GUARANTEED ADAPTIVE MONTE CARLO METHODS FOR ESTIMATING MEANS OF RANDOM VARIABLES

BY

LAN JIANG

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Applied Mathematics
in the Graduate College of the
Illinois Institute of Technology

Approved 
Advisor

Chicago, Illinois
May 2016

ACKNOWLEDGMENT

This dissertation could not have been written without Dr. Fred J. Hickernell, who not only served as my supervisor but also encouraged and challenged me throughout my academic program. He and the other faculty members, especially Dr. Greg Fasshauer guided me through the dissertation process, never accepting less than my best efforts. I also thank my committee members, Dr. Lulu Kang and Dr. Gady Agam, for their encouragement, insightful comments, and hard questions. I appreciate the joint work with Dr. Art Owen and all the GAIL team members including Dr. Sou-Cheng Choi, Yuhan Ding, Yizhi Zhang, Lluís Antoni Jiménez Rugama, Xin Tong and Xuan Zhou. I would also like to thank my parents, my husband, and my kids for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Monte Carlo is a versatile computational method that may be used to approximate the means, μ, of random variables, Y, whose distributions are not known explicitly. This thesis investigates how to reliably construct fixed width confidence intervals for μ with some prescribed absolute error tolerance, ε_a, relative error tolerance, ε_r or some generalized error criterion. To facilitate this, it is assumed that the kurtosis, κ, of the random variable, Y, does not exceed a user specified bound $\kappa_{\max}$. The key idea is to confidently estimate the variance of Y by applying Cantelli's Inequality. A Berry-Esseen Inequality makes it possible to determine the sample size required to construct such a confidence interval. When relative error is involved, this requires an iterative process. This idea for computing $\mu = \mathbb{E}(Y)$ can be used to develop a numerical integration method by writing the integral as $\mu = \mathbb{E}(f(\boldsymbol{x})) = \int_{\mathbb{R}^d} f(\boldsymbol{x})\rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, where $\boldsymbol{x}$ is a d dimensional random vector with probability density function ρ. A similar idea is used to develop an algorithm for computing $p = \mathbb{E}(Y)$ where Y is a Bernoulli random variable. All of the algorithms have been implemented in the Guaranteed Automatic Integration Library (GAIL).

CHAPTER 1

INTRODUCTION

1.1 Monte Carlo simulation

Monte Carlo is a widely used simulation method that can be applied to evaluate means of random variables, perform numerical integration, estimate probabilities, understand and control complex stochastic systems, price financial derivatives and solve other real world problems.

In the case of estimating the mean, μ, of a random variable, Y, i.e.: $\mu = \mathbb{E}(Y)$, one generates n independent and identically distributed (IID) Y_i from the distribution of Y and take the sample average,

$$\hat{\mu}_n := \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

The Strong Law of Large Numbers ensures that the sample mean converges to the true mean almost surely, i.e.: $\lim_{n\to\infty} \hat{\mu}_n = \mu$ a.s. [JP04, Theorem 20.1].

An important special case of computing $\mu = \mathbb{E}(Y)$ is when $Y = f(\boldsymbol{X})$ for some function $f : \mathbb{R}^d \to \mathbb{R}$ and some random vector $\boldsymbol{X}$ with probability density function $\rho : \mathbb{R}^d \to [0, \infty)$. One may then interpret the mean of Y as the multidimensional integral

$$I = \mu(f) = \mathbb{E}(Y) = \mathbb{E}(f(\boldsymbol{X})) = \int_{\mathbb{R}^d} f(\boldsymbol{x})\rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x}.$$

The Monte Carlo method becomes a method for numerical integration.

In this thesis, we mainly focus on IID sampling instead of quasi-Monte Carlo (qMC) sampling, since qMC usually requires the integrand f to have mixed partial derivatives which defines the variation of the integrand f in order to satisfy the specified error condition as suggested in Koksma-Hlawka Inequality [Hic14], whereas

IID sampling does not. In fact, IID sampling does not require $Y = f(\boldsymbol{X})$, The random variable Y can be quite general. We only need user provide the error tolerance and a blackbox to generate the random variables. It is much simpler than the traditional numerical integration approach.

If the random variable Y is a Bernoulli one, which takes only two values, 0 and 1, with $\Pr(Y = 1) = p$. The case of computing the mean becomes estimating the probability $p = \mathbb{E}(Y)$.

This thesis suggests the ways to estimate the means of the random variables mentioned above to a prescribed error tolerance with a high confidence level by determining an appropriate sample size required.

1.2 Guaranteed confidence interval

The Central Limit Theorem (CLT) stated in Theorem 2.2.1 provides a way to construct an *approximate* confidence interval for μ in terms of the sample mean assuming a known variance,

$$\Pr(|\hat{\mu}_n - \mu| \leq \varepsilon) \approx 1 - \alpha.$$

Unfortunately, this is an asymptotic result. The proposed procedure attains the desired coverage level in the limit as $\varepsilon \to 0$ but does not provide coverage guarantees for fixed $\varepsilon > 0$.

Consider a random variable Y with the density function

$$\rho(y) = \frac{1}{\sqrt{2\pi}} \left(0.99 \exp\left(-\frac{x^2}{2} \right) + 0.01 \exp\left(-\frac{(x-200)^2}{2} \right) \right). \tag{1.1}$$

It is a mixture of two Gaussian distributions. Figure 1.1 shows the probability density function. It is obvious that the data are clustered around 0 and it has a heavy tail around 200. The kurtosis $\kappa \approx 98$. The mean of the random variable Y can be written

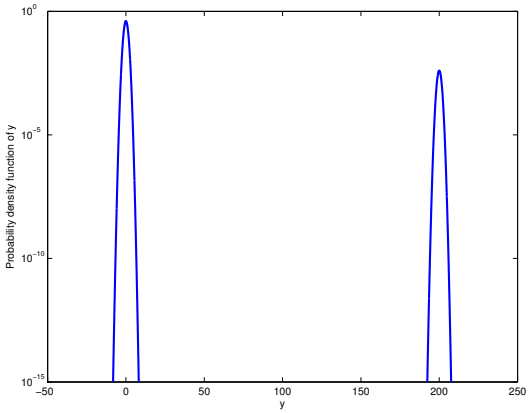

Figure 1.1. The probability density function in (1.1)

as

$$\mathbb{E}(Y) = \int_{-\infty}^{\infty} y\rho(y)dy.$$

If we use CLT to estimate the mean with confidence level 99%, absolute error tolerance 0.01 and initial sample size 100, we get the answer correct 65.6% of the time.

Why does CLT fail? The first reason is that insufficient samples (100) are used to estimate the variance. We do not have a conservative variance estimate for this heavy-tailed distribution. The other reason is, CLT uses an approximate error bound, not a guaranteed error bound.

In this thesis, we investigate ways to estimate the mean of a random variable that satisfies the fixed width confidence interval condition with some prescribed absolute error tolerance ε_a,

$$\Pr(|\mu - \hat{\mu}_n| \leq \varepsilon_a) \geq 1 - \alpha, \tag{1.2}$$

or relative error tolerance ε_r,

$$\Pr\left(|\mu - \hat{\mu}_n| \leq \varepsilon_r\right) \geq 1 - \alpha, \tag{1.3}$$

or some generalized error criterion, $\mathrm{tol}(\varepsilon_a, \varepsilon_r\,|\mu|)$,

$$\Pr\left(|\mu - \tilde{\mu}_n| \leq \mathrm{tol}\left(\varepsilon_a, \varepsilon_r\,|\mu|\right)\right) \geq 1 - \alpha. \tag{1.4}$$

Our assumption is a bounded kurtosis of the random variable Y,

$$\kappa = \frac{\mathbb{E}[(Y-\mu)^4]}{\sigma^4} \leq \kappa_{\max}, \tag{1.5}$$

where σ^2 is the variance of the random variable Y defined in (2.4), and $\kappa_{\max}$ is some bound defined in Definition 2.18.

After reviewing the existing literature, we found the existing theory of fixed width confidence interval may make distributional assumptions that are too strong. Stein [Ste45] proposed a similar two stage procedure to our Algorithm 4.2.1, however, his result requires normally distributed data and has no relative error criterion involved. In Monte Carlo applications one typically does not have much information about the underlying distribution. The form of the distribution for Y is generally not known, var(Y) is generally not known, and Y is not necessarily bounded. We are aiming to derive fixed width confidence intervals that do not require such assumptions.

The width of a confidence interval tends to become smaller as the number n of sampled function values increases. In special circumstances, we can choose n to get a confidence interval of at most the desired length and at least the desired coverage level, $1-\alpha$. For instance, if the variance, $\sigma^2 = \text{var}(Y)$, is known, then an approach based on Chebyshev's Inequality in Theorem 2.3.1 is available, though the actual coverage will usually be much higher than the nominal level, meaning that much narrower intervals would have sufficed. Known variance in addition to a Gaussian distribution for Y supports a fixed width confidence interval construction that is not too conservative [Ste45]. The CLT provides a confidence interval that is asymptotically correct, but our aim is for something that is definitely correct for finite sample sizes. Finally, conservative fixed width confidence intervals for means can be constructed for bounded random variables, by appealing to exponential inequalities such as Hoeffding's [Hoe63] or Chernoff's Inequality [Che52]. We will discuss the way to estimate the mean of Bernoulli random variables by such inequalities.

If the relevant variance or bound is unknown, then approaches based on sequential statistics [Sie85] may be available. In sequential methods one keeps increasing n until the interval is narrow enough. Sequential confidence intervals require us to take account of the stopping rule when computing the confidence level. Unfortunately, all existing sequential methods are lacking in some aspects.

Serfling and Wackerly [SW76] consider sequential confidence intervals for the mean (alternatively for the median) in parametric distributions, symmetric about their center point. The symmetry condition is not suitable for general purpose Monte Carlo applications.

Chow and Robbins[CR65] develop a sequential sampling fixed width confidence interval procedure for the mean, but its guarantees are only asymptotic (as $\varepsilon \to 0$). [MD96] give a procedure similar to Chow and Robbins's, and it has similar drawbacks.

Bayesian methods can support a fixed width interval containing μ with $1-\alpha$ posterior probability, and Bayesian methods famously do not require one to account for stopping rules. They do, however, require strong distributional assumptions.

1.3 Automatic algorithms

Automatic algorithms commonly determine the computational cost required to approximate the solutions that differ from the true solutions by no more than a given error tolerance, ε. In this way, one requires a user specified error tolerance ε and a black box that generate the random variables or function values. Several commonly used software packages have adaptive, automatic algorithms for integrating functions of a single variable. Unfortunately, they do not provide any guarantees. On the other hand, most existing guaranteed automatic algorithms are not adaptive, they do not determine the computational cost needed based on random variables sampled. Some examples of the adaptive automatic algorithms are

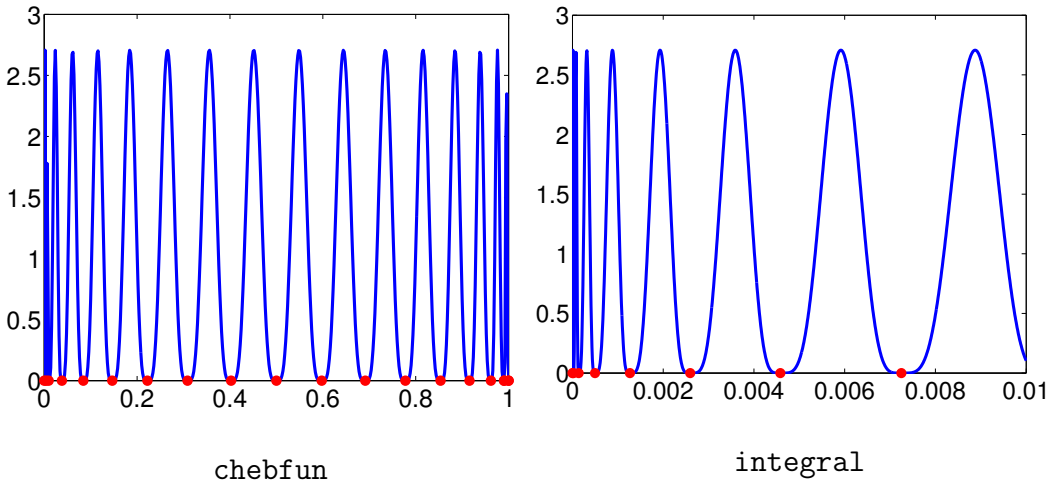

Figure 1.2. Plots of the fooling functions f with $\mu = \int_0^1 f(x)\mathrm{d}x = 1$, for which the corresponding algorithms return values of $\hat{\mu} = 0$. Since there are too many peaks for `integral` between 0 and 1, we decide to plot from 0 to 0.01 to show the basic shape of the function.

- `integral` [Sha08] in MATLAB, which uses the adaptive Gauss-Kronrod quadrature based on `quadva`;
- `chebfun` toolbox [DHT14] for MATLAB, which approximates integrals by integrating interpolatory Chebyshev polynomial approximations to the integrands.

For these two automatic algorithms, one can easily probe where they sample the integrand, feed the algorithms with zero values and then construct fooling functions for which the automatic algorithms will return zero values for the integral. Figure 1.2 displays these fooling functions for problem $\int_0^1 f(x)\mathrm{d}x$ for these two algorithms. Each of these algorithms is given the error tolerance 10^{-14}. Unfortunately, the true absolute error is 1. We are not criticizing that these algorithms can easily be fooled, instead, we have concerns that there is no theory to tell us when the algorithms are fooled or the problem with the integrand.

1.4 Outline of the thesis

The outline of this thesis is as follows: Chapter 2 provides some basic defi-

nitions, inequalities and theorems that support the algorithms. Chapter 3 describes two guaranteed algorithms that estimate the means of random variables and perform numerical integration with some absolute error tolerance. Two similar algorithms with some generalized error tolerance are described in Chapter 4. Chapter 5 explains the ways to estimate the means of Bernoulli random variables with some absolute or relative error tolerance with guarantees. An introduction to Guaranteed Automatic Integration Library (GAIL) is presented in Chapter 6. This thesis ends with a conclusion and some thought about the future work.

CHAPTER 2

BASIC THEOREMS AND INEQUALITIES

In Monte Carlo simulation, one wants to estimate the mean μ of a real valued random variable Y. Usually, this problem could be written as the expectation form $\mu := \mathbb{E}(Y)$. Sometimes, the random variable Y depends on some underlying random vector $\boldsymbol{X} \in \mathbb{R}^d$ with probability density function ρ, where $Y = f(\boldsymbol{X})$. In other situations, the random vector $\boldsymbol{X}$ may have a discrete distribution or may have infinite dimension. In some practical situations, the process governing Y may have a complex form, i.e. the probability of a bankruptcy or the fair price of an option. In these cases, we may be able to generate an IID sample of Y, but may not have a simple formula for computing ρ analytically. In this chapter, we will present some theorems, inequalities and terminologies that may be used to develop our algorithms and prove our theorems in Chapters 3,4 and 5.

2.1 Moments

Let Y be a random variable. Given the sample size n, one generates $Y_1, Y_2, \cdots, Y_n$ samples and calculates the sample mean

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i, \tag{2.1}$$

and the sample variance

$$s_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \hat{\mu}_n)^2. \tag{2.2}$$

The true mean of Y may be written as the expectation form,

$$\mu = \mathbb{E}(Y). \tag{2.3}$$

The true variance is defined as

$$\sigma^2 = \mathbb{E}\left[(Y-\mu)^2\right], \tag{2.4}$$

and the standard deviation is σ. The skewness of Y is given as

$$\gamma = \frac{\mathbb{E}\left[(Y-\mu)^3\right]}{\sigma^3}.$$

Also, define the third centered moment as

$$M_3 = \frac{\mathbb{E}\left[|Y-\mu|^3\right]}{\sigma^3}, \tag{2.5}$$

and the modified kurtosis has the definition,

$$\kappa = \frac{\mathbb{E}\left[(Y-\mu)^4\right]}{\sigma^4}. \tag{2.6}$$

Our main result of this thesis relies on an upper bound on the modified kurtosis, which also implies that the variance and skewness are both finite. See [HJLO14].

2.2 The Central Limit Theorem

The Central Limit Theorem (CLT) describes how the distribution of $\hat{\mu}_n$ approaches a Gaussian distribution as $n \to \infty$.

Theorem 2.2.1 (Central Limit Theorem [JP04, Theorem 21.1])**.** *If $Y_1, \ldots, Y_n$ are IID with $\mathbb{E}(Y_i) = \mu$ and $\operatorname{var}(Y_i) = \sigma^2$, then*

$$\frac{\hat{\mu}_n - \mu}{\sigma/\sqrt{n}} \to \mathcal{N}(0,1) \quad \textit{in distribution, as} \quad n \to \infty.$$

This theorem implies an approximate confidence interval, called a CLT confidence interval, of the form

$$\Pr(|\hat{\mu}_n - \mu| \le \varepsilon) \approx 1 - \alpha$$

for

$$n = N_{\mathrm{CLT}}(\sigma/\varepsilon, \alpha) := \left\lceil z_{\alpha/2}^2 (\sigma/\varepsilon)^2 \right\rceil, \tag{2.7}$$

where $z_{\alpha/2}$ is the $1 - \alpha/2$ quantile of the standard Gaussian distribution. When σ^2 is unknown, it may be replaced by the sample variance s_n^2 defined in (2.2).

2.3 Chebyshev's Inequality

Chebyshev's Inequality may be used to construct a fixed-width confidence interval for μ. It makes relatively mild assumptions on the distribution of the random variable.

Theorem 2.3.1 (Chebyshev's Inequality [LB10, 6.1.c]). *If X is a random variable, for any $\varepsilon > 0$,*

$$\Pr(|X - \mathbb{E}(X)| \geq \varepsilon) \leq \operatorname{var}(X)/\varepsilon^2.$$

Let $Y_1, \ldots, Y_n$ be IID. Choosing $X = \sum_{i=1}^{n} Y_i/n = \mu_n$, noting that $\mathbb{E}(X) = \mathbb{E}(Y) = \mu$, $\operatorname{var}(X) = \operatorname{var}(Y)/n = \sigma^2/n$, and setting $\operatorname{var}(X)/\varepsilon^2 = \sigma^2/(n\varepsilon^2) = \alpha$ leads to the fixed-width confidence interval

$$\Pr(|\hat{\mu}_n - \mu| \leq \varepsilon) \geq 1 - \alpha$$

for

$$n = N_{\text{Cheb}}(\sigma/\varepsilon, \alpha) := \left\lceil \frac{1}{\alpha}(\sigma/\varepsilon)^2 \right\rceil. \tag{2.8}$$

Chebyshev's Inequality ensures that the random variable $\hat{\mu}_n$ is not too far from its true value μ with a fixed width confidence interval. However, this inequality is too conservative and requires a significantly larger sample size than CLT. In some settings, we need a one sided inequality like Chebyshev's. We will appeal to Cantelli's Inequality.

2.4 Cantelli's Inequality

Theorem 2.4.1 (Cantelli's Inequality [LB10, 6.1.e]). *If Y is a random variable with mean μ, and variance σ^2 then for any $a \geq 0$, it follows that:*

$$\Pr(Y - \mu \geq a) \leq \frac{\sigma^2}{a^2 + \sigma^2}. \tag{2.9}$$

Cantelli's Inequality may be used to get a reliable upper bound on the variance σ^2 of a random variable Y in Chapters 3 and 4 for Algorithm 3.1.1, 3.4.1, 4.2.1 and 4.4.1.

2.5 Jensen's Inequality

Theorem 2.5.1 (Jensen's Inequality[LB10, 8.4a])**.** *LetX be a random variable and g be a convex function on* $\mathbb{R}$*. Suppose that the expectations of X and* $g(X)$ *exist, then*

$$g(\mathbb{E}(X)) \leq \mathbb{E}(g(X)).$$

Equality holds for strictly convex g if and only if $X = \mathbb{E}(X)$ *a.s.*

Jensen's Inequality could be used to bound the third moment M_3 defined in (2.5) of a random variable Y by its fourth moment κ from above. This following lemma shows the result.

Lemma 2.5.2. *Let Y be a random variable with mean* μ*, and variance* $\sigma^2 > 0$*, third centered moment* $M_3 < \infty$ *and modified kurtosis* $\kappa < \infty$*. Then the third moment* M_3 *may be bounded as*

$$M_3 \leq \kappa^{3/4}. \tag{2.10}$$

Proof. By Jensen's Inequality in Theorem 2.5.1, let the convex function be $g(y) = y^{4/3}$ and define the random variable $X = |Y - \mu|^3$. Then we have:

$$g(\mathbb{E}(X)) = \left(\mathbb{E}\,|Y-\mu|^3\right)^{4/3} \leq \mathbb{E}\left(\left(|Y-\mu|^3\right)^{4/3}\right) = \mathbb{E}(g(X)).$$

By dividing both sides of the inequality by σ^3, the inequality yields:

$$\frac{\mathbb{E}\left(|Y-\mu|^3\right)}{\sigma^3} \leq \left(\frac{\mathbb{E}\left(|Y-\mu|^4\right)}{\sigma^4}\right)^{3/4}.$$

By appealing the definition of M_3 and κ, the proof is complete. □

This inequality may be used to get the sample size determined by the Berry-Esseen Inequality.

