \dots, z_m > 0) P(z_1 > 0, \dots, z_m > 0) \\
&\quad = P(z_t \leq 0 | z_1 > 0, \dots, z_{m+1} > 0) P(z_1 > 0, \dots, z_{m+1} > 0) \quad ■
\end{aligned}$$

Now, we will prove Corollary 1 establishing the relation between m and ϵ_1 for equidistant design.

Proof of Corollary 1. It holds for an equidistant design

$$\Delta = \frac{1}{m-1} < \frac{\delta}{L_1 \gamma + \alpha(\epsilon_1) L_2}. \quad ■$$

Then, we will prove the two components of Theorem 2. The first part bounds the probability of still seeing an unsafe point $z_t \leq 0$ while the neighboring intermediate points are positive and even greater than a δ.

Lemma 2. *Using the same notation and setting as in Theorem 2, we can bound the probability*

$$P(z_t \leq 0 \ \wedge z_1 > 0, \dots, z_m > 0 \wedge (\delta < z_{t_+^m} \leq \gamma \wedge \delta < z_{t_-^m} \leq \gamma)) \leq \epsilon_1$$

if $m-1$ is large enough such that

$$|t_+^m - t_-^m| \leq \frac{\delta}{L_1 \gamma + \alpha(\epsilon_1) L_2}$$

for an arbitrary but fixed $\delta > 0$ and $\gamma > \delta$.

Proof.

$$
\begin{aligned}
& P(z_t \leq 0 \ \wedge z_1 > 0, \ldots, z_m > 0 \wedge z_{t_+^m} > \delta \wedge z_{t_-^m} > \delta)) \\
\leq \quad & P(z_t \leq 0 \ \wedge z_{t_+^m} > \delta) \\
= \quad & \int_{z_t \leq 0} \int_{z_{t_+^m} > \delta} p(z_t, z_{t_+^m}) dz_t dz_{t_+^m} \\
\leq \quad & \int_{z_{\hat{t}} \leq 0} p(z_{\hat{t}} | \hat{z}_{t_+^m}) dz_{\hat{t}} \int_{z_{t_+^m} > \delta} p(z_{t_+^m}) dz_{t_+^m} \\
\leq \quad & P(z_{\hat{t}} \leq 0 | \hat{z}_{t_+^m})
\end{aligned}
$$

where in the second step p denotes a probability density, and we define

$$
(\hat{t}, \hat{z}_{t_+^m}) = \text{argmax}_{t \in T, \delta < z_{t_+^m} \leq \gamma} \int_{z_t \leq 0} p(z_t | z_{t_+^m}) dz_t.
$$

As $z_t | \hat{z}_{t_+^m}$ is a GP, we can denote its mean as μ_{t,t_+^m} and its standard deviation as σ_{t,t_+^m}. For $t = t_+^m$ we can state $\mu_{t_+^m,t_+^m} - \alpha(\epsilon_1)\sigma_{t_+^m,t_+^m} = \hat{z}_{t_+^m} > \delta$. We have Lipschitz constants for μ_{t,t_+^m} and σ_{t,t_+^m}. As stated in the main text L_1 may linearly depend on $\hat{z}_{t_+^m}$. Therefore, we replace it by its upper bound $L_1 \cdot \gamma$, where we would assume γ to be greater than 1. Using these, we obtain $\mu_{t,t_+^m} \geq \mu_{t_+^m,t_+^m} - L_1\gamma|t_+^m - t|$ and $\sigma_{t,t_+^m} \geq \sigma_{t_+^m,t_+^m} - L_1|t_+^m - t|$. Therefore, it follows that at $\hat{t}$ the lower bound of the quantile of the GP $z_{\hat{t}} | \hat{z}_{t_+^m}$ is defined by $\delta - L_1\gamma|t_-^m - t_+^m| - \alpha(\epsilon_1)L_2|t_-^m - t_+^m|$ (blue line of Figure A.1). If this value is greater than zero, it means that the GPs probability of being smaller than zero can be bounded by ϵ_1:

$$
\begin{aligned}
& P(z_{\hat{t}} \leq 0 | \hat{z}_{t_+^m}) \\
\leq \quad & \begin{cases} \epsilon_1 & \text{if } \delta - L_1\gamma|t_-^m - t_+^m| - \alpha(\epsilon_1)L_2|t_-^m - t_+^m| \geq 0 \text{ (case 1)} \\ 1 & \text{otherwise (case 2)} \end{cases}
\end{aligned}
$$

According to the requirement on m, we are always in case 1. ■

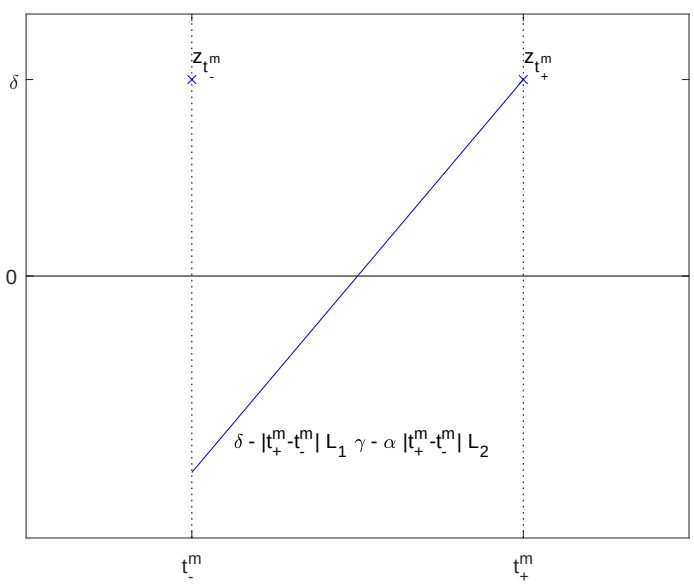


Figure A.1: Two safe points t_-^m and t_+^m with $z_{t_-^m} > \delta$ and $z_{t_+^m} > \delta$. Solid blue line: lower bound of mean minus α standard deviations of predictive distribution.