2.6 The Berry-Esseen Inequality

Theorem 2.6.1 (The Berry Esseen Inequality [HJLO14, Theorem 3]). *Let $Y_1, \cdots, Y_n$ be IID random variables with mean μ, variance $\sigma^2 > 0$, and third centered moment $M_3 < \infty$. Let $\hat{\mu}_n = (Y_1 + \cdots + Y_n)/n$ denote the sample mean. Then*

$$\begin{aligned} &\left| \Pr\left(\frac{\hat{\mu} - \mu}{\sigma/\sqrt{n}} < x \right) - \Phi(x) \right| \\ &\leq \delta_n(x, M_3) := \frac{1}{n} \min\left(A_1(M_3 + A_2), A_3(M_3 + A_4), A_5 M_3, \frac{A_6 M_3}{1 + |x|^3} \right) \end{aligned} \tag{2.11}$$

where Φ is the standard Gaussian cumulative distribution function, $A_1 = 0.3322$, $A_2 = 0.429$, $A_3 = 0.3031$, $A_4 = 0.646$, $A_5 = 0.469$ [She14] and $A_6 = 18.1139$ [NS13].

By applying Theorem 2.6.1 with given M_3, let $\hat{\mu}_n$ donate a sample mean of n IID random instances of Y, then the Berry-Esseen Inequality implies that

$$\begin{aligned} &\Pr\left[|\mu - \hat{\mu}_n| \leq \varepsilon\right] \\ &= \Pr\left[\frac{\hat{\mu}_n - \mu}{\sigma/\sqrt{n}} \leq \frac{\sqrt{n}\varepsilon}{\sigma} \right] - \Pr\left[\frac{\hat{\mu}_n - \mu}{\sigma/\sqrt{n}} < -\frac{\sqrt{n}\varepsilon}{\sigma} \right] \\ &\geq \left[\Phi(\sqrt{n}\varepsilon/\sigma) - \delta_n(\sqrt{n}\varepsilon/\sigma, M_3)\right] - \left[\Phi(-\sqrt{n}\varepsilon/\sigma) + \delta_n(-\sqrt{n}\varepsilon/\sigma, M_3)\right] \\ &= 1 - 2[\Phi(-\sqrt{n}\varepsilon/\sigma) + \delta_n(\sqrt{n}\varepsilon/\sigma, M_3)] =: g(n, \sigma, M_3, \varepsilon), \end{aligned} \tag{2.12}$$

since $\delta_n(-x, M_3) = \delta_n(x, M_3)$. The probability of making an error no greater than ε is bounded below by $1 - \alpha$, where α is the uncertainty level, i.e., we want $1 - 2[\Phi(-\sqrt{n}\varepsilon/\sigma) + \delta_n(\sqrt{n}\varepsilon/\sigma, M_3)] =: g(n, \sigma, M_3, \varepsilon) \geq 1 - \alpha$. The fixed width confidence interval (1.2) holds with $\hat{\mu} = \hat{\mu}_n$, provided $n \geq N_{\mathrm{BE}}(\sigma/\varepsilon, \alpha, M_3)$, where the Berry-Esseen sample size is

$$N_{\mathrm{BE}}(\sigma/\varepsilon, \alpha, M_3) := \min\left\{ n \in \mathbb{N} : \Phi\left(-\sqrt{n}/(\sigma/\varepsilon)\right) + \delta_n\left(\sqrt{n}/(\sigma/\varepsilon), M_3\right) \leq \frac{\alpha}{2} \right\}. \tag{2.13}$$

It is interesting to note that the first term $\Phi\left(-\sqrt{n}/(\sigma/\varepsilon)\right)$ is the CLT part, while the second term $\delta_n\left(\sqrt{n}/(\sigma/\varepsilon), M_3\right)$ is the Berry-Esseen extra part. To compute $N_{\mathrm{BE}}(\sigma/\varepsilon, \alpha, M_3)$, we need to know M_3. In practice, substituting an upper bound on M_3 yields an upper bound on the necessary sample size. As $M_3 \leq \kappa^{3/4}$ by Lemma 2.5.2, the sample size calculated by (2.13) is also $N_{\mathrm{BE}}\left(\sigma/\varepsilon, \alpha, \kappa^{3/4}\right)$.

2.7 Hoeffding's Inequality

As mentioned in the introduction, we investigate the way to estimate the mean of Bernoulli random variables, namely, the probability, p. In view of the particular form of the Bernoulli distribution, the inequalities that assume some bound on the random variable may be more feasible. Here we use Hoeffding's Inequality [Hoe63], which seems more suitable than the Chernoff bound [Che52],which is more conservative than Hoeffding's Inequality. Below is a special case of Hoeffding's Inequality for random variables lying in $[0, 1]$.

Theorem 2.7.1 (Hoeffding's Inequality [Hoe63]). *If Y_1, Y_2, $\cdots$, Y_n are IID observations such that $\mathbb{E}(Y_i) = p$ and $0 \leq Y_i \leq 1$, define $\hat{p}_n = \sum_{i=1}^{n} Y_i/n$. Then, for any $\varepsilon > 0$,*

$$\Pr(\hat{p}_n - p \geq \varepsilon) \leq e^{-2n\varepsilon^2}, \tag{2.14}$$

$$\Pr(p - \hat{p}_n \geq \varepsilon) \leq e^{-2n\varepsilon^2}, \tag{2.15}$$

$$\Pr(|\hat{p}_n - p| \geq \varepsilon) \leq 2e^{-2n\varepsilon^2}. \tag{2.16}$$

The reason we use p represent the mean instead of μ is that the Hoeffding's Inequality is used to develop an algorithm that estimate a probability, we feel p may be a better notation.

2.8 Fréchet Formula

Theorem 2.8.1 (Fréchet Formula). *[Fré35] Suppose $A_1, A_2, \cdots, A_n$ are events, the*

probability of a logical conjunction satisfies the following inequality:

$$\max(0, P(A_1) + \cdots + P(A_n) - (n-1)) \leq P\left(\bigcap_{i=1}^{n} A_i\right) \leq \min(P(A_1), \cdots, P(A_n)) \tag{2.17}$$

This inequality bounds the probability of joint events by the probability of the individual events without any dependence assumptions. It will be used to prove success of our algorithms in Chapter 4 and 5.

2.9 Additional definition and lemmas

Definition 2.9.1 (kurtosis maximum)**.** Given the sample size n, uncertainty level α and standard deviation inflation factor (SDIF) $\mathfrak{C}$ (also known as the square root of variance inflation factor (VIF)), define the maximum allowable kurtosis as

$$\kappa_{\max}(\alpha, n, \mathfrak{C}) := \frac{n-3}{n-1} + \left(\frac{\alpha n}{1-\alpha}\right)\left(1 - \frac{1}{\mathfrak{C}^2}\right)^2. \tag{2.18}$$

Definition 2.9.2 (generalized tolerance function)**.** Define a generalized error tolerance function $\mathrm{tol} : [0, \infty) \times [0, \infty) \to [0, \infty)$. Let it be non-decreasing in each of its arguments and satisfy a Lipschitz condition in terms of its second argument, i.e.:

$$|\,\mathrm{tol}(a, b) - \mathrm{tol}(a, b')| \leq |b - b'| \quad \forall a, b, b' \geq 0. \tag{2.19}$$

Two examples that one may choose are

$$\mathrm{tol}(a, b) = \max(a, b),$$

$$\mathrm{tol}(a, b) = (1-\theta)a + \theta b, \quad 0 \leq \theta \leq 1$$

Lemma 2.9.3. *[HJLO14, Lemma 1] Let $Y_1, \cdots, Y_n$ be IID random variables with variance $\sigma^2 > 0$ and modified kurtosis κ. Let s_n^2 be the sample variance defined in*

(2.2). *Then the following inequalities hold:*

$$\Pr\left(s_n^2 < \sigma^2\left(1+\sqrt{\left(\kappa-\frac{n-3}{n-1}\right)\left(\frac{1-\alpha}{\alpha n}\right)}\right)\right) \geq 1-\alpha, \tag{2.20a}$$

$$\Pr\left(s_n^2 > \sigma^2\left(1-\sqrt{\left(\kappa-\frac{n-3}{n-1}\right)\left(\frac{1-\alpha}{\alpha n}\right)}\right)\right) \geq 1-\alpha. \tag{2.20b}$$

Lemma 2.9.4. *Under same assumptions as in Lemma 2.9.3. Given sample size n_σ, uncertainty α_σ and SDIF $\mathfrak{C}$, define an upper bound on the variance as $\hat{\sigma}^2 = \mathfrak{C}^2 s_{n_\sigma}^2$. If the kurtosis of the random variable Y satisfies the condition*

$$\kappa \leq \kappa_{\max} = \kappa_{\max}(n_\sigma, \alpha_\sigma, \mathfrak{C}),$$

then,

$$\Pr(\hat{\sigma}^2 > \sigma^2) \geq 1-\alpha_\sigma.$$

Proof. Starting with inequality (2.20b) in Lemma 2.9.3, since the assumption is $\kappa \leq \kappa_{\max}$, then we have the following inequality:

$$\begin{aligned}
&\Pr\left(s_{n_\sigma}^2 > \sigma^2\left(1-\sqrt{\left(\kappa_{\max}-\frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\alpha_\sigma}{\alpha n_\sigma}\right)}\right)\right)\\
&\geq \Pr\left(s_{n_\sigma}^2 > \sigma^2\left(1-\sqrt{\left(\kappa-\frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\alpha_\sigma}{\alpha_\sigma n_\sigma}\right)}\right)\right)\\
&\geq 1-\alpha_\sigma
\end{aligned}$$

Plugging in $\kappa_{\max}(n_\sigma, \alpha_\sigma, \mathfrak{C})$ defined in (2.18) yields:

$$\begin{aligned}
&Pr\left(s_{n_\sigma}^2 > \sigma^2\left(1-\sqrt{\left(\kappa_{\max}-\frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\alpha_\sigma}{\alpha n_\sigma}\right)}\right)\right)\\
&= \Pr\left(s_{n_\sigma}^2 > \sigma^2\left(1-\left(1-1/\mathfrak{C}^2\right)\right)\right)\\
&= \Pr\left(\mathfrak{C}^2 s_{n_\sigma}^2 > \sigma^2\right)\\
&= \Pr\left(\hat{\sigma}^2 > \sigma^2\right)\\
&\geq 1-\alpha_\sigma
\end{aligned}$$

□

This inequality explains that if the modified kurtosis κ is bounded by $\kappa_{\max}$, then the variance σ^2 will have a probabilistic upper bound $\hat{\sigma}^2$ with confidence level $1-\alpha_\sigma$.

Lemma 2.9.5. *Under same assumptions as in Lemma 2.9.4, define a probabilistic upper bound on $\hat{\sigma}^2$ as:*

$$\hat{\sigma}^2_{\text{up}}(\beta) := \left(\mathfrak{C}^2 + (\mathfrak{C}^2-1)\sqrt{\frac{\alpha_\sigma(1-\beta)}{(1-\alpha_\sigma)\beta}}\right)\sigma^2. \tag{2.21}$$

Then the following inequality must be true:

$$\Pr\left(\hat{\sigma}^2 < \hat{\sigma}^2_{\text{up}}(\beta)\right) \geq 1-\beta. \tag{2.22}$$

Proof. Starting with inequality (2.20a) in Lemma 2.9.3, multiply both sides of the inequality inside the probability by the SDIF $\mathfrak{C}$. Since we assume $\kappa \leq \kappa_{\max}$, replacing the modified kurtosis κ by $\kappa_{\max}$ yields:

$$\begin{aligned}
1-\beta &\leq \Pr\left(s_n^2 < \sigma^2\left(1+\sqrt{\left(\kappa - \frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\beta}{\beta n_\sigma}\right)}\right)\right) \\
&= \Pr\left(\mathfrak{C}^2 s_n^2 < \mathfrak{C}^2\sigma^2\left(1+\sqrt{\left(\kappa - \frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\beta}{\beta n_\sigma}\right)}\right)\right) \\
&\leq \Pr\left(\hat{\sigma}^2 < \mathfrak{C}^2\sigma^2\left(1+\sqrt{\left(\kappa_{\max} - \frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\beta}{\beta n_\sigma}\right)}\right)\right).
\end{aligned}$$

Plugging in the $\kappa_{\max}$ defined in (2.18), and $\hat{\sigma}_{\text{up}}$ defined in (2.21) yields:

$$\begin{aligned}
&1-\beta \\
&\leq \Pr\left(\hat{\sigma}^2 < \mathfrak{C}^2\sigma^2\left(1+\sqrt{\left(\frac{n_\sigma-3}{n_\sigma-1} + \left(\frac{\alpha n_\sigma}{1-\alpha}\right)\left(1-\frac{1}{\mathfrak{C}^2}\right)^2 - \frac{n_\sigma-3}{n_\sigma-1}\right)\left(\frac{1-\beta}{\beta n_\sigma}\right)}\right)\right) \\
&= \Pr\left(\hat{\sigma}^2 < \sigma^2\left(\mathfrak{C}^2 + (\mathfrak{C}^2-1)\sqrt{\frac{\alpha(1-\beta)}{(1-\alpha)\beta}}\right)\right) \\
&= \Pr\left(\hat{\sigma}^2 < \hat{\sigma}^2_{\text{up}}\right)
\end{aligned}$$

□

The following two lemmas will be used to prove the stopping time in Algorithm 4.2.1 is finite.

Lemma 2.9.6 (continuity of probability)**.** *Let $A_1, A_2, \cdots$ be a sequence of events increasing to A. Then*

$$\lim_{n\to\infty} \Pr(A_n) = \Pr(A). \tag{2.23}$$

Proof. Since $A_i \subseteq A_{i+1}$, then we know $A_i \cap A_{i+1} = A_i$. Therefore, define $C_{i+1} = A_{i+1} \cap (A_{i+1} \backslash A_i)$ for $i = 2, 3, \cdots$, and let $C_1 = A_1$. Thus, we have

$$\cup_{i=1}^{n} C_i = A_n$$

$$\cup_{i=1}^{\infty} C_i = \cup_{i=1}^{\infty} A = A$$

As the probability has the property of countably additive, we conclude

$$\Pr(A) = \Pr(\cup_{i=1}^{\infty} C_i) = \sum_{i=1}^{\infty} \Pr(C_i) = \lim_{n\to\infty} \sum_{i=1}^{n} \Pr(C_i) = \lim_{n\to\infty} \Pr(A_n).$$

□

Lemma 2.9.7 (preservation of inequality)**.** *Suppose that $f, g : A \to \mathbb{R}$ and c is a limit point of A. If*

$$f(x) \leq g(x) \text{ for all } x \in A,$$

and $\lim_{x\to c} f(x)$, $\lim_{x\to c} g(x)$ exist, then

$$\lim_{x\to c} f(x) \leq \lim_{x\to c} g(x)$$

Proof. Let

$$\lim_{x\to c} f(x) = L, \lim_{x\to c} g(x) = M.$$

Prove by contradiction, suppose that $L > M$, and let

$$\varepsilon = \frac{1}{2}(L - M) > 0.$$

From the definition of the limit, there exsit $\delta_1, \delta_2 > 0$ such that

$$|f(x) - L| < \varepsilon \qquad \text{if } x \in A \text{ and } 0 < |x - c| < \delta_1,$$

$$|g(x) - M| < \varepsilon \qquad \text{if } x \in A \text{ and } 0 < |x - c| < \delta_2.$$

Let $\delta = \min(\delta_1, \delta_2)$. Since c is a limit point of A, there exists $x \in A$ such that $0 < |x - a| < \delta$, and it follows that

$$f(x) - g(x) = (f(x) - L) + (M - g(x)) + L - M > L - M - 2\varepsilon > 0,$$

which contradicts the assumption, hence the proof proceeds. $\square$

CHAPTER 3

GUARANTEED MONTE CARLO METHOD WITH AN ABSOLUTE ERROR TOLERANCE

In this chapter, we will present two guaranteed automatic adaptive algorithms: `meanMCabs_g`, which estimate the mean μ of a random variable Y; `cubMCabs_g`, which estimate the integral I with the integrand f with respect to some density function ρ within the multivariate interval $(\boldsymbol{a}, \boldsymbol{b})$. In order to estimate the mean μ of a random variable Y, we feed `meanMCabs_g` with a random number generator Yrand and an absolute error tolerance ε_a. This algorithm automatically and adaptively determines the sample size needed in order to guarantee that the answer $\hat{\mu}_n$ satisfy the fixed width confidence interval condition: $\Pr(|\mu - \hat{\mu}_n| \leq \varepsilon_a) \geq 1-\alpha$ if the random variable satisfies the bounded kurtosis condition $\kappa \leq \kappa_{\max}$. Similar for the `cubMCabs_g`, which estimate the integral $I = \int_{\boldsymbol{a}}^{\boldsymbol{b}} f(\boldsymbol{x})\rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, we feed this algorithm with the integrand f, integration interval $(\boldsymbol{a}, \boldsymbol{b})$, the probability density function ρ, and the absolute error tolerance ε_a. We have the guarantee that the answer $\hat{I}_n$ satisfies the fixed width confidence interval condition $\Pr\left(\left|I - \hat{I}_n\right| \leq \varepsilon_a\right) \geq 1 - \alpha$.

3.1 The Algorithm `meanMCabs_g`

Before illustrating the algorithm, we first define some parameters: the uncertainty level α, the sample size n_σ and the SDIF $\mathfrak{C}$, with default values listed in Table 3.1.

Define the sample size calculated by CLT in Theorem 2.2.1 as:

$$N_{\mathrm{CLT}}(\sigma/\varepsilon_a, \alpha) = \left\lceil z_{\alpha/2}^2 (\sigma/\varepsilon_a)^2 \right\rceil. \tag{3.1}$$

Define the sample size calculated by Chebyshev's Inequality in Theorem 2.3.1 as:

$$N_{\mathrm{Cheb}}(\sigma/\varepsilon_a, \alpha) = \left\lceil \frac{1}{\alpha} (\sigma/\varepsilon_a)^2 \right\rceil. \tag{3.2}$$

By applying Theorem 2.6.1, we get the sample size calculated by the Berry-Esseen Inequality as:

$$N_{\mathrm{BE}}\left(\sigma/\varepsilon_a, \alpha, M_3\right) := \min\left\{n \in \mathbb{N} : \Phi\left(-\sqrt{n}/(\sigma/\varepsilon_a)\right) + \delta_n\left(\sqrt{n}/(\sigma/\varepsilon_a), M_3\right) \leq \frac{\alpha}{2}\right\}. \tag{3.3}$$

As our assumption for the bounded kurtosis is $\kappa \leq \kappa_{\max}$, applying (2.10) yields

$$M_3 \leq \kappa_{\max}^{3/4}. \tag{3.4}$$

Thus, we define the sample size obtained by both Chebyshev's and the Berry-Esseen Inequality as:

$$N_{\mathrm{CB}}\left(\sigma/\varepsilon_a, \alpha, \kappa_{\max}^{3/4}\right) = \min\left\{N_{\mathrm{Cheb}}\left(\sigma/\varepsilon_a, \alpha\right), N_{\mathrm{BE}}\left(\sigma/\varepsilon_a, \alpha, \kappa_{\max}^{3/4}\right)\right\}. \tag{3.5}$$

In details, the two-stage algorithm works as follows:

Algorithm 3.1.1 (`meanMCabs_g`). *[HJLO14, Algorithm 1] This algorithm requires the user to provide a random number generator* Yrand *that takes the sample size n as the input and generates n random numbers from the distribution of Y and the absolute error tolerance ε_a.*

Table 3.1. Parameters specified in Algorithm 3.1.1 and 3.4.1.

parameters	description	default
α	the uncertainty level, $0 < \alpha < 1$	1%
α_σ	the uncertainty level for variance estimation, $0 < \alpha_\sigma < \alpha$	$\alpha/2$
n_σ	the sample size for variance estimation, $n_\sigma \in \mathbb{N}$	10^4
$\mathfrak{C}$	standard deviation inflation factor (SDIF), $\mathfrak{C} > 1$	1.2

With the parameters and their default values given in Table 3.1, we first cal-

culate

$$\kappa_{\max} = \kappa_{\max}(n_\sigma, \alpha_\sigma, \mathfrak{C}) = \frac{n_\sigma - 3}{n_\sigma - 1} + \left(\frac{\alpha_\sigma n_\sigma}{1 - \alpha_\sigma}\right)\left(1 - \frac{1}{\mathfrak{C}^2}\right)^2. \tag{3.6}$$

as defined in (2.18)*, and then do the following:*

1. *Compute the sample variance,* $s_{n_\sigma}^2$*, using* n_σ *IID random samples from the distribution of* Y*. Let* $\hat{\sigma}^2 = \mathfrak{C}^2 s_{n_\sigma}^2$ *be the upper bound on variance.*

2. *Compute the uncertainty for the second stage* $\alpha_\mu = 1 - (1 - \alpha)/(1 - \alpha_\sigma)$ *and the sample size for the mean estimation,*

$$n_\mu = N_{\mathrm{CB}}\left(\hat{\sigma}/\varepsilon_a, \alpha_\mu, \kappa_{\max}^{3/4}\right). \tag{3.7}$$

   Sample n_μ *IID samples from the distribution of* Y *that are independent of those used to estimate* σ^2*. Calculate the mean* $\hat{\mu}$*,*

$$\hat{\mu} = \frac{1}{n_\mu} \sum_{i=n_\sigma+1}^{n_\sigma+n_\mu} Y_i. \tag{3.8}$$

   Return $\hat{\mu}$ *as the answer.*

3.1.1 The sensitivity of parameters. We usually fix the uncertainty level α as 1%, which means our answer will be true with probability at least 99%. Decreasing α will increase the sample required according to the formulas (3.1), (3.2) and (3.3). Note that, our algorithm `meanMCabs_g` is a two stage one. We choose the uncertainty levels for the two stages as α_σ and α_μ, which satisfy the condition $(1-\alpha_\sigma)(1-\alpha_\mu) = 1-\alpha$.

The sample size n_σ is usually fixed as 10^4, which is used to estimate the sample variance. As the cost of `meanMCabs_g` is defined as $n_\sigma + n_\mu$, increasing this sample size will increase total sample used while we have a better estimation of the sample variance.

The SDIF $\mathfrak{C}$ is usually set as 1.2, which could be thought as inflating the sample standard deviation by 20% (inflating the sample variance by 44%). This factor ensures that the probability of underestimating the variance is low.

These three parameters, α_σ, n_σ and $\mathfrak{C}$, define $\kappa_{\max}$, which is not a parameter to be prescribed, but a reflection of the robustness of one's Monte Carlo algorithm having chosen α_σ, n_σ and $\mathfrak{C}$.

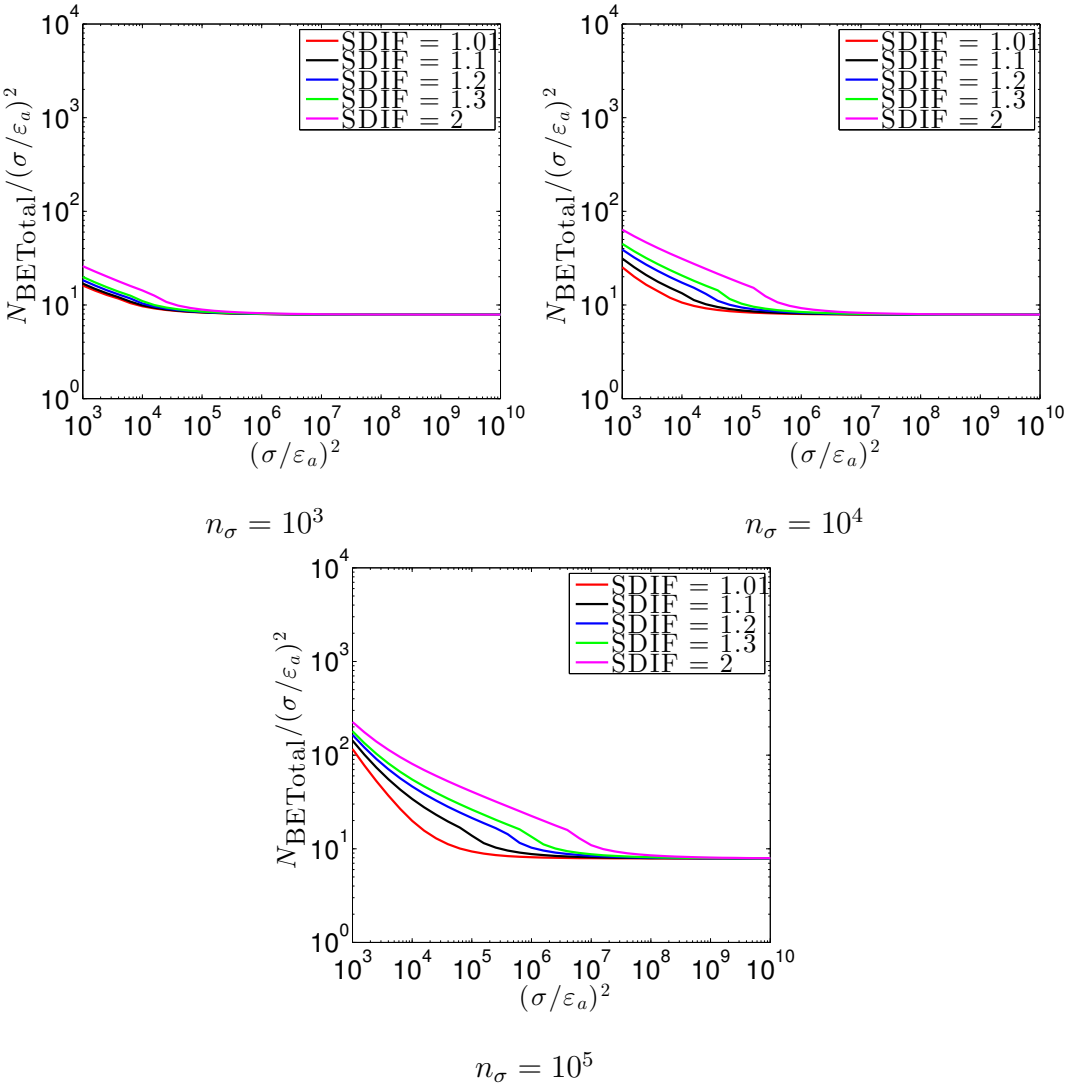

Figure 3.1. The ratio of the total sample size to $(\sigma/\varepsilon_a)^2$ with a fixed n_σ and a range of SDIF $\mathfrak{C}$.

In Figure 3.1, we fix the n_σ as 10^3, 10^4 and 10^5 with $\alpha = 1\%$ and $\alpha_\sigma = 0.5\%$. In each subplot, we choose the different SDIF, $\mathfrak{C}$, as 1.01, 1.1, 1.2, 1.3, and 2, which correspond to kurtosis maximum, $\kappa_{\max}$, of 1.02, 2.52, 5.7, 9.40 and 29.27. Then we plot the ratio of total sample size calculated by the Berry-Esseen inequality in (3.3) to $(\sigma/\varepsilon_a)^2$ ranging from 10^3 to 10^{10}. We expect this ratio to approach a constant.

From the figure we can see that no matter how we choose the initial sample size n_σ, the ratio gets large when $\mathfrak{C}$ gets large. Hence the total sample size gets large. The difference between different $\mathfrak{C}$ is significant when $(\sigma/\varepsilon_a)^2$ is small. With a large $(\sigma/\varepsilon_a)^2$, say, larger than 10^7, the ratios are approaching the same constant. This plot tells us, with a fixed initial sample size and uncertainty level, increasing $\mathfrak{C}$ will increase $\kappa_{\max}$, which could be thought as enlarging the "cone" [CDH+13], while the total sample size will increase also.

In Figure 3.2, we fix $\kappa_{\max}$ and choose a range of $\mathfrak{C}$ and calculate the corresponding n_σ. As in the previous figure, we choose a range of $(\sigma/\varepsilon_a)^2$ and plot the ratio of the total sample size to $(\sigma/\varepsilon_a)^2$. The kurtosis maximum is given as 2, 5, and 10 with $\alpha = 1\%$ and $\alpha_\sigma = 0.5\%$. We choose SDIF $\mathfrak{C}$ as 1.01, 1.1, 1.2, 1.3, and 2. The figure shows that if $\mathfrak{C}$ is 1.01, when $(\sigma/\varepsilon_a)^2$ is small, the ratio gets quite large. The reason is that, for $\kappa_{\max} = 2$, $\mathfrak{C} = 1.01$, $\alpha = 1\%$, the corresponding n_σ is 5×10^5. This term will dominate the total sample size when $(\sigma/\varepsilon_a)^2$ is less than 10^5. If $\mathfrak{C}$ is 2, the ratio is larger than those with SDIF equal to 1.1, 1.2, and 1.3. It is sub-optimal comparing to other choices of $\mathfrak{C}$.

Figure 3.3 shows the ratio of the Berry-Esseen sample size to the CLT sample size with $n_\sigma = 10^4$, $\alpha = 1\%$ and different SDIF $\mathfrak{C}$. Here, N_{CLT} denotes the computational cost if we know the ratio σ/ε_a. The cost ratio is the penalty for having a guaranteed algorithm with the sample size derived from the Berry-Esseen Inequality. For smaller values of N_{CLT} , equivalently smaller σ/ε_a, this cost ratio can be rather large. However the absolute effect of this large penalty is mitigated by the fact that the total number of samples needed is not much. For larger N_{CLT} , equivalently larger σ/ε_a, the cost ratio approaches 1.

Our suggestion is that, user should not skimp on n_σ, but choose n_σ as large as needed to ensure that $\kappa_{\max}$ is is large enough to bound the true kurtosis κ. The

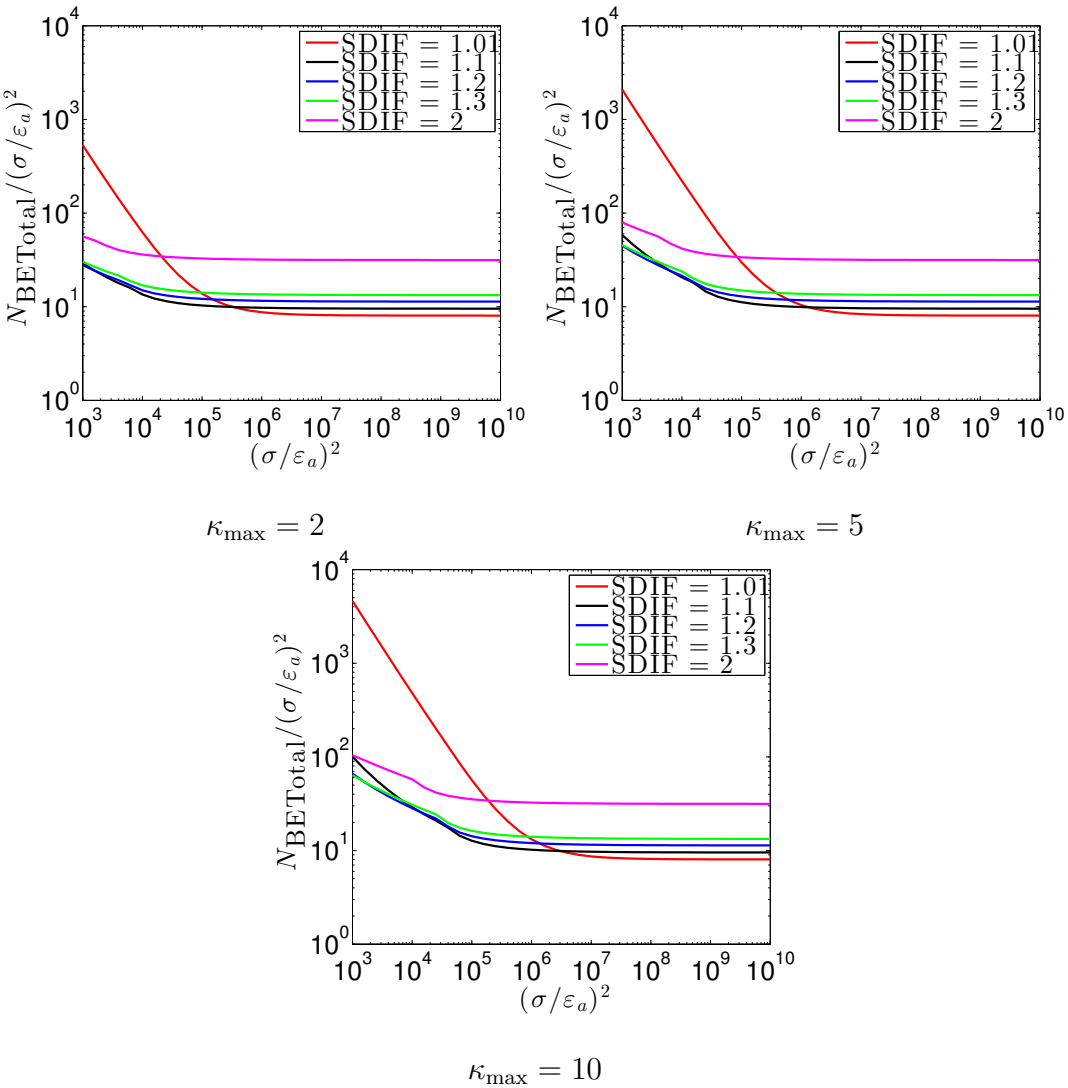

Figure 3.2. The ratio of the total sample size to $(\sigma/\varepsilon)^2$ with a fixed $\kappa_{\max}$ and a range of SDIF $\mathfrak{C}$.

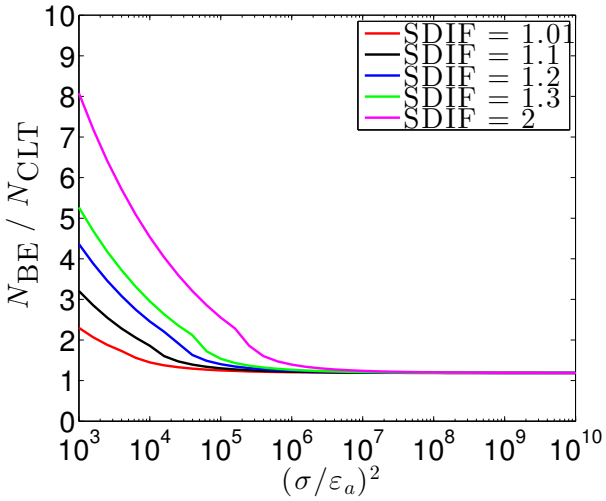

Figure 3.3. The ratio of the Berry-Esseen sample size versus the CLT sample size with different choice of the $\mathfrak{C}$.

SDIF $\mathfrak{C}$ is usually chosen between 1.1 and 1.3. The success of the Algorithm 3.1.1 is proved in the following theorem.

Theorem 3.1.2. *[HJLO14, Theorem 5] Let Y be a random variable with mean μ, variance $\sigma^2 > 0$, and modified kurtosis κ. Let $\kappa_{\max}(n_\sigma, \alpha_\sigma, \mathfrak{C})$ be as defined in* (2.18)*. For any random variable Y with a bounded kurtosis $\kappa \leq \kappa_{\max}(\alpha_\sigma, n_\sigma, \mathfrak{C})$, Algorithm 3.1.1 above yields an estimate $\hat{\mu}$ given by* (3.8) *that satisfies the fixed width confidence interval condition*

$$\Pr\left(|\mu - \hat{\mu}| \leq \varepsilon_a\right) \geq 1 - \alpha. \tag{3.9}$$

Proof. Let ε_a, α_σ, n_σ, $\mathfrak{C}$ be the parameters and $\hat{\sigma}^2$ be a random variable depending on the random samples Y_i, $i = 1, 2, \cdots, n_\sigma$ as given in Algorithm 3.1.1. By Lemma 2.9.4, if the modified kurtosis $\kappa \leq \kappa_{\max}(\alpha_\sigma, n_\sigma, \mathfrak{C})$, then we can find a probabilistic upper bound on σ^2, which is:

$$\Pr\left(\hat{\sigma}^2 > \sigma^2\right) \geq 1 - \alpha_\sigma.$$

Since $\hat{\sigma}$ is random, n_μ as defined in (3.7) is also random. Thus, the randomness of $\hat{\mu}$ comes both from the sample size n_μ and the random variable Y. By adding the extra

condition of the conservative variance estimation inside the probability, we have the inequality below:

$$\begin{aligned}\Pr(|\mu-\hat{\mu}|\le\varepsilon_a) &\ge \Pr(|\mu-\hat{\mu}|\le\varepsilon_a,\hat{\sigma}>\sigma)\\ &= \Pr(|\mu-\hat{\mu}|\le\varepsilon_a|\hat{\sigma}>\sigma)\Pr(\hat{\sigma}>\sigma)\\ &\ge \Pr(|\mu-\hat{\mu}|\le\varepsilon_a|\hat{\sigma}>\sigma)(1-\alpha_\sigma). \end{aligned} \tag{3.10}$$

As the sample n_μ used to calculate $\hat{\mu}$ is defined by Chebyshev's Inequality and the Berry-Esseen Inequality in (3.5), under the assumption that $\kappa\le\kappa_{\max}$ and given g as defined in (2.12), we have:

$$\begin{aligned}&\Pr(|\mu-\hat{\mu}|\le\varepsilon_a|\hat{\sigma}>\sigma)\\ &\ge \Pr\left(\left(\left|\mu-\frac{1}{n_\mu}\sum_{i=1}^{n_\mu}Y_i\right|\le\varepsilon_a, n_\mu=N_{\mathrm{CB}}\left(\hat{\sigma}/\varepsilon_a,\alpha_\mu,\kappa_{\max}^{3/4}\right)\right)\,\middle|\,\hat{\sigma}>\sigma\right)\\ &\ge g\left(n_\mu,\hat{\sigma},\kappa_{\max}^{3/4},\varepsilon_a\right)\\ &\ge (1-\alpha_\mu). \end{aligned} \tag{3.11}$$

Combining (3.10) and (3.11) completes the proof. □

3.2 An upper bound on cost of the Algorithm `meanMCabs_g`

The cost of the Algorithm 3.1.1 is determined by two parts: the sample size n_σ used to estimate the variance σ in stage one, and the sample size n_μ used to estimate the mean μ in stage two. Thus, the total cost is

$$n_{\mathrm{tot}}=n_\sigma+n_\mu.$$

Although the n_σ is deterministic, n_μ is a random variable. The cost of Algorithm 3.1.1 may be defined probabilistically. Here we have a theorem that bounds the cost of the Algorithm 3.1.1 from above.