Next, we focus on the probability that one of the neighboring intermediate points of a new point $\boldsymbol{x}_t$ are smaller than δ. We can bound the probability of a Gaussian distribution to be in an interval $[0, \delta]$ as follows

Lemma 3. *Under Assumption 1, for every $\epsilon_3 > 0$, there exists a $\delta > 0$ such that it holds for any $t \in T$ that $P(z_t \in [0, \delta]) < \epsilon_3$.*

Proof. If the GP is degenerated for a point t with $\sigma_t = 0$, then (according to Assumption 1), either $\mu_t < -\alpha_1$ or $\mu_t > \alpha_1$. In both cases, the inequality holds for any $\delta < \alpha_1$ and any ϵ_3. Let us assume $\sigma_t > \alpha_2$ from now on.

Define $\delta = \epsilon_3 \cdot \sqrt{2\pi}\beta$. Then it holds for all $t \in T$

$$\begin{aligned} P(z_t \in [0, \delta]) &= \int_0^\delta \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{(z-\mu_t)^2}{2\sigma_t}\right) dz \\ &\leq \delta \frac{1}{\sqrt{2\pi}\sigma_t} \\ &< \delta \frac{1}{\sqrt{2\pi}\alpha_2} \\ &\leq \delta \frac{1}{\sqrt{2\pi}\beta} \\ &\leq \epsilon_3. \end{aligned}$$

■

This results enables us to prove the remaining claim, namely that the probability of one of the neighboring points being smaller than δ

Lemma 4. *Assuming, that we have a set of $m-1$ points $\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{m-1}$ and introduce one new point $\boldsymbol{x}_{t_m}$. We can bound the probability*

$$P(z_t \leq 0 \ \wedge z_1 > 0, \ldots, z_m > 0 \wedge (z_{t_+^m} \leq \delta \vee z_{t_-^m} \leq \delta)) < \epsilon_2$$

for arbitrary $\epsilon_2 > 0$.
Set $\epsilon_3 = \epsilon_2/2$ and $\delta = \epsilon_3 \cdot \sqrt{2\pi}\beta$ (as in Lemma 3). β is the bound on GP degeneracy from Assumption 1.

Proof.

$$\begin{aligned} & P(z_t \leq 0 \ \wedge z_1 > 0, \ldots, z_m > 0 \wedge (0 < z_{t_+^m} \leq \delta \vee 0 < z_{t_-^m} \leq \delta)) \\ = \ & P(z_t \leq 0 \ \wedge z_1 > 0, \ldots, z_m > 0 | (0 < z_{t_+^m} \leq \delta \vee 0 < z_{t_-^m} \leq \delta)) \cdot P(0 < z_{t_+^m} \leq \delta \vee 0 < z_{t_-^m} \leq \delta) \\ \leq \ & 1 \cdot P(0 < z_{t_+^m} \leq \delta \vee 0 < z_{t_-^m} \leq \delta) \\ < \ & \epsilon_3 + \epsilon_3 \\ = \ & \epsilon_2 \end{aligned}$$

Note that $P(z_{t_+^m} = 0) = P(z_{t_-^m} = 0) = 0$ as a property of the Gaussian distribution. The last inequality is a consequence of the Lemma 3. ■

Lemma 4 holds independently of m.

Lemma 5. *For every $\epsilon_4 > 0$, there exists a $\gamma > 0$ such that it holds for any $t \in T$ that $P(z_t > \gamma) < \epsilon_4$:*

$$\gamma = \sqrt{2}\bar{\sigma} \log\left(\frac{1}{\epsilon_4}\right) + \bar{\mu}$$

with $\bar{\sigma}$ and $\bar{\mu}$ defined in the main text.

Proof.

$$\begin{aligned}
P_{\mu_t,\sigma_t}(z_t > \gamma) &= P_{(0,1)}(x\ \sigma_2 + \mu_t > \gamma) \\
&= P_{(0,1)}\left(x > \frac{\gamma - \mu_t}{\sigma_2}\right) \\
&= 1 - \Phi\left(\frac{\gamma - \mu_t}{\sigma_2}\right) \\
&= \Phi\left(-\frac{\gamma - \mu_t}{\sigma_2}\right) \\
&= 1/2 \text{ erfc}\left(\frac{\frac{\gamma-\mu_t}{\sigma_2}}{\sqrt{2}}\right) \\
&\leq \exp\left(-\frac{\gamma - \mu_t}{\sqrt{2}\sigma_2}\right) \\
&\leq \exp\left(-\frac{\gamma - \bar{\mu}}{\sqrt{2}\bar{\sigma}}\right) \\
&\leq \epsilon_4.
\end{aligned}$$

■

Taken together, we can prove Theorem 2:

Proof of Theorem 2. We use results from Lemma 2, i.e. $P(z_t \leq 0\ \wedge z_1 > 0, \ldots, z_m > 0 \wedge (\delta < z_{t_+^m} \leq \gamma \wedge \delta < z_{t_-^m} \leq \gamma)) \leq \epsilon_1$ under the conditions above, and Lemma 4, stating $P(z_t \leq 0\ \wedge z_1 > 0, \ldots, z_m > 0 \wedge (z_{t_+^m} \leq \delta \vee z_{t_-^m} \leq \delta)) < \epsilon_2$, and Lemma 5, stating $P(z_t > \gamma) < \epsilon_4$. When employing the following bound

$$\begin{aligned}
&P(z_t \leq 0\ \wedge z_1 > 0, \ldots, z_m > 0 \wedge (z_{t_+^m} > \gamma \vee z_{t_-^m} > \gamma)) \\
&\leq P(z_{t_+^m} > \gamma \vee z_{t_-^m} > \gamma) \\
&\leq 2 \cdot \max_{t \in \{t_-^m, t_+^m\}} P(z_t > \gamma) \\
&< 2\epsilon_4
\end{aligned}$$

it follows

$$\begin{aligned}
&P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot P(z_1 > 0, \ldots, z_m > 0) \\
&= P(z_t \leq 0 \wedge z_1 > 0, \ldots, z_m > 0) \\
&\leq P(z_t \leq 0 \wedge z_1 > 0, \ldots, z_m > 0 \wedge (\delta < z_{t_+^m} \leq \gamma \wedge \delta < z_{t_-^m} \leq \gamma)) \\
&\quad + P(z_t \leq 0 \wedge z_1 > 0, \ldots, z_m > 0 \wedge (z_{t_+^m} \leq \delta \vee z_{t_-^m} \leq \delta)) \\
&\quad + P(z_t \leq 0 \wedge z_1 > 0, \ldots, z_m > 0 \wedge (z_{t_+^m} > \gamma \vee z_{t_-^m} > \gamma)) \\
&< \epsilon_1 + \epsilon_2 + 2\epsilon_4 \\
&\leq \epsilon \quad ■
\end{aligned}$$

Now, we have completed the proofs for the claim that we can find an m for any ϵ and bound the probability in Eq. 5. Next, we focus on the convergence rate. We will need a few more intermediate results. First, we need to establish a relation between $\alpha(\epsilon_1)$ and ϵ_1.