Theorem 3.2.1. *The two stage Monte Carlo algorithm for fixed width confidence intervals based on IID sampling described in Algorithm 3.1.1 has a probabilistic cost bound* $n_{\text{up}}(\beta) = n_\sigma + \bar{n}_\mu(\beta)$*, where* $\bar{n}_\mu(\beta) = N_{\text{CB}}\left(\hat{\sigma}_{\text{up}}(\beta)/\varepsilon_a, \alpha_\mu, \kappa_{\max}^{3/4}\right)$.

$$\Pr\left(n_{\text{tot}} \leq n_{\text{up}}(\beta)\right) \geq 1 - \beta. \tag{3.12}$$

Proof. As the only random quantity in n_μ is $\hat{\sigma}^2$, we need to bound $\hat{\sigma}^2$ so as to bound n_μ. By applying the definition of $\hat{\sigma}^2_{\text{up}}(\beta)$ in Lemma 2.9.5, we have:

$$\begin{aligned}\Pr(n_\mu \leq \bar{n}_\mu(\beta)) &= \Pr\left(N_{\text{CB}}\left(\hat{\sigma}/\varepsilon_a, \alpha_\mu, \kappa_{\max}^{3/4}\right) \leq N_{\text{CB}}\left(\hat{\sigma}_{\text{up}}(\beta)/\varepsilon_a, \alpha_\mu, \kappa_{\max}^{3/4}\right)\right) \\ &\geq \Pr\left(\hat{\sigma}^2 < \hat{\sigma}^2_{\text{up}}(\beta)\right) \geq 1 - \beta\end{aligned}$$

Adding the constant n_σ inside the probability we have:

$$\Pr(n_\sigma + n_\mu \leq n_\sigma + \bar{n}_\mu(\beta)) \geq 1 - \beta.$$

Hence the proof is complete. □

3.3 Numerical integration via Monte Carlo sampling

An important special case of computing $I = \mathbb{E}(Y)$ is when $Y = f(\boldsymbol{x})$ for some function $f : \mathbb{R}^d \to \mathbb{R}$ and some random vector $\boldsymbol{x}$ with probability density function $\rho : \mathbb{R}^d \to [0, \infty)$. One may then interpret the mean of Y as the multidimensional integral

$$I = \mu(f) = \mathbb{E}(Y) = \int_{\mathbb{R}^d} f(\boldsymbol{x})\rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{3.13}$$

Note that given the problem of evaluating $I = \int_{\mathbb{R}^d} g(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, one must choose a probability density function ρ for which one can easily generate random vectors $\boldsymbol{x}$, then set $f = g/\rho$. The quantities σ^2 and κ defined above can be written in terms of weighted $\mathcal{L}_p$ norms of f:

$$\|f\|_p := \left\{\int_{\mathbb{R}^d} |f(\boldsymbol{x})|^p \rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right\}^{1/p}, \quad \sigma^2 = \|f - \mu\|_2^2, \quad \kappa = \frac{\|f - \mu\|_4^4}{\|f - \mu\|_2^4}.$$

For a given g, the choice of ρ is not unique, and making an optimal choice belongs to the realm of the importance sampling. The assumption of bounded kurtosis in (1.5) required by Algorithm 3.1.1 corresponds to an assumption that integrand f lies in the cone of functions [CDH$^+$13]

$$C_{\kappa_{\max}} = \left\{ f \in \mathcal{L}_4 : \|f - \mu(f)\|_4 \leq \kappa_{\max}^{1/4} \|f - \mu(f)\|_2 \right\}.$$

This is in contrast to a ball of functions, which wound be the case if one was satisfying a bounded variance condition. Below, we will present our Algorithm `cubMCabs_g` which estimates the integral (3.13) as $\hat{I}$ in order to satisfy the fixed width confidence interval condition

$$\Pr\left(\left|\hat{I} - I\right| \leq \varepsilon_a\right) \geq 1 - \alpha. \tag{3.14}$$

3.4 The Algorithm `cubMCabs_g`

Algorithm 3.4.1. *This algorithm requires the user to provide an integrand f, an integration interval $(\boldsymbol{a}, \boldsymbol{b})$, a measure ρ, and an absolute error tolerance ε_a. With the parameters with the default values specified in Table 3.1, we first calculate $\kappa_{\max} = \kappa_{\max}(n_\sigma, \alpha_\sigma, \mathfrak{C})$ as defined in* (2.18)*, and then do the following:*

1. *In the first stage, sample n_σ values of random vector $\boldsymbol{X}_i$ from the distribution of $\boldsymbol{X}$ with probability density function ρ and obtain n_σ function values of $f(\boldsymbol{X}_i)$. Use this to calculate the sample mean*

$$\hat{I}_{n_\sigma} = \frac{1}{n_\sigma} \sum_{i=1}^{n_\sigma} f\left(\boldsymbol{X}_i\right),$$

and sample variance

$$s_{n_\sigma}^2 = \frac{1}{n_\sigma - 1} \sum_{i=1}^{n_\sigma} \left(f(\boldsymbol{X}_i) - \hat{I}_{n_\sigma}\right)^2.$$

Also approximate the upper bound on variance by $\hat{\sigma}^2 = \mathfrak{C}^2 s_{n_\sigma}^2$.

2. *After the preparation of the upper bound on variance, compute the uncertainty level for the second stage* $\alpha_\mu = 1-(1-\alpha)/(1-\alpha_\sigma)$ *and the sample size for the mean estimation,*

$$n_\mu = N_{\mathrm{CB}}\left(\hat{\sigma}/\varepsilon_a, \alpha_\mu, \kappa_{\max}^{3/4}\right). \tag{3.15}$$

Sample n_μ *values of random vector* $\boldsymbol{X}_i$ *that are independent of those used to calculate* $\hat{\sigma}^2$ *from the distribution of* $\boldsymbol{X}$ *with probability density function* ρ*, then calculate the* n_μ *function values of* $f(\boldsymbol{X}_i)$*. Use this to calculate the sample mean*

$$\hat{I} = \frac{1}{n_\mu}\sum_{i=n_\sigma+1}^{n_\sigma+n_\mu} f(\boldsymbol{X}_i). \tag{3.16}$$

As `cubMCabs_g` uses the same argument as `meanMCabs_g`, just treat the function values $f(\boldsymbol{X})$ as the random numbers Y and estimate the mean. Hence, the proof of the success and the cost of the algorithm will be the same as in Theorem 3.1.2 and Theorem 3.2.1. The only difference is that, it requires users to input the integrand f and integral hyperbox $(\boldsymbol{a}, \boldsymbol{b})$ instead of the random number generator Yrand, which may be easier for user who is just seeking the result of a integral or wants to compare different quadrature methods. In the next section, we will show some numerical examples using `cubMCabs_g` to solve numerical integration problems.

3.5 Numerical examples for `cubMCabs_g`

In this section, we will show some numerical examples to integrate some test functions using different algorithms including `cubMCabs_g`, `cubSobol_g` [HJ14], `cubLattice_g` [JH14], `integral` [Sha08] and `chebfun` [DHT14].

3.5.1 Integrating a product function. A product function is used here to perform an integration problem $\mu = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, where the integrand is chosen as

$$f(\boldsymbol{x}) = \prod_{i=1}^{d}(x_i^2 + a_i), \tag{3.17}$$

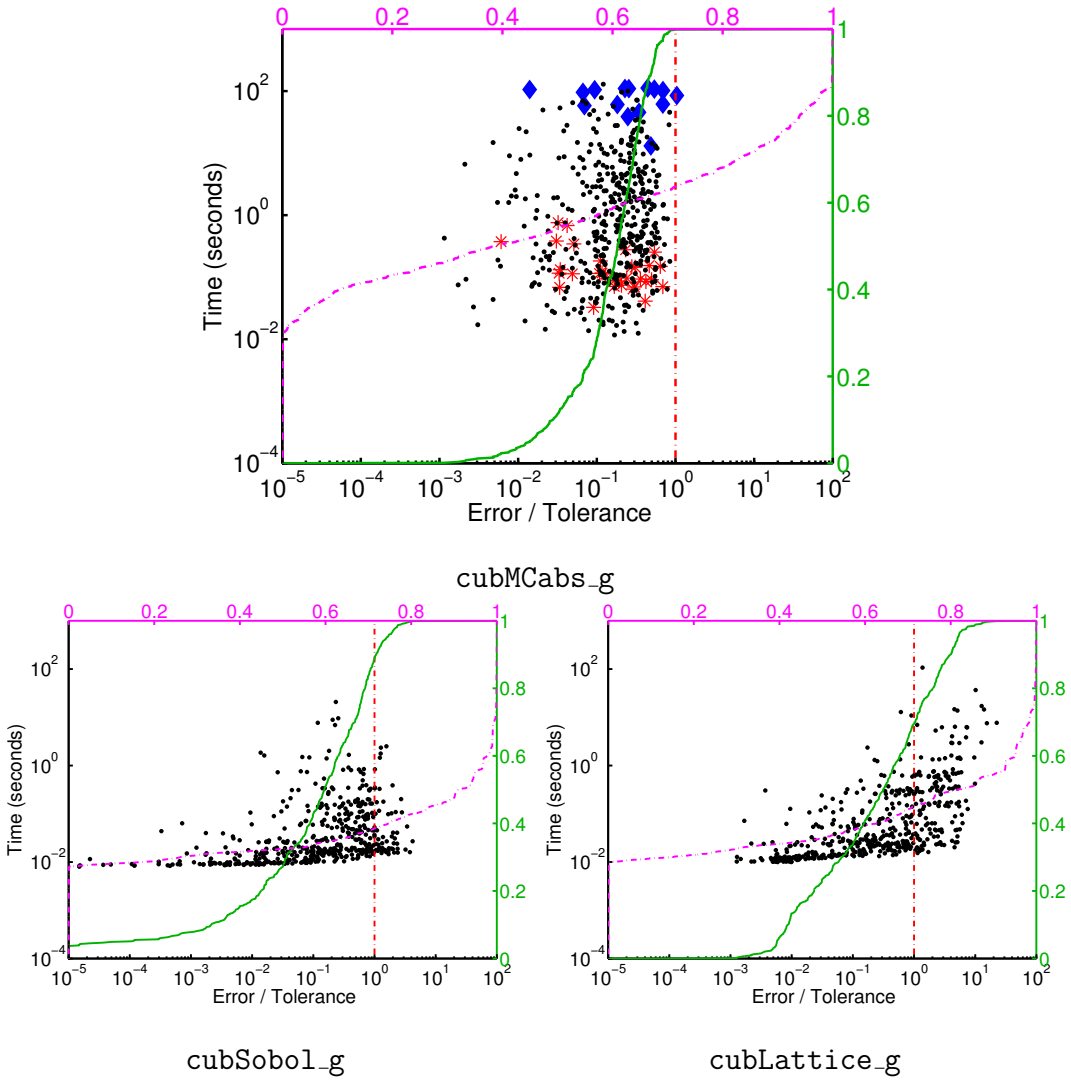

Figure 3.4. Execution times and errors for the product function (3.17) with d random chosen between 2 and 20, and the absolute error tolerance $\varepsilon_a = 10^{-3}$. Those points to the left/right of the dashed vertical line represent successes/failures of the automatic algorithms. The solid green curve shows that cumulative distribution of actual errors, and the dot-dashed curve shows the cumulative distribution of execution times.

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$, d is IID uniform on $[2, 20]$, a_i is IID uniform in $[0, 4/3]$, for $i = 1, 2, \cdots, d$. Accuracy and time are recorded in Figure 3.4 with 500 replications performed using algorithms `cubMCabs_g`, `cubSobol_g` [HJ14] and `cubLattice_g` [JH14]. For all of these three algorithms, the absolute error tolerance is 10^{-3}, the initial sample size is 2^{13}, and the sample budget is 10^{10}. With the uncertainty level $\alpha = 0.01$ and $\mathfrak{C} = 1.2$, the kurtosis maximum is $\kappa_{\max} = 4.84$ for `cubMCabs_g`.

In the plots for `cubMCabs_g`, an asterisk is used to label those points satisfying $\kappa \leq \kappa_{\max}$, where κ is defined in (2.6). All such points fall within the prescribed error tolerance, which is even better than the guaranteed confidence of 99%. Those points labeled with a dot, are those for which $\kappa > \kappa_{\max}$, and so no guarantee holds. The points labeled with a diamond are those for which `cubMCabs_g` attempts to exceed the cost budget that we set, i.e., it wants to choose n_μ such that $n_\sigma + n_\mu > n_{\max} := 10^{10}$. In these cases n_μ is chosen as $\lfloor 10^{10} - n_\sigma \rfloor$, which often is still large enough to get an answer that satisfies the error tolerance.

For `cubSobol_g` there are 444 out of 500 samples that reach the desired error tolerance, which is, the probability of success is 88.8%. Similarly, for `cubLattice_g`, there are 347 out of 500 samples that have the error less than the tolerance, i.e. the probability of success is 69.4%. Both of these quasi Monte Carlo methods have not reach the desired confidence level, 99%.

3.5.2 Integrating a single hump. Accuracy and timing results have been recorded for the integration problem $\mu = \int_{[0,1]^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$ for a single hump test integrand

$$f(\boldsymbol{x}) = a_0 + b_0 \prod_{j=1}^{d} \left[1 + b_j \exp\left(-\frac{(x_j - h_j)^2}{c_j^2} \right) \right]. \tag{3.18}$$

Here $\boldsymbol{x}$ is a d dimensional vector, and $a_0, b_0, \ldots, b_d, c_1, \ldots, c_d, h_1, \ldots, h_d$ are parameters. Figure 3.5 shows the results of different algorithms being used to integrate 500 different instances of f. For each instance of f, the parameters are chosen as follows:

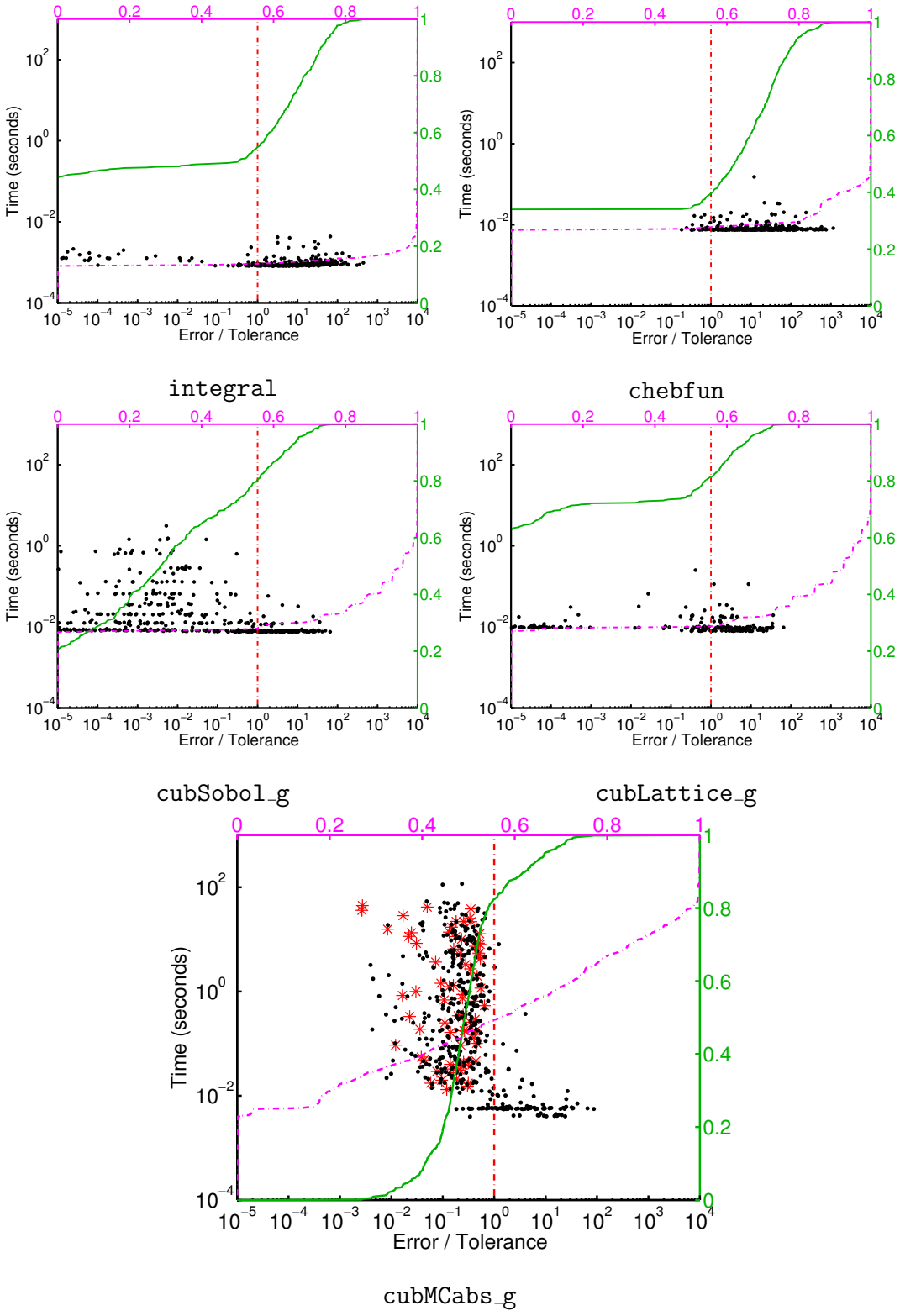

Figure 3.5. Execution times and errors for the single hump test function (3.18) for $d = 1$ and absolute error tolerance $\varepsilon_a = 10^{-3}$.

- $b_1, \ldots, b_d \in [0.1, 10]$ with $\log(b_j)$ being IID uniform,
- $c_1, \ldots, c_d \in [10^{-6}, 1]$ with $\log(c_j)$ being IID uniform,
- $h_1, \ldots, h_d \in [0, 1]$ with h_j being IID uniform,
- b_0 chosen in terms of the $b_1, \ldots, b_d, c_1, \ldots, c_d, h_1, \ldots, h_d$ to make $\sigma^2 = \|f - \mu\|_2^2 \in [10^{-2}, 10^2]$, with $\log(\sigma)$ being IID uniform for each instance, and
- a_0 chosen in terms of the $b_0, \ldots, b_d, c_1, \ldots, c_d, h_1, \ldots, h_d$ to make $\mu = 1$.

These ranges of parameters are chosen so that the algorithms being tested fail to meet the error tolerance a significant number of times.

These 500 random constructions of f with $d = 1$ are integrated using `integral` [Sha08], `chebfun` [DHT14], `cubSobol_g`, `cubLattice_g`, and `cubMCabs_g`. For all but `chebfun`, the specified absolute error tolerance is $\varepsilon_a = 10^{-3}$. The algorithm `chebfun` attempts to do all calculations to near machine precision but we still measure the success relative to absolute error tolerance $\varepsilon_a = 10^{-3}$. The initial sample size is set as 2^{13} for all of the three Monte Carlo algorithms. `cubMCabs_g` takes $\alpha = 0.01$ and $\mathfrak{C} = 1.2$, which corresponds to $\kappa_{\max} = 4.84$.The observed error and execution times are plotted in Figure 3.5.

Figure 3.5 shows that `integral` is quite fast, nearly always providing an answer in less than 0.01 seconds. Unfortunately, it successfully meets the error tolerance only about 55% of the time, which is, only 275 out of 500 samples meet the error criterion. The performance of `chebfun` is even worse than `integral`, only 199 out of 500 samples meet the error tolerance, thus, the accuracy is 39.8%. For `chebfun` plots, there are a significant proportion of the data that do not appear because their errors are smaller than 10^{-5}. The difficult cases are those where c_1 is quite small, and these algorithms miss the sharp peak.

The two quasi-Monte Carlo methods perform better than `integral` and `chebfun`. There are 402 out of 500 samples satisfy the error criterion for `cubSobol_g`, which provides a success rate of 80.4%. On the other hand, for `cubLattice_g`, 407 out of 500 samples attain desired accuracy, which gives a success rate 81.4%.

In the plots for `cubMCabs_g`, an asterisk is used to label those points satisfying $\kappa \leq \kappa_{\max}$, where κ is defined in (2.6). All such points fall within the prescribed error tolerance, i.e. the accuracy is 100%, which is even better than the guaranteed confidence of 99%. Those points labeled with a dot, are those for which $\kappa > \kappa_{\max}$, and so no guarantee holds. For `cubMCabs_g` the accuracy is 82.4%.

The algorithm `cubMCabs_g` performs somewhat more robustly than all other algorithms, because it does not assume any smoothness of the integrand. On the other hand, `cubMCabs_g` is generally much slower than the other algorithms. The more important point is that given the kurtosis of the random variable less than the kurtosis maximum, `cubMCabs_g` has a guarantee.

Figure 3.6 repeats the simulation shown in Figure 3.5 for the same test function (3.18) with $d = 2, \ldots, 8$ chosen randomly and uniformly. For this case the univariate integration algorithms are inapplicable, but the multidimensional routines can be used. There are more cases where the `cubMCabs_g` tries to exceed the maximum sample size allowed, i.e., $(n_\sigma + n_\mu)d > N_{\max} := 10^{10}$, but the behavior seen for $d = 1$ still generally applies.

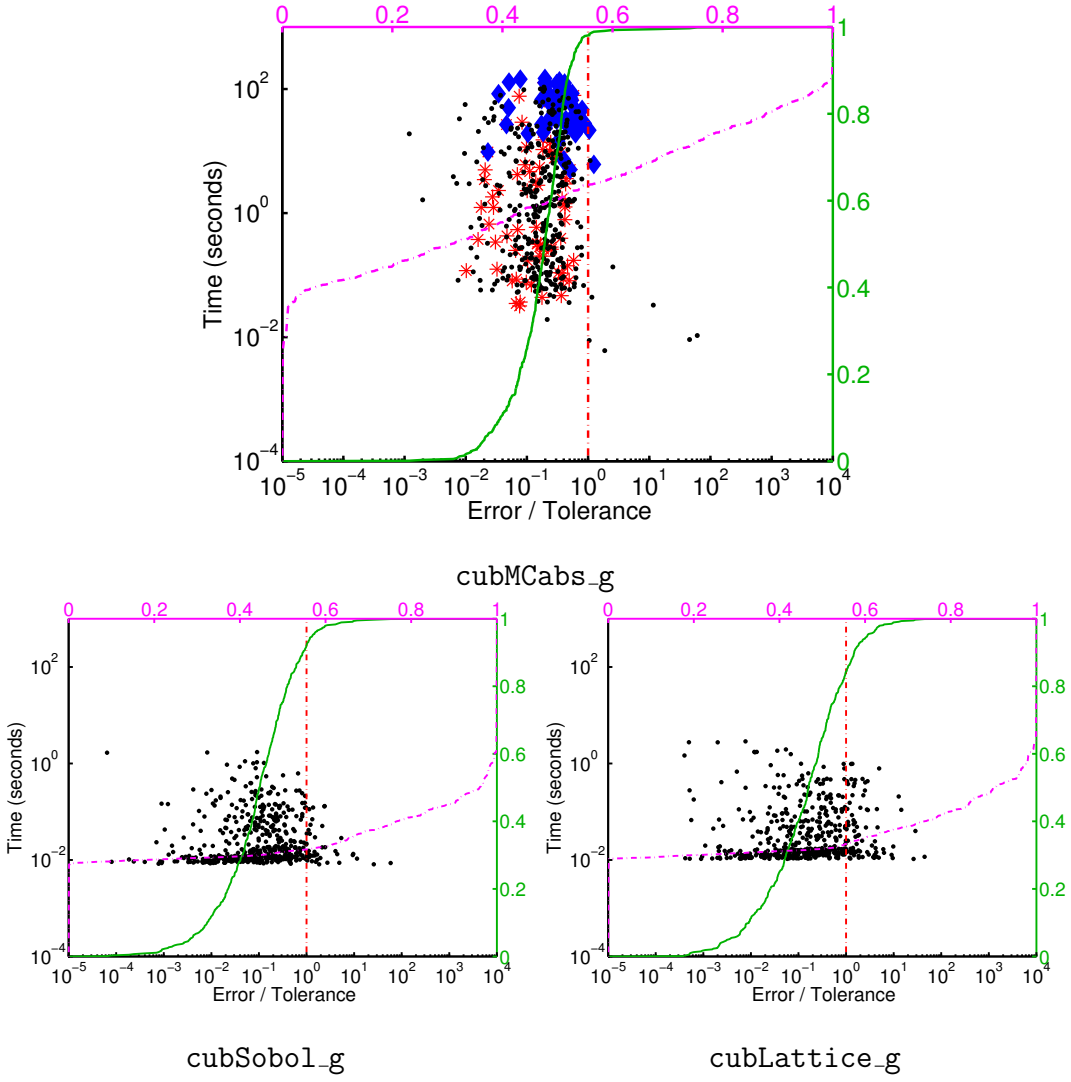

Figure 3.6. Execution times and errors for the single hump test function (3.18) for $d = 2, \ldots, 8$ and $\varepsilon_a = 10^{-3}$, with the rest of the parameters as in Figure 3.5.