Corollary 2. *Lemma 2 still holds for any $\alpha'(\epsilon_1) \geq \alpha(\epsilon_1)$.*

Lemma 6. *For any $\epsilon_1 > 0$ we can express $\alpha'(\epsilon_1)$ in terms of ϵ_1 as*

$$\alpha'(\epsilon_1) = \sqrt{2} \log\left(\frac{1}{\epsilon_1}\right).$$

Proof. We need to show that this $\alpha'(\epsilon_1)$ fulfills $P_{N(0,1)}([\mu - \alpha'(\epsilon_1)\sigma, \mu + \alpha'(\epsilon_1)\sigma]) \geq 1 - \epsilon_1$ for a standard normal distribution, hence $P_{N(0,1)}([-\alpha'(\epsilon_1), \alpha'(\epsilon_1)]) \geq 1 - \epsilon_1$. This is equivalent to

$$2\Phi(-\alpha'(\epsilon_1)) \leq \epsilon_1$$

Now, it holds

$$\begin{aligned} & 2\Phi(-\alpha'(\epsilon_1)) \\ = \quad & \operatorname{erfc}\left(-\frac{-\alpha'(\epsilon_1)}{\sqrt{2}}\right) \\ \leq \quad & \exp\left(-\frac{\alpha'(\epsilon_1)}{\sqrt{2}}\right) \\ = \quad & \exp\left(-\frac{\sqrt{2}\log\left(\frac{1}{\epsilon_1}\right)}{\sqrt{2}}\right) \\ = \quad & \epsilon_1 \end{aligned}$$

■

Now, we can precisen the relation between m and ϵ

Corollary 3. *It is a consequence of Corollary 1 and Lemma 6, that*

$$m > \frac{L_1\gamma + \sqrt{2}\log(\frac{1}{\epsilon_1})L_2}{\delta} + 1$$

is sufficient for the claim of Corollary 1.

Now, we can finally proof Theorem 3.

Proof of Theorem 3. Let $\epsilon > 0$ and $\epsilon_3 = \epsilon/4$, $\epsilon_1 = \epsilon/2$, and $\epsilon_4 = \epsilon/8$. Taking together the result from Corollary 3 and Theorem 2 we obtain

$$m \geq \frac{L_1\gamma + \sqrt{2}\log(\frac{1}{\epsilon_1})L_2}{\epsilon_3 \cdot \sqrt{2\pi}\beta} + 1.$$

γ can be expressed as $\gamma = \sqrt{2}\bar{\sigma}\log(\frac{1}{\epsilon_4}) + \bar{\mu}$ (Theorem 2) and it follows

$$m \geq \frac{1}{\epsilon_3}\frac{L_1\sqrt{2}\bar{\sigma}\log(\frac{1}{\epsilon_4}) + \bar{\mu}}{\sqrt{2\pi}\beta} + \frac{L_2}{\sqrt{\pi}\beta}\frac{\log\left(\frac{1}{\epsilon_1}\right)}{\epsilon_3} + 1 =: \tilde{m}_0$$

For any $\kappa > 0$, there exists an ϵ_1^0 (and similar for ϵ_4) such that for every $\epsilon_1 < \epsilon_1^0$, it holds $\log(1/\epsilon_1) < (1/\epsilon_1)^\kappa$ and, therefore, for every $\epsilon_1 < \epsilon_1^0$

$$\begin{aligned} \tilde{m}_0 \quad < \quad & \frac{1}{\epsilon_3}\frac{L_1\sqrt{2}\bar{\sigma}(\frac{1}{\epsilon_4})^\kappa + \bar{\mu}}{\sqrt{2\pi}\beta} + \frac{L_2}{\sqrt{\pi}\beta}\frac{\left(\frac{1}{\epsilon_1}\right)^\kappa}{\epsilon_3} + 1 \\ < \quad & \frac{4}{\epsilon}\frac{L_1\sqrt{2}\bar{\sigma}(\frac{8}{\epsilon})^\kappa + \bar{\mu}\frac{1}{\epsilon^\kappa}}{\sqrt{2\pi}\beta} + \frac{L_2}{\sqrt{\pi}\beta}\frac{\left(\frac{2}{\epsilon}\right)^\kappa 4}{\epsilon} + 1 \\ \leq \quad & \frac{1}{\epsilon^{1+\kappa}}4\frac{L_1\sqrt{2}\bar{\sigma}8^\kappa + \bar{\mu}}{\sqrt{2\pi}\beta} + \frac{L_2}{\sqrt{\pi}\beta}\frac{2^\kappa 4}{\epsilon^{1+\kappa}} + \epsilon^{-(1+\kappa)} \\ = \quad & C\ \epsilon^{-(1+\kappa)} =: m_0 \end{aligned}$$

and defining C as

$$C := 4\frac{L_1\sqrt{2}\bar{\sigma}8^\kappa + \bar{\mu}}{\sqrt{2\pi}\beta} + \frac{L_2}{\sqrt{\pi}\beta}2^\kappa 4 + 1$$

Now, we can rephrase to

$$\epsilon = m_0^{-\frac{1}{1+\kappa}}C$$

and further to

$$\lceil m_0 \rceil^{-\frac{1}{1+\kappa}}C =: \tilde{\epsilon} \leq \epsilon = m_0^{-\frac{1}{1+\kappa}}C$$

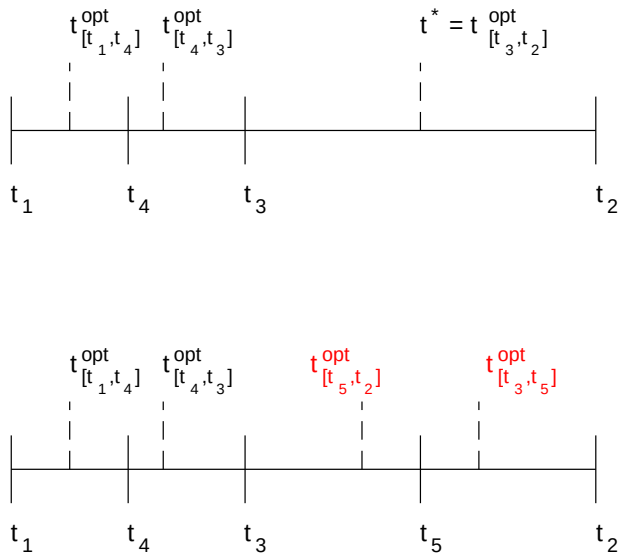


Figure A.2: Illustration of computation scheme: Upper part showing previous iteration 4 and four points $x_{t_1}, \ldots, x_{t_4}$ and corresponding local optima in the intervals denoted as $t^{opt}_{[\cdot]}$. The best of those local optima denoted as t^*. In the next iteration (lower part of Figure), we add the new point (from the previous) iteration and recompute the local optima on the adjacent intervals (red). We use the previous values for the other intervals. As we expect the impact of a new point to become less for more distant intervals, this is a reasonable approximation. The number of neighboring intervals in which the local optima is recomputed depends on the user's choice.

Now, for each $\epsilon > 0$ there exists a $0 < \tilde{\epsilon} \leq \epsilon$ and Eq. (5) of Theorem 2 let us obtain

$$P(z_t \leq 0 | z_1 > 0, \ldots, z_{\lceil m_0 \rceil} > 0) \cdot P(z_1 > 0, \ldots, z_{\lceil m_0 \rceil} > 0) < \tilde{\epsilon} = \lceil m_0 \rceil^{-\frac{1}{1+\kappa}} C \leq \epsilon$$

for any $t \in T$, leading to a convergence rate of $\frac{1}{1+\kappa}$.

This can be rephrased to

$$P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot \xi_m = \mathcal{O}\left(m^{-\frac{1}{1+\kappa}}\right). \qquad \blacksquare$$

A.2 Further Remarks on Main Text

A.2.1 How to compute the solution of the optimization problem in Eq. 4

The optimization problem in Eq. (4) has $m-1$ local optima, as the objective function $P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot P(z_1 > 0, \ldots, z_m > 0)$ is greater than zero for $t \neq t_i, i = 1, \ldots, m$ but zero at $t = t_i, i = 1, \ldots, m$. In practice, we use the following scheme for speeding up the computation. We save the results of the $m-1$ local optima in iteration m as $t^*_1, \ldots, t^*_{m-1}$. In iteration $m+1$, we use $m-2$ of them, namely all those for which it holds $t^* \notin [t^{*m}_-, t^{*m}_+]$, where t^{*m}_- is the preceding point of t^* and t^{*m}_+ is the following point of t^*. Thus, it is only required to solve two optimization problems: one on $[t^{*m}_-, t^*]$ and one on $[t^*, t^{*m}_+]$ as illustrated in Figure A.2.

A.2.2 Remarks on related work on excursion sets [Azzimonti et al., 2016, Azzimonti et al., 2019]

The references [Azzimonti et al., 2016, Azzimonti et al., 2019] use a fundamentally different approach than our work: Given a function f and a (safety) threshold u > 0, they focus on the probability of the set $\Gamma = \{x \in X : f(x) > u\}$. This is

the probability of the set Γ of safe points. We focus on $P(x_1 > 0, \ldots, x_m > 0)$. This is the probability to be safe at all m discretization points. These two definitions of cost/loss/safety are fundamentally different: for example, let x to vary in $[0, T]$ and

$$f(x) = \begin{cases} 2u & x < t_2 \\ -u & x = x_{t_2} \end{cases}$$

so f is larger than u for almost all x except $x = x_{t_2}$. Failure occurs only for $x = x_{t_2}$. Their criterion would return 1 (if uniform distribution over $[t_1, t_2]$), indicating safety. Our criterion would return zero, indicating failure.

Another difference between their work and ours is the following: they focus on finding a smaller set of sample point best representing a bigger set of sample points (with respect ot the above mentioned criterion). We, however, try to adaptively find a set of good points (reducing the remaining risk for safety valuation). We do not start from a bigger set, however.

A.3 Algorithms of GP-PRM of Section 6

Here we state the pseudo code of the algorithms for the GP-PRM from Sec. 6. Algorithm 2 is the creation of the probabilistic roadmap, where in step 11 the safety of an edge is evaluated with our proposed adaptive discretization algorithm.

For sampling from X_{free} itself (step 5), many strategies have been proposed in the literature, e.g. uniform sampling. In the experiments of this work, a new configuration is drawn from X_{free} such that it has maximum distance to the already sampled configurations in V.

As mentioned in the main text, Sec. 6, given the constructed probabilistic roadmap (V, E), planning a path from a target configuration to a goal configuration reduces to a graph search problem. Next we describe the two approaches we followed in this work, the weighting approach and the waypoint approach in more detail.

To generate a variety of candidate paths on the roadmap, one idea is to first sample a waypoint $\boldsymbol{x}_w \in V$. The candidate path then consists of the shortest path from the start, over the way-point, to the target configuration (see Algorithm 3). After checking whether the path contains no vertex twice, the resulting safety probability of the complete path is determined adaptively with our proposed algorithm. Optionally, testing candidate paths could be terminated if a desired safety is reached. However, in the experiments we calculate every possible candidate path over all way-points.

The other approach is to compute a weighted shortest path problem on the PRM (V, E), where the weight of each edge $(\boldsymbol{x}, \boldsymbol{x}') \in E$ is given by $w_{\boldsymbol{x},\boldsymbol{x}'} = d(\boldsymbol{x}, \boldsymbol{x}') + \rho\,(1 - p(\boldsymbol{x}, \boldsymbol{x}'))$, where $d(\boldsymbol{x}, \boldsymbol{x}')$ is the euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, $\rho \in \mathbb{R}^+$ a scaling parameter and $p(\boldsymbol{x}, \boldsymbol{x}')$ the safety probability of the path connecting $\boldsymbol{x}$ and $\boldsymbol{x}'$, which is calculated during the construction of the roadmap.