CHAPTER 4

GUARANTEED MONTE CARLO METHOD WITH A GENERALIZED ERROR TOLERANCE

In this chapter, we will illustrate another two algorithms, `meanMC_g` and `cubMC_g`. They are used to estimate the mean μ of a random variable Y and perform numerical integration under some generalized error tolerance criterion. This is a hybrid error criterion that combines the absolute error tolerance ε_a and relative error tolerance ε_r. For example, we can take the tolerance $\max(\varepsilon_a, \varepsilon_r |\mu|)$. These two algorithms do not default to the previous two algorithms described in Chapter 3 if the absolute error tolerance is quite small but nonzero. The algorithms introduced in this chapter will proceed iteratively until the stopping criterion is met. The details of the algorithms are described in Section 4.2 and Section 4.4. The computational cost bound is derived in Section 4.3 and several numerical examples are given in Section 4.5.

4.1 A generalized error criterion

Suppose Y is a random variable, one wants to estimate its mean, i.e. $\mu = \mathbb{E}(Y)$. One way to do it is to sample n values of Y_i, and let $\hat{\mu} = \sum_{i=1}^{n} Y_i/n$, when n is large enough, $\hat{\mu}$ may be a good estimator of μ. In some practical situations, one may seek to approximate the answer with a generalized error criterion of the form:

$$|\mu - \tilde{\mu}| \leq \mathrm{tol}\left(\varepsilon_a, \varepsilon_r |\mu|\right), \tag{4.1}$$

where tol is defined in Definition 2.9.2. Here $\tilde{\mu}$ might not be the same as $\hat{\mu}$, but is defined in terms of $\hat{\mu}$. Suppose that one has a reliable upper bound on the error of a non-adaptive Monte Carlo estimate $\hat{\mu}$, which satisfies the following condition:

$$|\mu - \hat{\mu}| \leq \hat{\varepsilon}. \tag{4.2}$$

It is equivalent to

$$\hat{\mu} - \hat{\varepsilon} \leq \mu \leq \hat{\mu} + \hat{\varepsilon}. \tag{4.3}$$

That mean the true mean μ lies in the interval

$$[\hat{\mu} - \hat{\varepsilon}, \hat{\mu} + \hat{\varepsilon}]. \tag{4.4}$$

If a relative error criterion is required, we insist that this interval (4.4) not contain zero. Otherwise, the relative error criterion may never be satisfied. That is same as requiring $\hat{\mu} + \hat{\varepsilon} < 0$ or $\hat{\mu} - \hat{\varepsilon} > 0$, which is equivalent to

$$|\hat{\mu}| > \hat{\varepsilon}. \tag{4.5}$$

Suppose there is another estimator $\tilde{\mu}$ that satisfies the error condition

$$|\mu - \tilde{\mu}| \leq \tilde{\varepsilon} \, |\mu| \, , \tag{4.6}$$

where $\tilde{\varepsilon}$ is some relative error tolerance less than 1. It is equivalent to

$$\mu - \tilde{\varepsilon} \, |\mu| \leq \tilde{\mu} \leq \mu + \tilde{\varepsilon} \, |\mu| \, . \tag{4.7}$$

Since the functions $f(x) = x - \tilde{\varepsilon} \, |x|$, $g(x) = x + \tilde{\varepsilon} \, |x|$, $0 < \tilde{\varepsilon} < 1$ are non-decreasing, by inequality (4.3), we know

$$\mu - \tilde{\varepsilon} \, |\mu| \leq \hat{\mu} + \hat{\varepsilon} - \tilde{\varepsilon} \, |\hat{\mu} + \hat{\varepsilon}|$$

and

$$\hat{\mu} - \hat{\varepsilon} + \tilde{\varepsilon} \, |\hat{\mu} - \hat{\varepsilon}| \leq \mu + \tilde{\varepsilon} \, |\mu| \, .$$

If the following condition

$$\hat{\mu} + \hat{\varepsilon} - \tilde{\varepsilon} \, |\hat{\mu} + \hat{\varepsilon}| \leq \hat{\mu} - \hat{\varepsilon} + \tilde{\varepsilon} \, |\hat{\mu} - \hat{\varepsilon}| \Leftrightarrow \hat{\varepsilon} \leq \tilde{\varepsilon}(|\hat{\mu} + \hat{\varepsilon}| + |\hat{\mu} - \hat{\varepsilon}|)/2 \Leftrightarrow \hat{\varepsilon} \leq \tilde{\varepsilon} \, |\hat{\mu}| \tag{4.8}$$

is satisfied, we will be able to find a $\tilde{\mu}$ in the interval $[\hat{\mu} + \hat{\varepsilon} - \tilde{\varepsilon} \, |\hat{\mu} + \hat{\varepsilon}| \, , \hat{\mu} - \hat{\varepsilon} + \tilde{\varepsilon} \, |\hat{\mu} - \hat{\varepsilon}|]$ that satisfies (4.6). One option is to pick $\tilde{\mu}$ as the middle point, which is

$$\tilde{\mu} = \hat{\mu} + \tilde{\varepsilon}(|\hat{\mu} - \hat{\varepsilon}| - |\hat{\mu} + \hat{\varepsilon}|)/2.$$

Note that, the estimator $\tilde{\mu}$ is a biased one. It is smaller than $|\hat{\mu}|$ in magnitude if $\tilde{\varepsilon} > 0$. We call the part $\tilde{\varepsilon}(|\hat{\mu} - \hat{\varepsilon}| - |\hat{\mu} + \hat{\varepsilon}|)/2$ as Δ_- in the following proposition,

which could be thought as a shift toward zero for the unbiased estimator $\hat{\mu}$. Then if the estimator $\hat{\mu}$ satisfies the condition (4.2), the shirnkage estimator $\tilde{\mu}$ will satisfy condition (4.6). Motivated by the discussion above, we have a proposition below.

Proposition 4.1.1. *Given an absolute error tolerance, $\varepsilon_a \geq 0$, a relative error tolerance, $0 \leq \varepsilon_r \leq 1$, and some $\hat{\mu}$ and $\hat{\varepsilon} \geq 0$, define $\Delta_{\pm}$ by*

$$\Delta_{\pm}(\hat{\mu}, \hat{\varepsilon}) = \frac{1}{2}[\mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} - \hat{\varepsilon}|) \pm \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} + \hat{\varepsilon}|)] \tag{4.9}$$

and the approximation $\tilde{\mu}$ by

$$\tilde{\mu} = \hat{\mu} + \Delta_{-}. \tag{4.10}$$

If error bound (4.2) *holds, and $\hat{\varepsilon}$ satisfies the following condition (stopping criterion),*

$$\hat{\varepsilon} \leq \Delta_{+}(\hat{\mu}, \hat{\varepsilon}), \tag{4.11}$$

then the generalized error criterion for $\tilde{\mu}$, (4.1)*, is satisfied.*

Proof. Suppose $\Delta_{+} \geq \hat{\varepsilon}$, which is

$$\frac{1}{2}[\mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} - \hat{\varepsilon}|) + \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} + \hat{\varepsilon}|)] > \hat{\varepsilon}.$$

This is equivalent to

$$2\hat{\varepsilon} - \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} + \hat{\varepsilon}|) \leq \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} - \hat{\varepsilon}|) \tag{4.12}$$

$$-\mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} + \hat{\varepsilon}|) \leq \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} - \hat{\varepsilon}|) - 2\hat{\varepsilon}. \tag{4.13}$$

Combining inequalities (4.10) (4.12), and (4.13) yields

$$\begin{aligned}\hat{\mu} + \hat{\varepsilon} - \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} + \hat{\varepsilon}|) \leq \tilde{\mu} = \hat{\mu} + \frac{1}{2}[\mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} - \hat{\varepsilon}|) - \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} + \hat{\varepsilon}|)] \\ \leq \hat{\mu} - \hat{\varepsilon} + \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} - \hat{\varepsilon}|)\end{aligned} \tag{4.14}$$

According to the Definition 2.9.2 of the function $\mathrm{tol}(\cdot,\cdot)$, which is non-decreasing in each of its argument and is Lipschitz continuous in its second argument, we know that $f(x) = x \pm \mathrm{tol}(\varepsilon_a, \varepsilon_r |x|)$ is non-decreasing. By (4.3) we have

$$\mu - \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|) \leq \hat{\mu} + \hat{\varepsilon} - \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} + \hat{\varepsilon}|) \tag{4.15}$$

$$\hat{\mu} - \hat{\varepsilon} + \mathrm{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} - \hat{\varepsilon}|) \leq \mu + \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|). \tag{4.16}$$

Combining inequalities (4.14), (4.15) and (4.16) yields:

$$\mu - \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|) \leq \tilde{\mu} \leq \mu + \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|). \tag{4.17}$$

The result (4.1) is obtained and the proof finished. □

Table 4.1. The scenarios with different $\hat{\varepsilon}$ and $\hat{\mu}$

	Condition	Parameters	Stopping criterion
1	$0 < \hat{\varepsilon} < \hat{\mu}$	$\Delta_+ = \varepsilon_r \lvert\hat{\mu}\rvert$, $\Delta_- = -\varepsilon_r\hat{\varepsilon}$, $\tilde{\mu} = \hat{\mu} - \varepsilon_r\hat{\varepsilon}$	$\hat{\varepsilon} \leq \varepsilon_r \lvert\hat{\mu}\rvert$
2	$\hat{\mu} < -\hat{\varepsilon} < 0 < \hat{\varepsilon}$	$\Delta_+ = \varepsilon_r \lvert\hat{\mu}\rvert$, $\Delta_- = \varepsilon_r\hat{\varepsilon}$, $\tilde{\mu} = \hat{\mu} + \varepsilon_r\hat{\varepsilon}$	$\hat{\varepsilon} \leq \varepsilon_r \lvert\hat{\mu}\rvert$
3	$0 < \hat{\mu} < \hat{\varepsilon}$	$\Delta_+ = \varepsilon_r\hat{\varepsilon}$, $\Delta_- = -\varepsilon_r\hat{\mu}$, $\tilde{\mu} = \hat{\mu} - \varepsilon_r\hat{\mu}$	$\hat{\varepsilon} \leq \varepsilon_r\hat{\varepsilon}$
4	$0 < -\hat{\mu} < \hat{\varepsilon}$	$\Delta_+ = \varepsilon_r\hat{\varepsilon}$, $\Delta_- = -\varepsilon_r\hat{\mu}$, $\tilde{\mu} = \hat{\mu} - \varepsilon_r\hat{\mu}$	$\hat{\varepsilon} \leq \varepsilon_r\hat{\varepsilon}$

Table 4.1 shows the different scenarios when $\hat{\mu}$ and $\hat{\varepsilon}$ varies with the pure relative error criterion, $\mathrm{tol} = \varepsilon_r |\mu|$. At scenario 3 and 4, where $|\hat{\varepsilon}| > |\hat{\mu}|$, the stopping criterion is equivalent to $\varepsilon_r \geq 1$. This also verifies the assumptions we made in (4.5) and (4.8).

Note that, if $\hat{\mu} > 0$, $\Delta_-(\hat{\mu}, \hat{\varepsilon})$ will be negative according to the definition of tol and Δ_-. Similarly, if $\hat{\mu} < 0$, $\Delta_-(\hat{\mu}, \hat{\varepsilon})$ will be positive. Thus, $\tilde{\mu}$ is smaller than $\hat{\mu}$ in magnitude. The shift Δ_- may offset the scenario that we over estimate $\hat{\mu}$, which makes $\tilde{\mu}$ more easy to satisfy the generalized error criterion. Suppose $\hat{\mu} > \mu > \hat{\varepsilon} > 0$,

and we pursue a pure relative case, which is $\text{tol}(\varepsilon_a, \varepsilon_r |\mu|) = \varepsilon_r |\mu|$. Without the shift Δ_-, we can get

$$|\mu - \hat{\mu}| \leq \hat{\varepsilon} \leq \Delta_+(\hat{\mu}, \hat{\varepsilon}) = \varepsilon_r |\hat{\mu}|, \tag{4.18}$$

by (4.2) and the stopping criterion (4.11). It is equivalent to

$$|\mu - \hat{\mu}| \leq \text{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu}|).$$

We want the right hand side of the inequality be $\text{tol}(\varepsilon_a, \varepsilon_r |\mu|)$. According to the derivation above, if we shift the estimator $\hat{\mu}$ towards zero then the new estimator $\tilde{\mu}$ will satisfy the condition

$$|\mu - \tilde{\mu}| \leq \text{tol}(\varepsilon_a, \varepsilon_r |\mu|).$$

4.2 The Algorithm `meanMC_g`

In this section, we introduce an algorithm that estimates the mean of a random variable to a specified generalized error tolerance $\text{tol}(\varepsilon_a, \varepsilon_r |\mu|)$ defined in Definition 2.9.2. Before presenting our algorithm, we first introduce some notations. For any fixed $n \in \mathbb{R}$, $\alpha \in (0, 1)$, and $M > 0$, define the inverse of the functions $N_{\text{Cheb}}(\cdot, \alpha)$, $N_{\text{BE}}(\cdot, \alpha, M)$, and $N_{\text{CB}}(\cdot, \alpha, M)$ defined in (2.8), (2.13) and (3.5):

$$N_{\text{Cheb}}^{-1}(n; \alpha) := \sqrt{n\alpha}, \tag{4.19}$$

$$N_{\text{BE}}^{-1}(n; \alpha, M) := \min\left\{b > 0 : \Phi\left(-\sqrt{n}/b\right) + \delta_n(\sqrt{n}/b, M) \leq \frac{\alpha}{2}\right\}, \tag{4.20}$$

$$N_{\text{CB}}^{-1}(n; \alpha, M) := \min\left(N_{\text{Cheb}}^{-1}(n, \alpha), N_{\text{BE}}^{-1}(n, \alpha, M)\right), \tag{4.21}$$

where Φ is the standard Gaussian cumulative distribution function and δ_n is defined in (2.11). It then follows by Chebyshev's Inequality and the Berry-Esseen Inequality that

$$\Pr[|\hat{\mu}_n - \mu| < \hat{\varepsilon}] \geq 1 - \alpha, \quad \text{provided } \kappa \leq \kappa_{\max}, \text{ where } \hat{\varepsilon} = \sigma / N_{\text{CB}}^{-1}\left(n, \alpha, \kappa_{\max}^{3/4}\right),$$

and σ is the standard deviation of the random variable Y defined in (2.4).

To prepare the algorithm, given the uncertainty $\alpha \in (0,1)$, let $\alpha_\sigma, \alpha_1, \alpha_2, \ldots$ be an infinite sequence of positive numbers all less than one, such that

$$(1-\alpha_\sigma)\left[1-(\alpha_1+\alpha_2+\cdots)\right] = 1-\alpha. \tag{4.22}$$

For example, one might choose α_σ and

$$\alpha_t = \frac{1-\alpha_\sigma}{\alpha-\alpha_\sigma} 2^{-t},\ t \in \mathbb{N}, \quad \text{where } a \in (1,\infty). \tag{4.23}$$

Table 4.2. Parameters specified in Algorithm 4.2.1 and 4.4.1 .

parameters	description	default
α	the uncertainty level, $1 < \alpha < 1$	1%
α_σ	the uncertainty level for variance estimation, $1 < \alpha_\sigma < \alpha$	$\alpha/2$
α_t	the uncertainty level satisfying condition (4.22)	$\frac{1-\alpha_\sigma}{\alpha-\alpha_\sigma}2^{-t}$
n_σ	the sample size for variance estimation, $n_\sigma \in \mathbb{N}$	10^4
n_1	initial sample size for mean estimation, $n_1 \in \mathbb{N}$	10^4
$\mathfrak{C}$	standard deviation inflation factor (SDIF), $\mathfrak{C} > 1$	1.2
θ	parameter used to choose next tolerance, $0 < \theta < 1$	0.95
tol	a generalized error criterion in Definition 2.9.2	max

Algorithm 4.2.1. *This algorithm requires an user to provide several inputs:*

- *a random number generator* Yrand *that takes the sample size n as an input. It generates n IID random numbers from the distribution of Y ,*
- *the absolute error tolerance* $\varepsilon_a \geq 0$*,*

- *the relative error tolerance* $0 \leq \varepsilon_r < 1$ *(we require* $\varepsilon_a + \varepsilon_r > 0$*),*

With the parameters given in Table 4.2, we first calculate $\kappa_{\max} = \kappa_{\max}(n_\sigma, \alpha_\sigma, \mathfrak{C})$ *as defined in* (2.18)*, then do the following:*

1. *Compute the sample variance,* $s^2_{n_\sigma}$*, using* n_σ *IID samples from the distribution of* Y*. Then approximate the variance of* Y *by* $\hat{\sigma}^2 = \mathfrak{C}^2 s^2_{n_\sigma}$*.*

2. *If the relative error tolerance* $\varepsilon_r = 0$*, the algorithm becomes a two stage one instead of multiple stage one. Then calculate the uncertainty level for the second stage estimation,* $\alpha_1 = 1 - (1-\alpha)/(1-\alpha_\sigma)$*, and the sample size for mean estimation,*

$$n_1 = N_{\mathrm{CB}}\left(\hat{\sigma}/\varepsilon_a, \alpha_1, \kappa_{\max}^{3/4}\right).$$

   Sample n *IID random numbers from distribution of* Y *and compute*

$$\tilde{\mu} = \hat{\mu}_1 = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

   Terminate the algorithm.

3. *Else, calculate the width of the initial confidence interval for the mean estimation,*

$$\hat{\varepsilon}_1 = \hat{\sigma}/N_{\mathrm{CB}}^{-1}\left(n_1, \alpha_1, \kappa_{\max}^{3/4}\right). \tag{4.24}$$

   For $t = 1, 2, \cdots$*, do the following:*

   (a) *Compute* $\hat{\mu}_t$ *and* $\Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t)$ *using sample size* n_t *and tolerance* $\hat{\varepsilon}_t$*, where* Δ_+ *is defined in* (4.9)*.*

   (b) *If* $\Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t) \geq \hat{\varepsilon}_t$*, then* Δ_+ *is large enough. Set stopping time* $\tau = t$ *and Let* $\tilde{\mu} = \hat{\mu}_\tau + \Delta_-(\hat{\mu}_\tau, \hat{\varepsilon}_\tau)$*. Terminate the algorithm.*

(c) Else, define the next tolerance,

$$\hat{\varepsilon}_{t+1} = \max\left(\hat{\varepsilon}_{t-1}/10, \min\left(\hat{\varepsilon}_t/2, \max(\varepsilon_a, \theta\varepsilon_r\left|\hat{\mu}_t\right|)\right)\right)$$

and the next sample size,

$$n_{t+1} = N_{\mathrm{CB}}\left(\hat{\sigma}/\hat{\varepsilon}_{t+1}, \alpha_{t+1}, \kappa_{\max}^{3/4}\right).$$

Increase t by one and go back to step (3a)*.*

The success of Algorithm 4.2.1 has been proved in several steps stated below

- Algorithm 4.2.1 terminates in a finite step almost surely is proved in Lemma 4.2.5.
- The generalized error criterion, (1.4), is satisfied when the algorithm terminates is proved in Theorem 4.2.7.
- The cost upper bound of Algorithm 4.2.1 is derived in Theorem 4.3.1.

Before proving the success of the algorithm, we need the following lemmas.

Lemma 4.2.2. *If the function* $\mathrm{tol} : [0,\infty)\times[0,\infty) \to [0,\infty)$ *is non-decreasing in each of its arguments and satisfies a Lipschitz condition in terms of its second argument, i.e.:*

$$|\,\mathrm{tol}(a,b) - \mathrm{tol}(a,b')| \le |b-b'| \quad \forall a,b,b' \ge 0. \tag{4.25}$$

Then, for $\varepsilon_a \ge 0$, $\varepsilon_r \ge 0$ *and* $\varepsilon' \ge 0$*, the following inequality must be true:*

$$\mathrm{tol}(\varepsilon_a, \varepsilon_r\left|\hat{\mu} \pm \hat{\varepsilon}\right|) \ge \mathrm{tol}(\varepsilon_a, \varepsilon_r\left|\mu\right|) - \varepsilon_r(|\mu - \hat{\mu}| + \hat{\varepsilon}). \tag{4.26}$$

Proof. By triangle inequality and Lipschitz condition of tol, we have the inequality below

$$\mathrm{tol}(a,b) - \mathrm{tol}(a,b') \le |\mathrm{tol}(a,b) - \mathrm{tol}(a,b')| \le |b-b'|\,,$$

which is equivalent to:

$$\text{tol}(a, b') \geq \text{tol}(a, b) - |b - b'| . \tag{4.27}$$

Then we choose $a = \varepsilon_a$, $b = \varepsilon_r |\mu|$ and $b' = \varepsilon_r |\hat{\varepsilon} \pm \hat{\mu}|$. By applying inequality (4.27) and the triangle inequality, the following inequality must be true:

$$\begin{aligned}
\text{tol}(\varepsilon_a, \varepsilon_r |\hat{\mu} \pm \hat{\varepsilon}|) &\geq \text{tol}(\varepsilon_a, \varepsilon_r |\mu|) - \varepsilon_r ||\mu| - |\hat{\mu} \pm \hat{\varepsilon}|| \\
&\geq \text{tol}(\varepsilon_a, \varepsilon_r |\mu|) - \varepsilon_r |-\mu + \hat{\mu} \pm \hat{\varepsilon}| \\
&\geq \text{tol}(\varepsilon_a, \varepsilon_r |\mu|) - \varepsilon_r(|\mu - \hat{\mu}| + \hat{\varepsilon}).
\end{aligned}$$

□

Lemma 4.2.3. *Given an absolute error tolerance $\varepsilon_a \geq 0$, a relative error tolerance $\varepsilon_r \geq 0$ and an error tolerance $\hat{\varepsilon}$. If $\hat{\varepsilon}$ satisfies this condition:*

$$\hat{\varepsilon} \leq \frac{\text{tol}(\varepsilon_a, \varepsilon_r |\mu|)}{1 + \varepsilon_r},$$

and condition (4.2) *Then the inequality*

$$\hat{\varepsilon} \leq \Delta_+(\hat{\mu}, \hat{\varepsilon})$$

must be satisfied.

Proof. Since we assume $|\mu - \hat{\mu}| \leq \hat{\varepsilon}$, it follows by the triangle inequality that,

$$|\mu| \leq |\hat{\mu}| + \hat{\varepsilon}.$$

Since tol is non-decreasing on its second argument, we have the following inequality

$$\text{tol}(\varepsilon_a, \varepsilon_r(|\hat{\mu}| + \hat{\varepsilon})) \geq \text{tol}(\varepsilon_a, \varepsilon_r |\mu|). \tag{4.28}$$

Next, applying (4.26) and (4.2) yields:

$$\begin{aligned}
\text{tol}\left(\varepsilon_a, \varepsilon_r ||\hat{\mu}| - \hat{\varepsilon}|\right) &\geq \text{tol}\left(\varepsilon_a, \varepsilon_r |\mu|\right) - \varepsilon_r \left(|\mu - \hat{\mu}| + \hat{\varepsilon}\right) \\
&\geq \text{tol}\left(\varepsilon_a, \varepsilon_r |\mu|\right) - 2\varepsilon_r \hat{\varepsilon}.
\end{aligned} \tag{4.29}$$

By the definition of Δ_+ in (4.9) and the inequality (4.28) and (4.29),

$$\begin{aligned}\Delta_+(\hat{\mu},\hat{\varepsilon}) &= \frac{1}{2}\left(\mathrm{tol}\left(\varepsilon_a,\varepsilon_r\left(|\hat{\mu}|+\hat{\varepsilon}\right)\right)+\mathrm{tol}\left(\varepsilon_a,\varepsilon_r\left||\hat{\mu}|-\hat{\varepsilon}\right|\right)\right)\\ &\geq \frac{1}{2}\left(\mathrm{tol}\left(\varepsilon_a,\varepsilon_r\left|\mu\right|\right)+\mathrm{tol}\left(\varepsilon_a,\varepsilon_r\left|\mu\right|\right)-2\varepsilon_r\hat{\varepsilon}\right)\\ &= \mathrm{tol}\left(\varepsilon_a,\varepsilon_r\left|\mu\right|\right)-\varepsilon_r\hat{\varepsilon}. \end{aligned} \tag{4.30}$$

If we subtract $\hat{\varepsilon}$ from both side of the inequality (4.30), it becomes:

$$\Delta_+(\hat{\mu},\hat{\varepsilon})-\hat{\varepsilon}\geq \mathrm{tol}(\varepsilon_a,\varepsilon_r\left|\mu\right|)-(1+\varepsilon_r)\hat{\varepsilon}\geq 0.$$

By the hypothesis of the theorem, this completes the proof. □

With the preparation of Lemma 4.2.2 and Lemma 4.2.3, we are going to prove the stopping time τ in Algorithm 4.2.1 has an upper bound, hence, it is finite almost surely.

Lemma 4.2.4. *Given ε_a, ε_r, n_1, α_1, and $\kappa_{\max}(n_\sigma,\alpha_\sigma,\mathfrak{C})$ defined in (3.6), which are parameters given in Algorithm 4.2.1. Let β be an uncertainty level and define the upper bound on tolerance $\hat{\varepsilon}_1$ as:*

$$\hat{\varepsilon}_{1\,\mathrm{up}}(\beta) := \hat{\sigma}_{\mathrm{up}}(\beta)/N_{\mathrm{CB}}^{-1}\left(n_1,\alpha_1,\kappa_{\max}^{3/4}\right), \tag{4.31}$$

where $\hat{\sigma}_{\mathrm{up}}(\beta)$ is defined in (2.21). The stopping time τ for Algorithm 4.2.1, has a probabilistic upper bound

$$\bar{\tau}(\beta) := \left\lceil 1+\log_2\left(\frac{\hat{\varepsilon}_{1\,\mathrm{up}}(\beta)(1+\varepsilon_r)}{\mathrm{tol}(\varepsilon_a,\varepsilon_r\left|\mu\right|)}\right)\right\rceil \tag{4.32}$$

$$= 1+\left\lceil \log_2\left(\frac{\sqrt{\left(\mathfrak{C}^2+(\mathfrak{C}^2-1)\sqrt{\frac{\alpha_\sigma(1-\beta)}{(1-\alpha_\sigma)\beta}}\right)}\sigma(1+\varepsilon_r)}{N_{\mathrm{CB}}^{-1}\left(n_1,\alpha_1,\kappa_{\max}^{3/4}\right)\mathrm{tol}\left(\varepsilon_a,\varepsilon_r\left|\mu\right|\right)}\right)\right\rceil \tag{4.33}$$

with confidence level $1-\beta$, i.e.:

$$\Pr(\tau<\bar{\tau}(\beta))\geq 1-\beta. \tag{4.34}$$

Proof. The stopping time τ is defined as:

$$\tau = \min\{t : \Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t) \geq \hat{\varepsilon}_t\} \tag{4.35}$$

and the tolerance in each step of iteration is defined as:

$$\hat{\varepsilon}_{t+1} = \min(\hat{\varepsilon}_t/2, \text{tol}(\varepsilon_a, \theta\varepsilon_r \, |\hat{\mu}_t|)).$$

The new tolerance must be no greater than half of the previous tolerance, i.e. $\hat{\varepsilon}_{t+1} \leq \hat{\varepsilon}_t/2$, which yields:

$$\hat{\varepsilon}_t \leq \hat{\varepsilon}_1/2^{t-1}. \tag{4.36}$$

If the tolerance ε_t satisfies the upper bound in Lemma 4.2.3, which is,

$$\varepsilon_t \leq \frac{\text{tol}(\varepsilon_a, \varepsilon_r \, |\mu|)}{1+\varepsilon_r},$$

then it must also satisfy

$$\varepsilon_t \leq \Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t).$$

Define an upper bound on stopping time τ in terms of the random $\hat{\varepsilon}_1$ as follows

$$T := 1 + \left\lceil \log_2 \left(\frac{\hat{\varepsilon}_1(1+\varepsilon_r)}{\text{tol}(\varepsilon_a, \varepsilon_r \, |\mu|)} \right) \right\rceil.$$

This definition of T then implies that:

$$\begin{aligned}
&T \geq 1 + \log_2 \left(\frac{\hat{\varepsilon}_1(1+\varepsilon_r)}{\text{tol}(\varepsilon_a, \varepsilon_r \, |\mu|)} \right) \\
&\Leftrightarrow \frac{\text{tol}(\varepsilon_a, \varepsilon_r \, |\mu|)}{1+\varepsilon_r} \geq \hat{\varepsilon}_1/2^{T-1} \\
&\Rightarrow \frac{\text{tol}(\varepsilon_a, \varepsilon_r \, |\mu|)}{1+\varepsilon_r} \geq \hat{\varepsilon}_\tau \quad \text{by inequality (4.36)} \\
&\Rightarrow \Delta_{+,T} \geq \hat{\varepsilon}_T \quad \text{by Lemma 4.2.3.}
\end{aligned}$$

Since the first occurrence $\hat{\varepsilon}_t \leq \Delta_+(\hat{\varepsilon}_t, \hat{\varepsilon}_t)$ defines the stopping time τ, we must have

$$\tau \leq T = 1 + \left\lceil \log_2 \left(\frac{\hat{\varepsilon}_1(1+\varepsilon_r)}{\text{tol}(\varepsilon_a, \varepsilon_r \, |\mu|)} \right) \right\rceil. \tag{4.37}$$

Then by (4.37) and (4.32), we have the relationship below:

$$\hat{\varepsilon}_1 < \hat{\varepsilon}_{1\,\mathrm{up}}(\beta) \Rightarrow \tau \leq T < \bar{\tau}(\beta). \tag{4.38}$$

Also, according to the definition of $\hat{\varepsilon}_1$ in (4.24) and $\hat{\varepsilon}_{1\,\mathrm{up}}(\beta)$ in (4.31), which are functions of $\hat{\sigma}$ and $\hat{\sigma}_{\mathrm{up}}(\beta)$ respectively, we have the relationship below

$$\hat{\sigma} < \hat{\sigma}_{\mathrm{up}}(\beta) \Rightarrow \hat{\varepsilon}_1 < \hat{\varepsilon}_{1\,\mathrm{up}}(\beta). \tag{4.39}$$

Combining (4.38) and (4.39) yields

$$\hat{\sigma} < \hat{\sigma}_{\mathrm{up}}(\beta) \Rightarrow \tau < \bar{\tau}(\beta).$$

Apply Lemma 2.9.5, we have the probability relationship as below:

$$\Pr(\tau < \bar{\tau}(\beta)) \geq \Pr(\hat{\sigma} < \hat{\sigma}_{\mathrm{up}}(\beta)) \geq 1 - \beta.$$

□

Equation 4.33 gives an explicit form of the upper bound on the stopping time τ, i.e. $\bar{\tau}(\beta)$. As we could see, when the tolerance tol becomes small, the stopping time becomes large, which means we need more iteration steps. When β becomes small, the stopping time becomes large, and $\lim_{\beta\to 0} \bar{\tau}(\beta) = \infty$. Also, the stopping time is a function of the true standard deviation σ, when σ becomes large, we need more iteration steps. Next, we will prove that the stopping time is finite almost surely.

Lemma 4.2.5. *The stopping time τ is finite almost surely, i.e.:*

$$\Pr(\tau < \infty) = 1.$$

Proof. By Lemma 4.2.4, the stopping time τ has a probabilistic upper bound $\bar{\tau}(\beta)$ defined in (4.34). As $\bar{\tau}(\beta)$ is a function in terms of β, taking the limit that $\beta \to 0$ yields $\lim_{\beta\to 0} \bar{\tau}(\beta) = \infty$, which is

$$\Pr(\tau < \infty) = \Pr(\tau < \lim_{\beta\to 0} \bar{\tau}(\beta)). \tag{4.40}$$

By the continuity of probability in Lemma 2.9.6,

$$\Pr(\tau < \lim_{\beta\to 0} \bar{\tau}(\beta)) = \lim_{\beta\to 0} \Pr(\tau < \bar{\tau}(\beta)), \tag{4.41}$$

and preservation of inequality in Lemma 2.9.7, we have

$$\lim_{\beta\to 0} \Pr(\tau < \bar{\tau}(\beta)) \geq \lim_{\beta\to 0}(1-\beta) = 1. \tag{4.42}$$

By inequality (4.40), (4.41) and (4.42), the proof proceeds. □

Lemma 4.2.6. *Let Y be a random variable with mean μ, and either zero variance or positive variance with modified kurtosis $\kappa \leq \kappa_{\max}(\alpha_\sigma, n_\sigma, \mathfrak{C})$. Suppose that the width of the confidence interval, $\hat{\varepsilon}$, is a random number. It follows that Algorithm 3.1.1 above yields an estimate $\hat{\mu}$ given by* (3.8) *that satisfies the following condition*

$$\Pr\left(|\mu - \hat{\mu}| > \hat{\varepsilon} | \hat{\sigma} \geq \sigma\right) < \alpha_\mu. \tag{4.43}$$

Here, the samples of Y used to calculate $\hat{\mu}$ are independent of $\hat{\varepsilon}$, although the number of samples, n_μ, depends on $\hat{\varepsilon}$.

Proof. This result could be derived from the proof of the Theorem 3.1.2, by applying the conditional expectation on $\hat{\varepsilon}$. Let α_σ, n_σ, $\mathfrak{C}$ be the parameters and $\hat{\sigma}^2$ be a random variable depending on the random samples Y_i, $i = 1, 2, \cdots, n_\sigma$ as given in Algorithm 3.1.1. Also, let the tolerance $\hat{\varepsilon}$ be a random variable depending on $\hat{\sigma}^2$. The probability (4.43) can be written as:

$$\Pr(|\mu - \hat{\mu}| > \hat{\varepsilon} | \hat{\sigma} \geq \sigma) = \mathbb{E}_{\hat{\varepsilon}}\left[\Pr(|\mu - \hat{\mu}| > \hat{\varepsilon} | \hat{\sigma} \geq \sigma, \hat{\varepsilon})\right]. \tag{4.44}$$

We choose the random sample size, n_μ, to calculate $\hat{\mu}$ according to (2.12), namely $n_\mu = N_{\mathrm{CB}}\left(\hat{\sigma}/\hat{\varepsilon}, \alpha_\mu, \kappa_{\max}^{3/4}\right)$. The following inequality holds:

$$\begin{aligned}
&\mathbb{E}_{\hat{\varepsilon}}\left[\Pr(|\mu - \hat{\mu}| > \hat{\varepsilon} | \hat{\sigma} \geq \sigma, \hat{\varepsilon})\right] \\
&\geq \mathbb{E}_{\hat{\varepsilon}}\left[\Pr\left(\left|\mu - \frac{1}{n_\mu}\sum_{i=1}^{n_\mu} Y_i\right| > \hat{\varepsilon} \middle| \hat{\sigma} \geq \sigma, \hat{\varepsilon}, n_\mu = N_{\mathrm{CB}}\left(\hat{\sigma}/\hat{\varepsilon}, \alpha_\mu, \kappa_{\max}^{3/4}\right)\right)\right] \\
&< \alpha_\mu
\end{aligned}$$

□

Theorem 4.2.7. *Let Y be a random variable with mean μ, and nonnegative variance with modified kurtosis $\kappa \leq \kappa_{\max}(\alpha_\sigma, n_\sigma, \mathfrak{C})$. It follows that Algorithm 4.2.1 terminates in a finite time step almost surely. If the algorithm terminates, then the general error criterion,* (1.4)*, is satisfied.*

Proof. The stopping time is finite is proved in Lemma 4.2.5. Next, we will prove the success of the algorithm. In Algorithm 4.2.1, several quantities are random: the random variable Y, the stopping time τ, the sample size n_t, the confidence interval width $\hat{\varepsilon}_t$ in each iteration, for $t = 1, 2, \cdots, \tau$. τ is the first t for which the stopping criterion $\Delta_{+,t} \geq \hat{\varepsilon}_t$ is satisfied. Hence, we know the following events are equivalent

$$\{\tau = s\} \Leftrightarrow \{\Delta_+(\hat{\mu}_s, \hat{\varepsilon}_s) \geq \hat{\varepsilon}_s\} \& \{\Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t) \leq \hat{\varepsilon}_t, \text{for } t = 1, 2, \cdots, s-1\}. \tag{4.45}$$

Now let's consider the worst case. Expand the definition of $\hat{\mu}_t$ and $\hat{\varepsilon}_t$ given in Algorithm 4.2.1 for all $t = \tau + 1, \tau + 2, \cdots$ by defining $\hat{\varepsilon}_t = \hat{\varepsilon}_\tau$ for $t > \tau$. And let $\hat{\mu}_t$ be defined according to step 3a in Algorithm 4.2.1. Suppose that at stopping time τ, the desired error tolerance is not obtained, the probability of this event happens is

$$\begin{aligned}
&\Pr\left(|\mu - \tilde{\mu}| > \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|)\right) \\
&= \sum_{t=1}^{\infty} \Pr\left(|\mu - (\hat{\mu}_t + \Delta_-(\hat{\mu}_t, \hat{\varepsilon}_t))| > \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|), \tau = t\right) + \Pr(\tau = \infty) \\
&= \sum_{t=1}^{\infty} \Pr\left(|\mu - (\hat{\mu}_t + \Delta_-(\hat{\mu}_t, \hat{\varepsilon}_t))| > \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|), \tau = t\right).
\end{aligned} \tag{4.46}$$

By Proposition 4.1.1, we know

$$\begin{aligned}
&\Pr\left(|\mu - (\hat{\mu}_t + \Delta_-(\hat{\mu}_t, \hat{\varepsilon}_t))| > \mathrm{tol}(\varepsilon_a, \varepsilon_r |\mu|), \tau = t\right) \\
&\leq \Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t \text{ or } \Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t) < \hat{\varepsilon}_t, \tau = t\right).
\end{aligned}$$

The definition of stopping time in (4.45) implied that $\tau = t$ is incompatible with

$\Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t) < \hat{\varepsilon}_t$. Then the inequality is equal to

$$\Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t \text{ or } \Delta_+(\hat{\mu}_t, \hat{\varepsilon}_t) < \hat{\varepsilon}_t, \tau = t\right) = \Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t, \tau = t\right).$$

Adding extra condition of $\hat{\sigma}$ we have:

$$\begin{aligned}
&\Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t, \tau = t\right) \\
&= \Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t, \tau = t, \hat{\sigma} \geq \sigma\right) + \Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t, \tau = t, \hat{\sigma} < \sigma\right) \\
&\leq \Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t, \hat{\sigma} \geq \sigma\right) + \Pr\left(\tau = t, \hat{\sigma} < \sigma\right).
\end{aligned}$$

By Lemma 4.2.6, the inequality has the form

$$\Pr\left(|\mu - \hat{\mu}_t| > \hat{\varepsilon}_t, \hat{\sigma} \geq \sigma\right) + \Pr\left(\tau = t, \hat{\sigma} < \sigma\right) < \alpha_t \Pr(\hat{\sigma} \geq \sigma) + \Pr\left(\tau = t, \hat{\sigma} < \sigma\right).$$

Thus, by (4.46) and (4.47)

$$\begin{aligned}
\Pr\left(|\mu - \tilde{\mu}| > \text{tol}(\varepsilon_a, \varepsilon_r |\mu|)\right) &\leq \Pr(\hat{\sigma} \geq \sigma) \sum_{t=1}^{\infty} \alpha_t + \Pr(\hat{\sigma} < \sigma) \\
&= 1 - \Pr(\hat{\sigma} \geq \sigma) \left(1 - \sum_{t=1}^{\infty} \alpha_t\right),
\end{aligned}$$

which is

$$\begin{aligned}
\Pr\left(|\mu - \tilde{\mu}| \leq \text{tol}(\varepsilon_a, \varepsilon_r |\mu|)\right) &\geq \Pr(\hat{\sigma} \geq \sigma) \left(1 - \sum_{t=1}^{\infty} \alpha_t\right) \\
&\geq \left(1 - \sum_{t=1}^{\infty} \alpha_t\right) (1 - \alpha_\sigma) \\
&= 1 - \alpha.
\end{aligned}$$