Algorithm 2 GP-PRM Construction

1: **Input**: number of sampled states n_V, number of nearest neighbors k_{NN}, $p_{\min}$.
2: **Output:** PRM (V, E)
3: **Initialization**: $V = \emptyset, E = \emptyset$
4: **while** $|V| < n_V$ **do**
5: $\quad \boldsymbol{x} = \text{Sample}(X_{\text{free}})$
6: $\quad V \leftarrow V \cup \{\boldsymbol{x}\}$
7: **end while**
8: **for all** $\boldsymbol{x} \in V$ **do**
9: $\quad N_{\text{near}} = \text{NearestNeighbors}(\boldsymbol{x}, k_{\text{NN}}, V)$
10: $\quad$ **for all** $\boldsymbol{x}'$ **do**
11: $\quad\quad p = \text{EvaluateSafety}((\boldsymbol{x}, \boldsymbol{x}'), \epsilon)$
12: $\quad\quad$ **if** $p > p_{\min}$ **then**
13: $\quad\quad\quad E \leftarrow E \cup \{(\boldsymbol{x}, \boldsymbol{x}')\}$
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: **end for**
17: remove all unconnected $\boldsymbol{x} \in V$ from V

Algorithm 3 GP-PRM Waypoint Planning

1: **Input**: Start $\boldsymbol{x}_s$, goal $\boldsymbol{x}_g$, $\epsilon > 0$, GP cost function g, PRM (V, E)
2: $\boldsymbol{x}'_s = \text{NearestNeighbor}(\boldsymbol{x}_s, V)$
3: $\boldsymbol{x}'_g = \text{NearestNeighbor}(\boldsymbol{x}_g, V)$
4: **Output:** Path $\boldsymbol{\tau}_{sg}$ with safety indicator p_{best}
5: **Initialization**: $p_{\text{best}} = 0$, $\boldsymbol{\tau}_{\text{best}} = ()$
6: **for all** $\boldsymbol{x}_w \in \text{Shuffle}(V)$ **do**
7: $\quad \boldsymbol{\tau}_{sw} = \text{ShortestPath}(\boldsymbol{x}'_s, \boldsymbol{x}'_w)$
8: $\quad \boldsymbol{\tau}_{wg} = \text{ShortestPath}(\boldsymbol{x}'_w, \boldsymbol{x}'_g)$
9: $\quad \boldsymbol{\tau} = \text{Concatenate}(\boldsymbol{\tau}_{sw}, \boldsymbol{\tau}_{wg})$
10: $\quad$ **if** $\text{SimplePath}(\boldsymbol{\tau})$ **then**
11: $\quad\quad p = \text{EvaluateSafety}(\boldsymbol{\tau}, \epsilon)$
12: $\quad\quad$ **if** $p > p_{\text{best}}$ **then**
13: $\quad\quad\quad p_{\text{best}} \leftarrow p$
14: $\quad\quad\quad \boldsymbol{\tau}_{\text{best}} \leftarrow \boldsymbol{\tau}$
15: $\quad\quad$ **end if**
16: $\quad$ **end if**
17: **end for**

A.4 Further Evaluations & Experimental Details

A.4.1 Experimental Details

Here we state all hyperparameters of the experiments performed in Section 7.

	Adaptive		Equidistant	
	Creation	Planning	Creation	Planning
Fig. 2b	7.8 ± 0.1	51 ± 4	14 ± 2.7	66 ± 6
Fig. 2c	8.2 ± 0.2	53 ± 3	16 ± 2.0	74 ± 2
Fig. 2d	7.3 ± 0.7	48 ± 1	16 ± 2.1	85 ± 7
Fig. 2e	7.7 ± 0.2	48 ± 3	13 ± 2.0	70 ± 3

Table A.1: Runtime in seconds for creation and planning of scenarios considered in section 7.2.

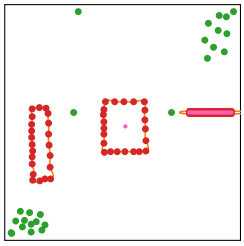


Figure A.3: Data points of the GP for the experiment with the thin obstacle (Section 7.1). Red are on-surface observations, green collision free and pink observations inside an object.

In Section 7.1 and 7.2, the number of Monte-Carlo samples to evaluate $\xi_m(\boldsymbol{\tau})$ was 3000, in Section 7.3 1000, the precision to terminate the adaptive evaluation algorithm was $\epsilon = 0.01$ in Section 7.1 and 7.2 and $\epsilon = 0.02$ in Section 7.3. The number of samples for constructing the roadmap was $n_V = |V| = 100$ in Section 7.1 and 7.2 and $n_V = |V| = 230$ in Section 7.3. The minimum safety probability to add an edge to the graph was $p_{\min} = 0.7$ in Section 7.1 and 7.2. For the workspace exploration experiment in Section 7.3, the construction of the roadmap for exploration used $p_{\min} = 0.7$ and $p_{\min} = 0.8$ for the roadmap for planning the evaluation paths between the targets. All experiments utilized a Gaussian kernel. Its hyperparameters was $\sigma_g^2 = 0.5$ in 7.1 and 7.2, the length-scale $\Lambda_g = \text{diag}(1/l, \ldots, 1/l)$, where $l = 0.05$ in Section 7.1, $l = 0.08$ in Section 7.2 and $l = 0.07$ in Section 7.3. Furthermore, $\sigma = 0.1$ in Section 7.2, $\sigma = 0.01$ in Section 7.1 and $\sigma = 0.05$ in Section 7.3. The number of nearest neighbors was $k_{\text{NN}} = 5$ in Sections 7.2, 7.3 and $k_{\text{NN}} = 10$ in Section 7.1. In Sections 7.1 and 7.2, given the constructed roadmap, the paths are planned using the waypoint approach, in Section 7.3, the weighting approach is used with $\rho = 10.0$.

The target values in the dataset for the GP implicit surface model are as follows: In Sections 7.1 and 7.2, the values are $z(\boldsymbol{x}) = 0$ for points directly on the surface and $z(\boldsymbol{x}) = 1$ in free space. The experiment in Section 7.1 also has observations inside an object (to generate such a thin obstacle) with $z(\boldsymbol{x}) = -1$. In Section 7.3, it is $z(\boldsymbol{x}) = 0.3$ for a free space observation and $z(\boldsymbol{x}) = -0.2$ if the robot has touched the surface. This means that in this case, the training points are independent from the distance to the object, i.e. that there are either on-surface observations or free-space observations, each with the same value of $z(\boldsymbol{x})$. In implicit surface modeling with GPs, this is a common technique [Driess et al., 2017]. However, one could also interpret the GP as a signed distance function if the observations are actual distances to the obstacles, if such sensor information is available.

We use our own GP implementation, because we have to calculate the full covariance matrix prediction efficiently, which most GP packages do not allow. In order to do so, we save the Cholesky decomposition of the kernel matrix $\boldsymbol{K}_n + \lambda^2 \boldsymbol{I}$, such that at prediction time, only triangular systems have to be solved.

In Table 1, equidistant coarse means that the maximum distance between two discretization points before the algorithm terminates was 0.012, equidistant fine means a maximum distance of 0.006.

Figure A.3 shows the datapoints for the experiment in Section 7.1. As one can see, relatively many points are necessary to generate the thin obstacle.

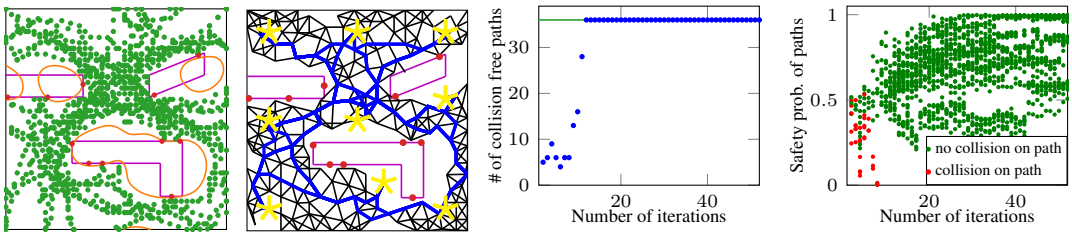
The runtimes given in Tables A.1 and 1 are determined on a laptop with an Intel© Core™ i7-6500U CPU 2.50GHz x 4. All experiments have been implemented in Python 3.5.2 using NumPy for calculations.

For the scenario considered in Section 7.2 that do not contain thin obstacles, the equidistant evaluation leads to the similar paths without any violation of the obstacles. However, as summarized in Table A.1, the runtime for the equidistant evaluation is considerably higher compared to our proposed adaptive scheme. Here the equidistant algorithm iterates until the predictive covariance matrix becomes numerically indefinite.

A.4.2 Exploration and Safe Path Planning for Autonomous Robot – Further Runs

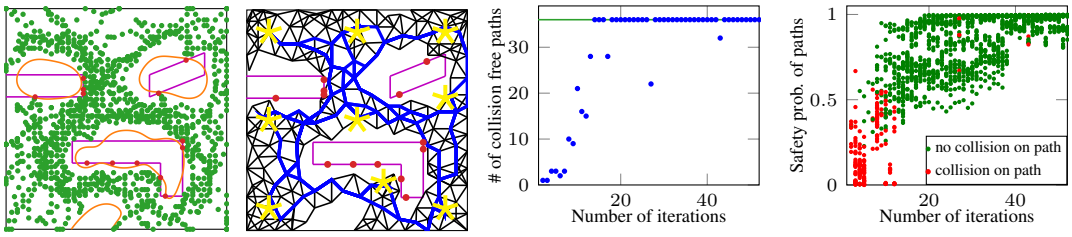
Here, in Figures A.4 - A.12, we show 9 further runs of the workspace exploration experiment from Section 7.3 with the same hyperparameters, but different random seeds. Please note that the runs that contain collision paths even at later

stages of the exploration does not mean that our proposed algorithm falsely calculates unsafe paths. Instead, the GP does not contain the right information (yet) about the obstacles in the environment (both in terms of collected data and chosen kernel hyperparameters). This particularly happens if many free space data points haven been observed which are very close to an obstacle, but no obstacle data is observed in the same region. Nevertheless, as can be seen, in all cases in the end of the exploration, all 36 path between the 9 evaluation targets are collision free.



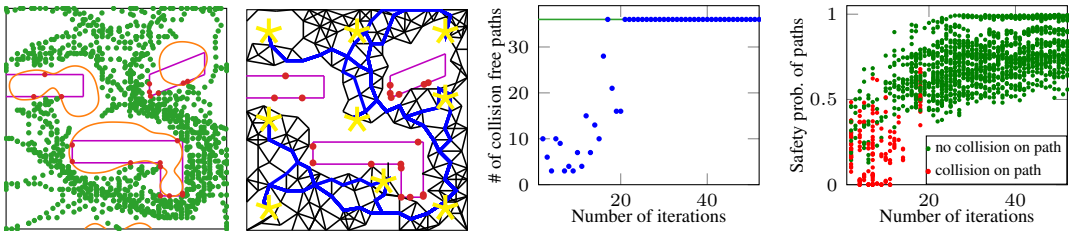
(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.4: Workspace exploration result after 51 iterations. Violet are the true obstacles, orange in (a) the obstacle estimations based on the GP data. Green points in (a), (b) are collision free observations, red obstacle observations. The yellow stars in (b) are the 9 desired targets with the planned paths in blue. (c) shows the number of planned collision free paths between the 9 targets (36 are possible in total), (d) the safety probabilities of these paths.



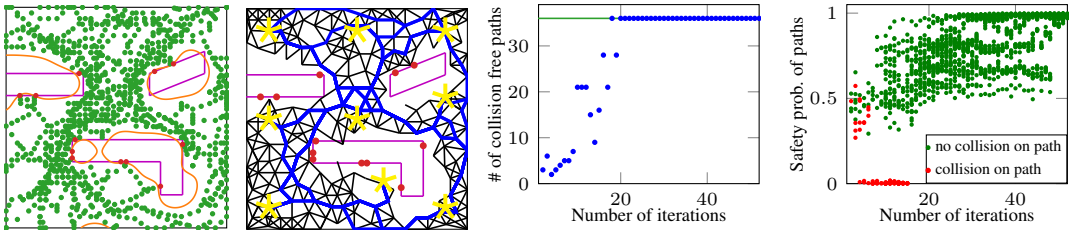
(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.5: Workspace exploration result after 51 iterations.



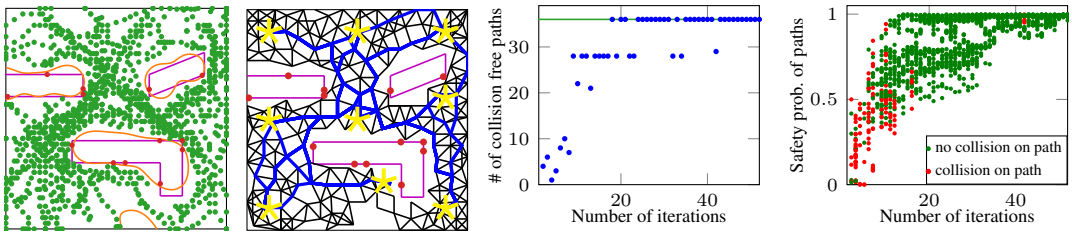
(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.6: Workspace exploration result after 51 iterations.