Figure 4.1 illustrates the worst case when estimating $\tilde{\mu}$. Summing up all the 'bad' scenarios ends up with a worst case error bound. This completes the proof. □

4.3 The upper bound on cost of Algorithm `meanMC_g`

Theorem 4.3.1. *The cost of Algorithm 4.2.1 is defined as*

$$n_{\text{tot}} = n_\sigma + \sum_{t=1}^{\tau} n_t,$$

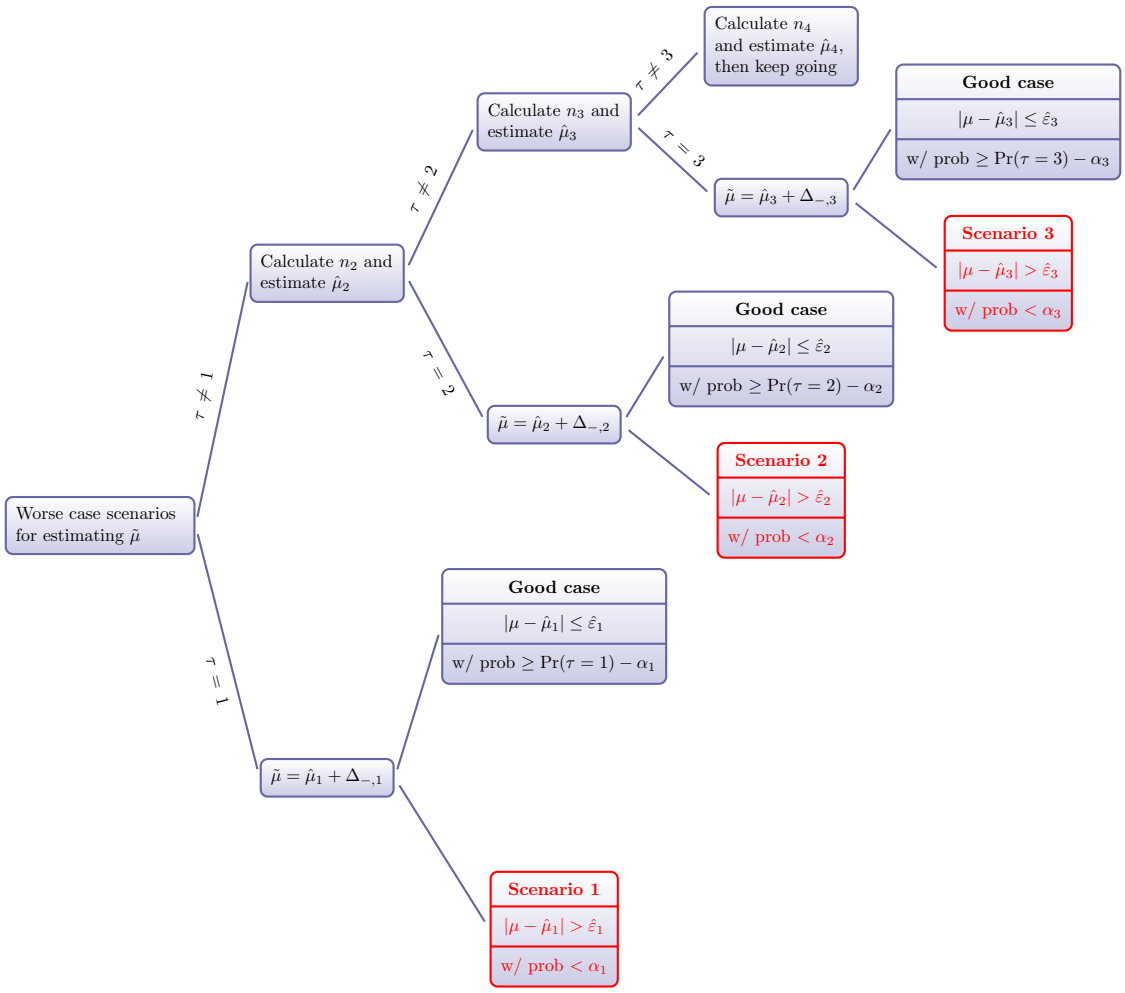

Figure 4.1. The red boxes are those 'bad' scenarios that the algorithm terminated with the estimator $\hat{\mu}$ not satisfying the fixed width confidence interval condition.

where τ is the stopping time, n_σ is the sample size used to calculate the sample variance $s^2_{n_\sigma}$, and n_t is the sample size used to estimate $\hat{\mu}_t$ in each iteration. It has a probabilistic upper bound

$$n_{\text{up}}(\beta) = n_\sigma + n_1 + \sum_{t=2}^{\bar{\tau}(\beta)} N_{\text{CB}}\left(10^{t-1} N_{\text{CB}}^{-1}\left(n_1, \alpha_1, \kappa_{\max}^{3/4}\right), \alpha_t, \kappa_{\max}^{3/4}\right),$$

where N_{CB}, N_{CB}^{-1} and $\bar{\tau}(\beta)$ are defined in (3.5), (4.21) *and* (4.32). *It satisfies the probability inequality:*

$$\Pr\left(n_{\text{tot}} \leq n_{\text{up}}(\beta)\right) \geq 1 - \beta. \tag{4.47}$$

Proof. The number of samples required by the Algorithm 4.2.1 is $n_\sigma + \sum_{t=1}^{\tau} n_t$. Although n_σ and n_1 are deterministic, the $n_2, n_3, \cdots, n_\tau$ and τ are random variables. The cost of this algorithm may be defined probabilistically. The dependency of the random variables is as follows:

- $\hat{\sigma}$ depends on the random samples $Y_1, \cdots, Y_{n_\sigma}$;
- n_1 is given; $\hat{\varepsilon}_1$ depends on $\hat{\sigma}$ and n_1;
- $\hat{\mu}_t$ depends on n_t and random samples $Y_1, \cdots, Y_{n_t}$ where $t = 1, 2, \cdots, \tau$;
- $\hat{\varepsilon}_t$ depends on $\hat{\varepsilon}_{t-1}$, $\hat{\mu}_{t-1}$, where $t = 2, 3, \cdots, \tau$;
- n_t depends on $\hat{\sigma}$ and $\hat{\varepsilon}_t$, where $t = 2, 3, \cdots, \tau$;

In order to bound the total cost, we need to bound the stopping time τ and also the sample size needed in each iteration. The stopping time τ has a probabilistic bound by $\bar{\tau}$ defined in (4.33) as been proved in Lemma 4.2.4. In each iteration, the sample size n_t is defined as:

$$n_t = N_{\text{CB}}\left(\hat{\sigma}/\hat{\varepsilon}_t, \alpha_t, \kappa_{\max}^{3/4}\right).$$

Since $\hat{\varepsilon}_t$ has the form

$$\hat{\varepsilon}_t = \max\left(\hat{\varepsilon}_{t-1}/10, \min(\hat{\varepsilon}_{t-1}/2, \max(\varepsilon_a, \theta\varepsilon_r\left|\hat{\mu}_{t-1}\right|))\right),$$

we have $\hat{\varepsilon}_t \geq \hat{\varepsilon}_{t-1}/10$, which is also equivalent to $\hat{\varepsilon}_t \geq 10^{-t+1}\hat{\varepsilon}_1$. Define the lower bound of $\hat{\varepsilon}_t$ as

$$\underline{\varepsilon_t} = 10^{-t+1}\hat{\varepsilon}_1 = 10^{-t+1}\hat{\sigma}/N_{\mathrm{CB}}^{-1}\left(n_1, \alpha_1, \kappa_{\max}^{3/4}\right),$$

then we have

$$n_t = N_{\mathrm{CB}}\left(\hat{\sigma}/\hat{\varepsilon}_t, \alpha_i, \kappa_{\max}^{3/4}\right) \leq N_{\mathrm{CB}}\left(10^{t-1}N_{\mathrm{CB}}^{-1}\left(n_1, \alpha_1, \kappa_{\max}^{3/4}\right), \alpha_t, \kappa_{\max}^{3/4}\right). \tag{4.48}$$

Thus, the total cost can be bounded as follows:

$$\begin{aligned}
&\Pr(n_{\mathrm{tot}} \leq n_{\mathrm{up}}(\beta)) \\
&= \Pr\left(\sum_{t=2}^{\tau} n_t \leq \sum_{t=2}^{\bar{\tau}(\beta)} N_{\mathrm{CB}}\left(10^{t-1}N_{\mathrm{CB}}^{-1}\left(n_1, \alpha_1, \kappa_{\max}^{3/4}\right), \alpha_t, \kappa_{\max}^{3/4}\right)\right) \\
&\geq \Pr\left(n_t \leq N_{\mathrm{CB}}\left(10^{-t+1}N_{\mathrm{CB}}^{-1}\left(n_1, \alpha_1, \kappa_{\max}^{3/4}\right), \alpha_t, \kappa_{\max}^{3/4}\right), t = 1, \cdots, \bar{\tau}(\beta), \tau < \bar{\tau}(\beta)\right).
\end{aligned}$$

Since the first condition inside the probability is true for sure as been derived in (4.48), we have:

$$\Pr\left(n_{\mathrm{tot}} \leq n_{\mathrm{up}}(\beta)\right) \geq \Pr\left(\tau < \bar{\tau}(\beta)\right) \geq 1 - \beta.$$

□

Note that, this bound may be loose, finding a tighter bound could be a possible future work.

4.4 The Algorithm `cubMC_g`

An algorithm that performs numerical integration via Monte Carlo sampling with a generalized error tolerance will be presented here. The integrand and parameters are same as in Section 3.3, but with a generalized error tolerance $\mathrm{tol}(\varepsilon_a, \varepsilon_r\left|\mu\right|)$ as

given in Definition 2.9.2. The algorithm takes inputs: the integrand f, the integration hyperbox $(\boldsymbol{a}, \boldsymbol{b})$, and the probability density function ρ, the absolute error tolerance ε_a and relative error tolerance ε_r, and does the iteration until the estimator $\tilde{I}$ satisfies the fixed width confidence interval condition

$$\Pr\left(\left|I - \tilde{I}\right| \leq \operatorname{tol}\left(\varepsilon_a, \varepsilon_r \left|I\right|\right)\right) \geq 1 - \alpha. \tag{4.49}$$

The detailed algorithm is explained in Algorithm 4.4.1.

Algorithm 4.4.1. *The algorithm requires the user to provide the following inputs:*

- *the integrand f,*
- *the integration interval $(\boldsymbol{a}, \boldsymbol{b})$,*
- *the probability density function ρ,*
- *the absolute error tolerance $\varepsilon_a \geq 0$, and the relative error tolerance $0 \leq \varepsilon_r < 1$, that satisfy the condition $\varepsilon_a + \varepsilon_r > 0$,*

With the given parameters given in Table 4.2, we first calculate $\kappa_{\max} = \kappa_{\max}(n_\sigma, \alpha_\sigma, \mathfrak{C})$ as defined in (2.18)*, then do the following:*

1. *Sample n_σ values of random vector $\boldsymbol{X}_i$ from the distribution of $\boldsymbol{X}$ with probability density function $\rho(\boldsymbol{X})$ within the integration interval $(\boldsymbol{a}, \boldsymbol{b})$ and calculate the function values of $f(\boldsymbol{X}_i)$. Use this to calculate the sample mean*

$$\hat{I}_{n_\sigma} = \frac{1}{n_\sigma} \sum_{i=1}^{n_\sigma} f(\boldsymbol{X}_i),$$

and sample variance

$$s_{n\sigma}^2 = \frac{1}{n_\sigma - 1} \sum_{i=1}^{n_\sigma} \left(f(\boldsymbol{X}_i) - \hat{I}_{n_\sigma}\right)^2.$$

Also approximate the upper bound on variance by $\hat{\sigma}^2 = \mathfrak{C}^2 s_{n_\sigma}^2$.

2. *If the relative error tolerance $\varepsilon_r = 0$, then the algorithm becomes a two stage one instead of multiple stage one. Then calculate the uncertainty level for the second stage estimation, $\alpha_1 = 1 - (1-\alpha)/(1-\alpha_\sigma)$, and the sample size for mean estimation,*

$$n_1 = N_{\mathrm{CB}}\left(\hat{\sigma}/\varepsilon_a, \alpha_1, \kappa_{\max}^{3/4}\right),$$

where N_{CB} is defined in (3.5)*. Sample n_1 values of random vector $\boldsymbol{X}_i$ that are independent of those used to calculate $\hat{\sigma}^2$ from the distribution of $\boldsymbol{X}$ with probability density function $\rho(\boldsymbol{X})$, then calculate the n_1 function values of $f(\boldsymbol{X}_i)$. Use this to calculate the sample mean*

$$\tilde{I} = \frac{1}{n_1} \sum_{i=n_\sigma+1}^{n_\sigma+n_1} f(\boldsymbol{X}_i). \tag{4.50}$$

Terminate the algorithm.

3. *Else, calculate the width of the initial confidence interval for the integral estimation,*

$$\hat{\varepsilon}_1 = \hat{\sigma} N_{\mathrm{CB}}^{-1}\left(n_1, \alpha_1, \kappa_{\max}^{3/4}\right), \tag{4.51}$$

where N_{CB}^{-1} is defined in (4.21)*. For $t = 1, 2, \cdots$, do the following:*

   (a) *Compute $\hat{I}_t$ and $\Delta_+(\hat{I}_t, \hat{\varepsilon}_t)$ using sample size n_t and tolerance $\hat{\varepsilon}_t$, where Δ_+ is defined in* (4.9)*.*

   (b) *If $\Delta_+(\hat{I}_t, \hat{\varepsilon}_t) \geq \hat{\varepsilon}_t$, then Δ_+ is large enough. Set the stopping time $\tau = t$ and let $\tilde{I} = \hat{I}_\tau + \Delta_-(\hat{I}_t, \hat{\varepsilon}_t)$. Terminate the algorithm.*

   (c) *Else, define the next tolerance,*

$$\hat{\varepsilon}_{t+1} = \max\left(\hat{\varepsilon}_{t-1}/10, \min\left(\hat{\varepsilon}_t/2, \max\left(\varepsilon_a, \theta\varepsilon_r \left|\hat{I}_t\right|\right)\right)\right),$$

$$n_{t+1} = N_{\mathrm{CB}}\left(\hat{\sigma}/\hat{\varepsilon}_{t+1}, \alpha_{t+1}, \kappa_{\max}^{3/4}\right).$$

   Increase t by one and go back to step (3a)*.*

If this algorithm terminates, then the general error criterion, (4.49)*, is satisfied.*

As `cubMC_g` applies the same argument as `meanMC_g`, hence, the proof is the same. In the next section, several numerical examples are implemented to test and compare different algorithms.

4.5 Numerical examples for `cubMC_g`

In this section, several integrands are used to test the timing and accuracy results for integrating problems over different algorithms such as `cubMC_g`, `cubSobol_g`, `cubLattice_g` and `integral` in MATLAB, The first three algorithms are part of Guaranteed Automatic Integration Library(GAIL) [CDH$^+$15], which includes several algorithms that do numerical integration and function approximation with guarantee.

4.5.1 Integrating a product function. A numerical integration problem with integrand (3.17) and the generalized error tolerance $\max(10^{-3}, 10^{-3}|I|)$ has been tested. Time and accuracy are recorded in Figure 4.2. The initial sample size is 2^{13} and the sample budget is 10^{10} for all of the algorithms. As we can see, `cubMC_g` has an accuracy of 100%, which is higher than the nominal value of 99%. As discussed before, `cubMC_g` uses a conservative way to construct the fixed width confidence interval for the numerical integration, which may take more time and cost more samples. It, however, gives a more reliable solution. On the other hand, `cubSobol_g` has 455 out of 500 samples that reach the desired accuracy, whereas, `cubLattice_g` performs poorly with this test function, only 328 out of 500 samples have met the tolerance criterion, which means the accuracy is only 65.6%.

4.5.2 Integrating Keister's test function. Keister [Kei96] considered the

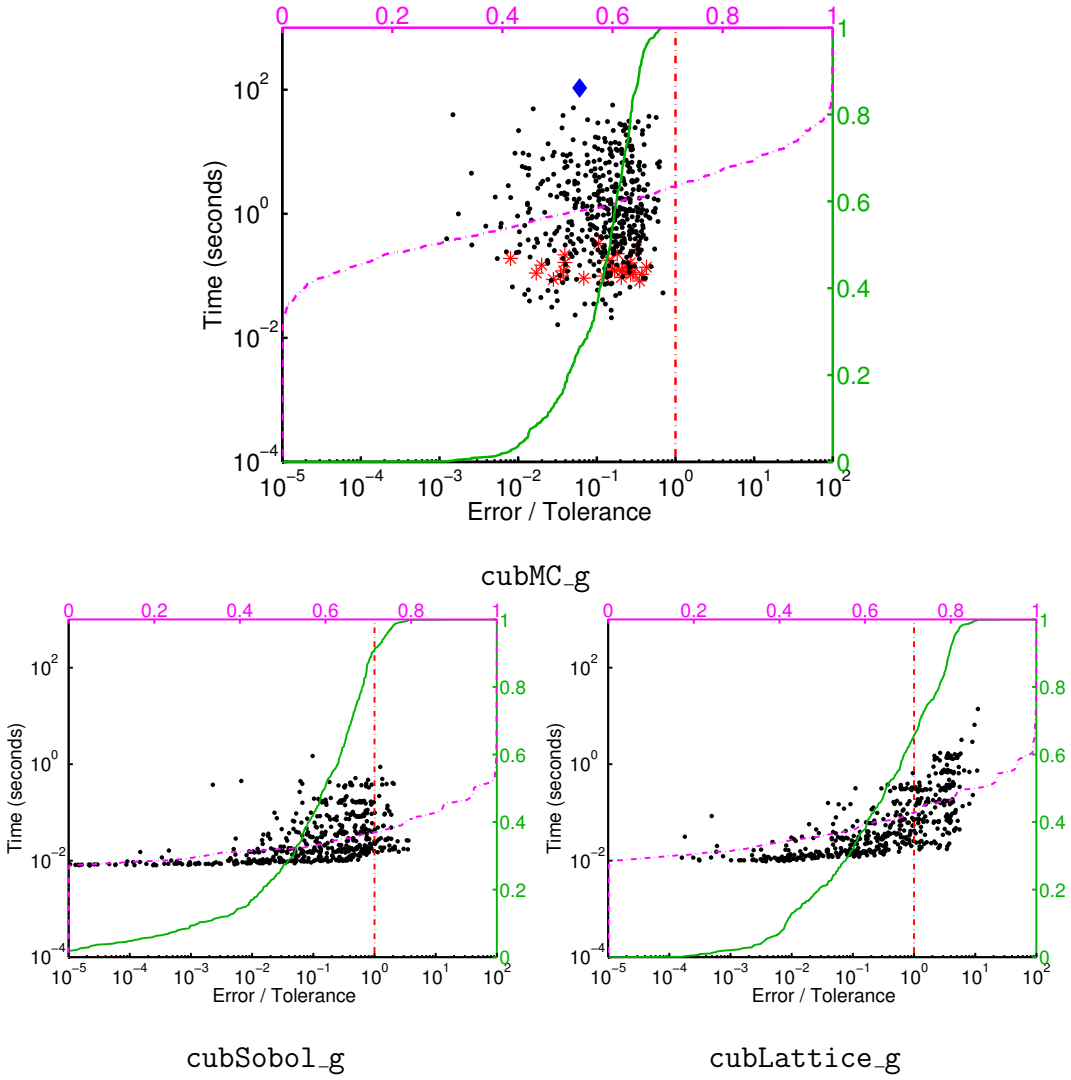

Figure 4.2. Execution times and errors for the prodcut test function (3.17) for $d = 2, 3, \cdots, 20$ and error tolerance $\max(10^{-3}, 10^{-3}|I|)$. The horizontal axis denotes the error/tolerance. The points lies on the left of the red vertical line means the tolerance was met. The points labeled with a diamond are those for which `cubMC_g` attempts to exceed the cost budget. The solid green curve denotes the empirical distribution function of the error, and the dot-dashed pink curve denotes the empirical distribution function of the time.