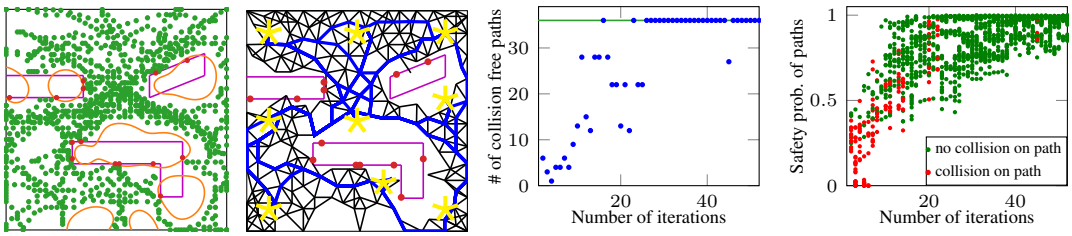
(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.7: Workspace exploration result after 51 iterations.



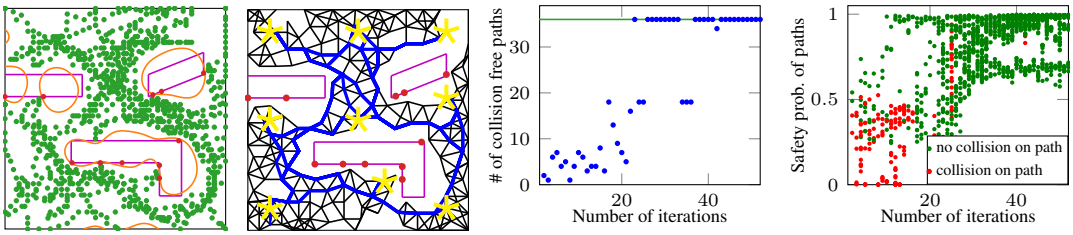
(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.8: Workspace exploration result after 51 iterations.



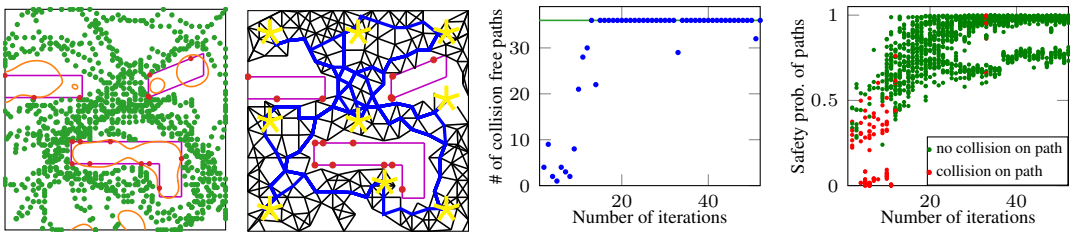
(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.9: Workspace exploration result after 51 iterations.



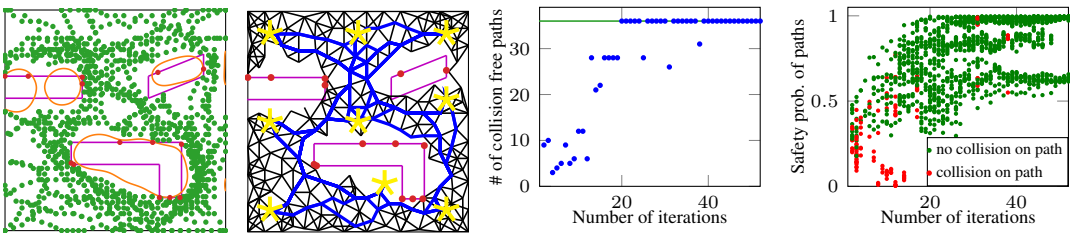
(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.10: Workspace exploration result after 51 iterations.



(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.11: Workspace exploration result after 51 iterations.



(a) Collected data (b) Planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure A.12: Workspace exploration result after 51 iterations.

A.4.3 Additional Toy Example

Use a toy function as ground truth safety model: $g : [5, 25] \to \mathbb{R}$

$$g(x) = 0.3 + 1/80(x - 5) + 1/4 \ \sin(1000/x^2)$$

and $z = g(x) + \varepsilon$ with $\varepsilon \sim N(0, 0.05)$. This is a function with two minima and three maxima two of which occur at the boundaries. Sample 1000 noisy training points and train a GP and use this GP as current safety model to evaluate the cost of a trajectory τ from $t_1 = 8$ to $t_2 = 17$. We let the Algorithm 1 run until $\epsilon \leq 0.001$. For comparison we use an algorithm that places the points according to a grid: this benchmark also starts with $t_1 = 8$ and $t_2 = 17$ and places the third point in the middle, $t_3 = 1/2(t_1 + t_2)$. Then, the fourth and fifth point are set as $t_4 = 1/2(t_1 + t_3)$ and $t_5 = 1/2(t_3 + t_2)$.

Results are shown in Figure A.13, comparing the Algorithm 1 with a benchmark with 1000 iterations. The Algorithm 1 converges much faster in terms of iterations and also reaches a better final result as Figure A.14 shows. As MC sampling introduces an uncertainty for the evaluation of our termination criterion, we display this effect in Figure A.15 showing the precise termination of our algorithm. A video (see Appendix File) illustrates how the adaptive Algorithm 1 works compared to an equidistant benchmark for the first 50 iterations on this example.

A.5 Appendix - Files

Adaptive Cost Evaluation algorithm – video. File: Adapt-Cost-Eval-in-action.avi
Upper panel: Path cost estimate; lower panel: An equidistant benchmark placing points in green colored circels and our Adaptive Cost Evaluation algorithm optimally placing point in red asteriscs. The optimally placed points lead much quicker to a good cost safety estimate.

Robot Exploration Experiment – video. File: robotExploration.mov
Video on experiment of exploration and path planning of a vacuum cleaner robot

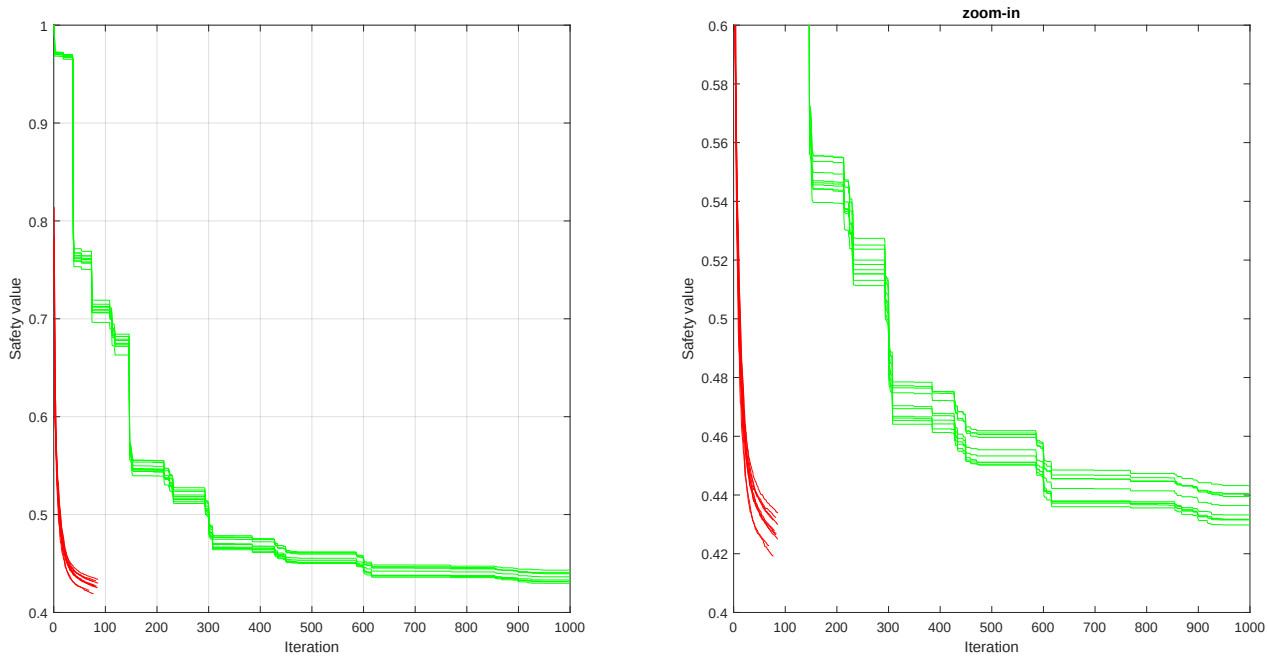


Figure A.13: Left panel: The adaptive Algorithm 1 in red, and an equidistant benchmark in green in dependence of number of iterations. Right panel: zoom-in.

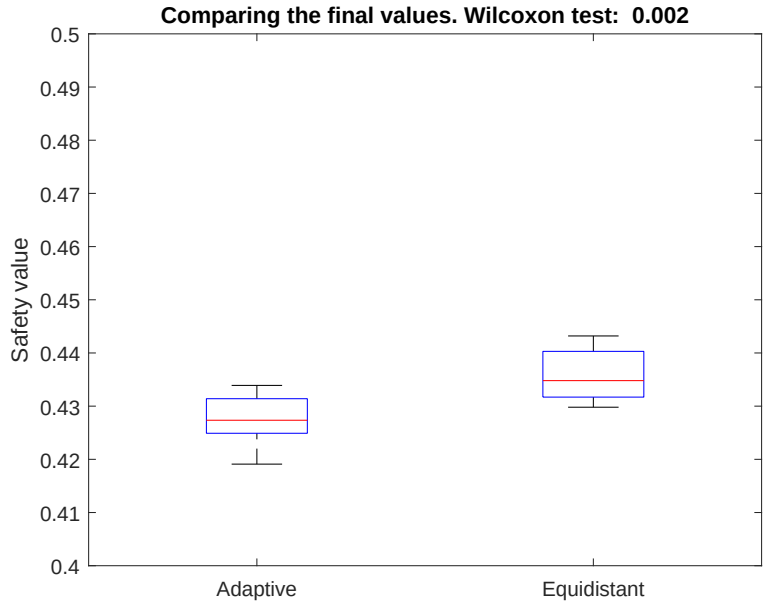


Figure A.14: Final results of the adaptive Algorithm 1 (left) and the equidistant benchmark (right). A Wilcoxon text yields a p-value of 0.002 indicating an improved outcome. This means that the adaptive Algorithm 1, even though using a lot less iterations and intermediate points, achieves an even better final results as the equidistant benchmark using much more intermediate points.

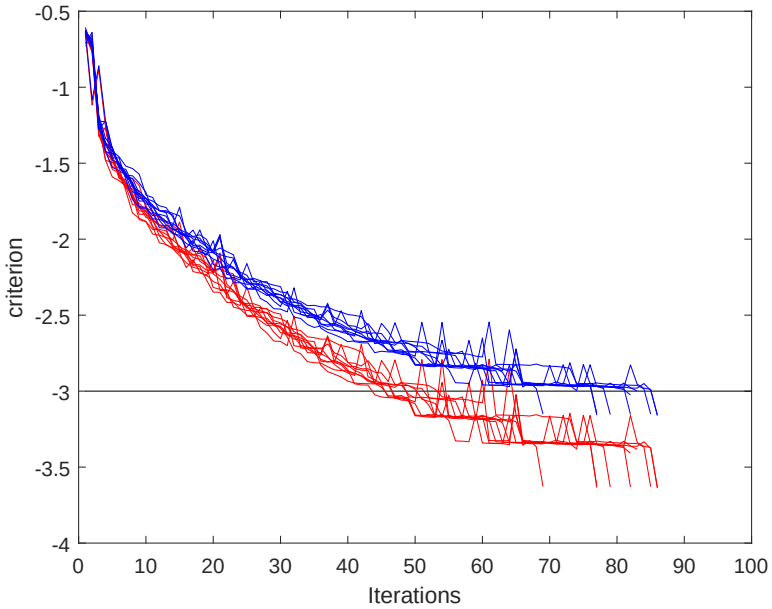


Figure A.15: Termination criterion of Algorithm 1 and the uncertainty from MC sampling: Red color displays the safety value as in Figure A.13 and blue color the safety value plus two standard deviations from the MC sampling in a logarithmic plot. Black line is the termination criterion.