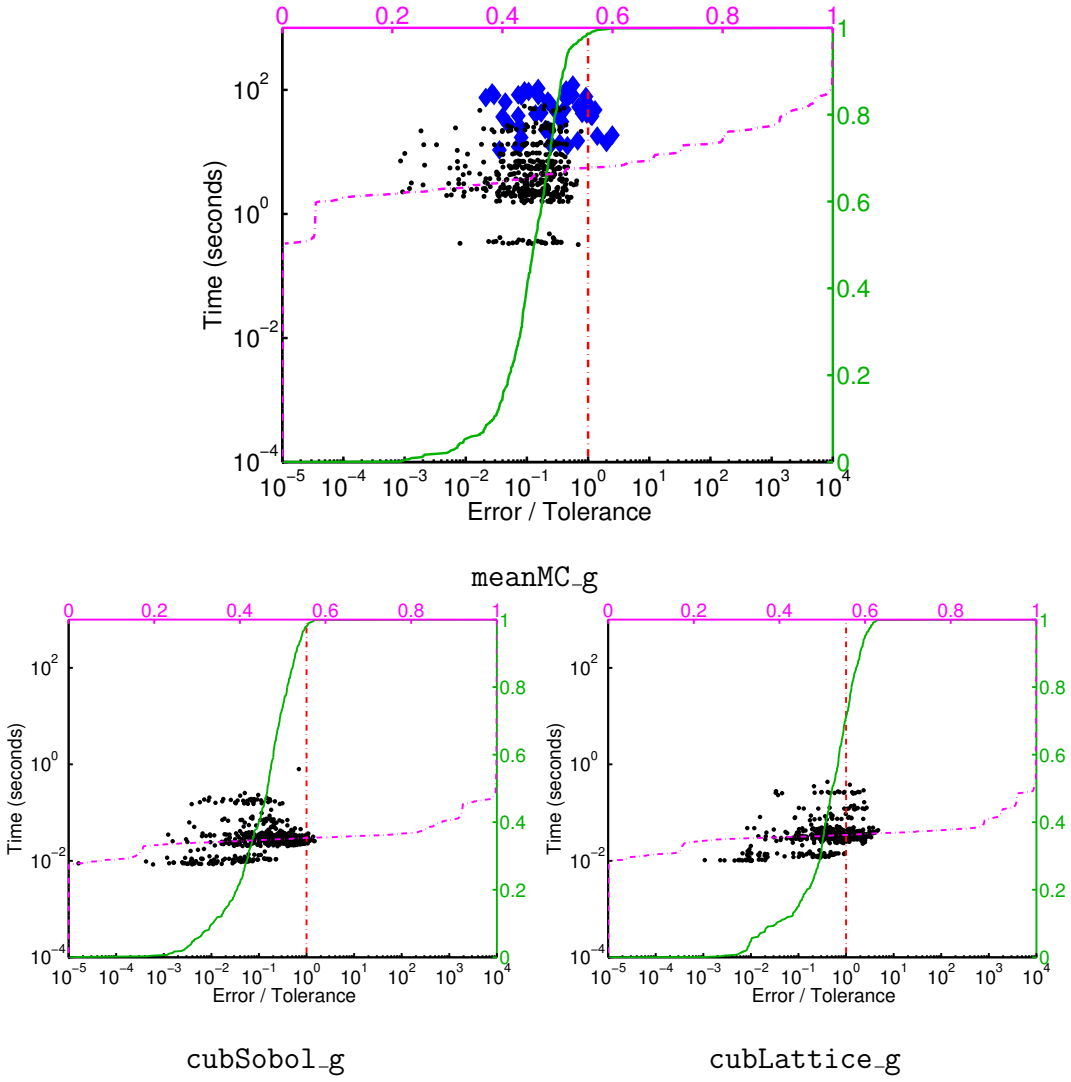

Figure 4.3. Execution times and errors for the Keister test function (4.52) with the dimension uniformly random chosen between 1 and 20. The tolerance is $\max(10^{-3}, 10^{-3}|I|)$. All other settings are same as in Figure 4.2.

following multidimensional integral that has applications in physics

$$
\begin{aligned}
I = \int_{\mathbb{R}^d} \cos(\|\boldsymbol{x}\|_2) e^{-\|\boldsymbol{x}\|_2^2} \mathrm{d}\boldsymbol{x} &= 2^{-d/2} \int_{\mathbb{R}^d} \cos(\|\boldsymbol{y}\|_2 / \sqrt{2}) e^{-\|\boldsymbol{y}\|_2^2/2} \mathrm{d}\boldsymbol{y} \\
&= \pi^{d/2} \int_{\mathbb{R}^d} \cos(\|\boldsymbol{y}\|_2 / \sqrt{2}) \frac{e^{-\|\boldsymbol{y}\|_2^2/2}}{(2\pi)^{d/2}} \mathrm{d}\boldsymbol{y} \\
&= \pi^{s/2} \int_{[0,1)^d} \cos\left(\sqrt{\sum_{j=1}^{d} \frac{[\Phi^{-1}(y_i)]^2}{2}} \right) \mathrm{d}\boldsymbol{y}, \qquad (4.52)
\end{aligned}
$$

where Φ denotes the standard Gaussian cumulative distribution function defined as below:

$$
\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-s^2/2} ds, x \in [-\infty, \infty].
$$

Let the integrand be

$$
f(\boldsymbol{x}) = \pi^{d/2} \cos\left(\sqrt{\frac{1}{2} \sum_{i=1}^{d} (\Phi^{-1}(x_i))^2} \right).
$$

The integration problem can be written as

$$
I = \int_{[0,1]^d} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.
$$

By applying the Monte Carlo method, the integral can be estimated as follows:

$$
\hat{I}_n = \frac{\pi^{d/2}}{n} \sum_{i=1}^{n} \cos\left(\sqrt{\frac{1}{2} \sum_{j=1}^{d} (\Phi^{-1}(x_{i,j}))^2} \right), \qquad (4.53)
$$

where $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{id}) \in \mathbb{R}$, $i = 1, \cdots, n$. Keister [Kei96] showed a formula for calculating the exact integral. With the analytical solution being computed, we compared the numerical results over three different algorithms with a generalized error criterion. Timing and accuracy were recorded in Figure 4.3.

Five hundred replications are performed for dimension d IID uniform in $[1, 20]$, with the integration interval $[0, 1]^d$. The analytic solution, I, is calculated and the true error is recorded. The generalized error tolerance is $\max(10^{-3}, 10^{-3}|I|)$. The initial sample size used to estimate the variance in `cubMC_g` is 2^{13} whereas the initial

Sobol's and Lattice points taken in `cubSobol_g` and `cubLattice_g` are 2^{13}. The sample budget is 10^{10}.

As we can see, `cubMC_g` takes significantly more time than `cubSobol_g` and `cubLattice_g`. The accuracy is 98.6%, which is a bit less than our confidence level, 99%. The reason is that the points labeled as a diamond are those attempt to exceed the sample budget 10^{10}, the guarantees do not hold for those points. If we exclude those 50 points labeled as a diamond and calculate the success rate again, it is 100% of the time. Although we have not calculated the exact kurtosis, which has a very complicated form, nor we check if it is less than the kurtosis maximum, our numerical result using `cubMC_g` is still better than other two quasi-Monte Carlo methods in the sense of the error criterion. For `cubSobol_g`, the error meets tolerance 98% of the time, whereas, for `cubLattice_g`, the success rate is only 71%. For this test function, we do not see the significant advantage of `cubMC_g`, the reason is that the test function is smooth and quasi-Monte Carlo methods perform well for smooth integrand. `cubMC_g` is more robust and does not assume any smoothness of the integrand. In this case `cubMC_g` does not take the advantage of the robustness and the mild assumption for the integrand, hence, it costs more time than `cubSobol_g` and `cubLattice_g`.

4.5.3 Integrating Multivariate Normal Probability with a deterministic covariance matrix. A function that has been used in many statistics applications is the numerical computation of Multivariate Normal Probability function

$$F(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{\sqrt{|\Sigma|\,(2\pi)^d}} \int_{a_1}^{b_1} \cdots \int_{a_d}^{b_d} e^{-\frac{1}{2}\boldsymbol{\theta}'\Sigma^{-1}\boldsymbol{\theta}} \mathrm{d}\boldsymbol{\theta}, \tag{4.54}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are $d \times 1$ vectors, $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_d)'$ and Σ is given as $d \times d$ symmetric positive definite covariance matrix. Genz [Gen92] points that if a_i is $-\infty$ and b_i is ∞ for some i, an appropriate transformation allows the ith variable to be integrated explicitly, which reduces the number of variables in the problem. Thus, for each i, we assume at most one of a_i and b_i be infinite.

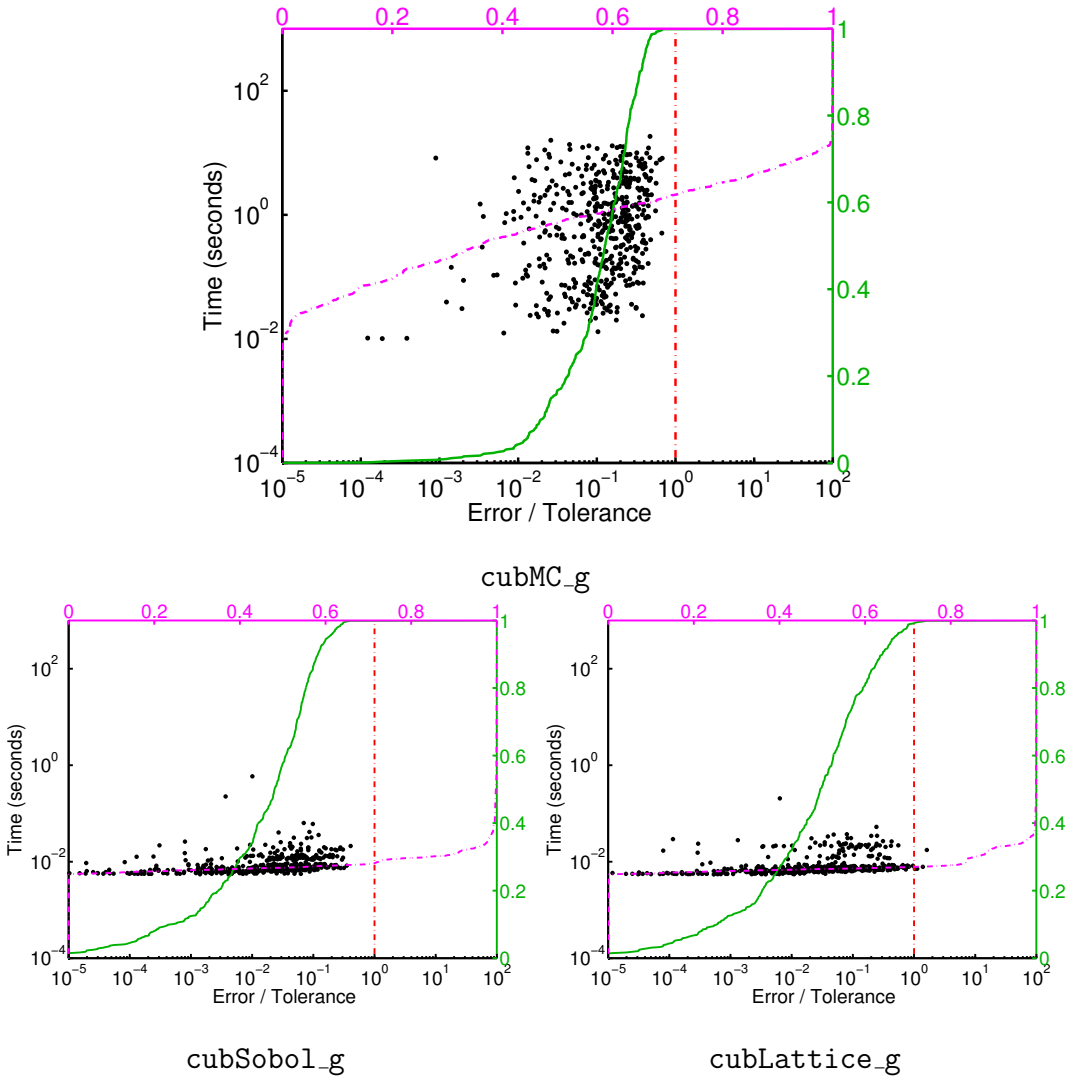

Figure 4.4. Execution times and errors for Multivariate Normal Probability test function (4.54) with the dimension d IID uniform in $[2, 8]$. The tolerance is $\max(10^{-3}, 10^{-4}|I|)$. All other settings are same as in Figure 4.2.

One way to compute F is by Monte Carlo. However, the infinite integration limits need to be handled either by some type of transformation to a finite region or by using some carefully selected cutoff values. Genz [Gen92, GB02] proposed a transformation that used to transform the original d dimensional integral into a new $d-1$ dimensional integral over a unit cube. The transformation was done by three steps:

1. First, apply Cholesky decomposition to the covariance matrix Σ, i.e. $\Sigma = CC'$, where C is a lower triangular $d \times d$ matrix. Let $\boldsymbol{\theta} = C\boldsymbol{y}$, so that $\boldsymbol{\theta}'\Sigma^{-1}\boldsymbol{\theta} = \boldsymbol{y}'C'(C')^{-1}C^{-1}C\boldsymbol{y} = y'y$, and $\mathrm{d}\boldsymbol{\theta} = |C|\,\mathrm{d}\boldsymbol{y} = |\Sigma|^{1/2}\,\mathrm{d}\boldsymbol{y}$.

2. Second, let

$$a_1' = a_1/c_{11}, \quad b_1' = b_1/c_{11}, \quad f_1 = \Phi(a'), \quad e_1 = \Phi(b'),$$

   where Φ is the standard Gaussian cumulative distribution function. Thus, for $i = 1, 2, \cdots, d$, recursively define

$$\begin{aligned}
y_i(w_1, \cdots, w_{i-1}) &= \Phi^{-1}(d_i + w_i(e_i - f_i)),\\
a_i'(w_1, \cdots, w_{i-1}) &= \frac{a_i - \sum_{j=1}^{i-1} c_{ij}y_j}{c_{ii}},\\
b_i'(w_1, \cdots, w_{i-1}) &= \frac{b_i - \sum_{j=1}^{i-1} c_{ij}y_j}{c_{ii}},\\
f_i(w_1, \cdots, w_{i-1}) &= \Phi(a_i'),\\
e_i(w_1, \cdots, w_{i-1}) &= \Phi(b_i').
\end{aligned}$$

3. Last, the multivariate normal probability in (4.54) may be written as

$$F(\boldsymbol{a}, \boldsymbol{b}) = (e_1 - f_1)\int_0^1 (e_2 - f_2)\cdots\int_0^1 (e_d - f_d)\mathrm{d}\boldsymbol{w}. \tag{4.55}$$

The integral (4.55) can be evaluated using different techniques, including Monte Carlo, quasi-Monte Carlo and adaptive methods. Two test problems are shown in the next section to test the time and error when evaluating the integral with different algorithms.

Now, consider the parameters defined as follows

$$a_1 = \cdots = a_d = -\infty, \tag{4.56a}$$

$$b_j \text{ IID uniformly on } [0, \sqrt{d}], \tag{4.56b}$$

$$\Sigma = (\sigma_{ij})_{d\times d}, \text{where } \sigma_{ij} = \begin{cases} 1 & i = j \\ \sigma & i \neq j \end{cases}, \tag{4.56c}$$

$$\sigma \text{ distributed uniformly on } [0, 1]. \tag{4.56d}$$

As the off diagonal elements of the covariance matrix are all σ and the lower limits of the integration are all $-\infty$, the analytic solution has the form,

$$F(-\infty, \boldsymbol{b}, \Sigma) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{t^2}{2}} \prod_{j=1}^{d} \left(\Phi\left(\left(b_j + \sqrt{\sigma} t \right) / \sqrt{1-\sigma} \right) \right) dt, \tag{4.57}$$

which could be evaluated by univariate quadrature techniques.

Numerical experiments are performed using 500 replications with dimension d IID uniformly chosen from 2 to 8, all other parameters are chosen according to (4.56). The error tolerance is $\max(10^{-3}, 10^{-4}|\mu|)$, where μ is computed by (4.57) using univariate quadrature function `integral` in MATLAB. The initial sample size for estimating the variance is 2^{10}, while the initial sample size using `cubSobol_g` and `cubLattice_g` are 2^{10} also. Time and error are recorded and plotted in Figure 4.4 using these three algorithms.

As we could see from Figure 4.4, `cubMC_g` used significantly more time than `cubSobol_g` and `cubLattice_g`, while 100% of the points satisfy the error tolerance.

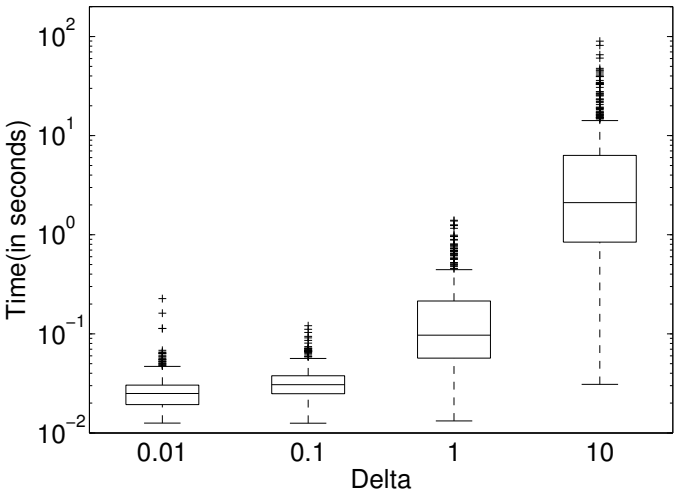

Figure 4.5. Box and whisker plot for execution time of integrating Multivariate Normal Probability function (4.54) with lower triangular matrix C (4.58) and other parameters (4.59). The horizonal axis denotes the different δ_t ranged from 0.01 to 10, the vertical axis denotes the computation time.

For `cubSobol_g`, 100% of the points satisfy the tolerance whereas 99.2% of the points satisfy the error tolerance for `cubLattice_g`.

4.5.4 Integrating Multivariate Normal Probability with a random covariance matrix. Another example with a random covariance matrix can be used to test the adaption of the algorithms. The way to construct the test function is described in the following steps.

First, generate a random lower triangular matrix C and covariance matrix Σ:

- Construct the lower triangular matrix of the form

$$C = \begin{pmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ \cdots & & & & \\ 0 & 0 & \cdots & 0 & \sigma_d \end{pmatrix} + \delta_t \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ x_{21} & 0 & 0 & \cdots & 0 \\ \cdots & & & & \\ x_{d1} & x_{d2} & \cdots & x_{dd} & 0 \end{pmatrix}, \qquad (4.58)$$

where $\log_{10}\sigma_i$ is IID uniform on $[1,3]$, δ_t is a shrinkage parameter, x_{ij} is IID uniform on $[-1,1]$.

- The lower triangular matrix C multiplied by its transpose C' will be the covariance matrix $\Sigma = CC'$.

Second, choose integration dimension and interval as follows:

$$d \text{ IID uniform on } [2,8] \tag{4.59a}$$

$$a_1 = \cdots = a_d = -\infty, \tag{4.59b}$$

$$b_j \text{ IID uniform on } \left[0, \sqrt{d}\right]. \tag{4.59c}$$

The computational time is recorded and plotted in Figure 4.5. 500 replications are performed with each of four different shrinkage factor $\delta_t = 0.01, 0.1, 1, 10$. The tolerance is $\max\left(10^{-5}, 10^{-5}\left|\mu\right|\right)$. As we can see, when δ_t gets large, the algorithm uses relatively more time. This plot shows the adaption of the algorithm `cubMC_g`, which, may adjust the sample or time required according the difficulty of the problem.

CHAPTER 5

GUARANTEED METHOD FOR BERNOULLI RANDOM VARIABLES

The Bernoulli trial is one of the simplest, yet most important, random processes. It takes two outcomes, namely, success or failure with a constant probability. Many real world problems can be modeled in terms of Bernoulli trials, i.e. estimating the probability of bankruptcy or a power failure.

A Bernoulli random variable, Y, takes two values, 0 and 1, with the probability $\Pr(Y = 1) = p$. Denote the distribution as $\mathrm{Ber}(p)$. Sometimes the process governing Y may have a complex form. This means that we may be able to generate IID Y_i, but not have a simple formula for computing p analytically.

As being discussed in Chapter 3 and 4, we present two algorithms for evaluating the mean of an arbitrary random variable based on the assumption that the kurtosis has a known upper bound. With knowledge of the distribution, we are looking into a better way to construct the fixed width confidence intervals for the Bernoulli random variables. Before presenting our algorithms, we first review some existing literature that based on the confidence interval construction of Bernoulli random variables.

Wald confidence interval [Agr02, Section 1.3.3] is a commonly used one for Bernoulli random variables based on maximum likelihood estimate. Unfortunately, it performs poorly when the sample size n is small or the true p is close to 0 or 1. Agresti [Agr02, Section 1.4.2] suggested constructing confidence intervals for binomial proportion by adding a pseudo-count of $z_{\alpha/2}/2$ successes and failures. Thus, the estimated mean will be $\tilde{p}_n = (n\hat{p}_n + z_{\alpha/2}/2)/(n + z_{\alpha/2})$. This method is also called adjusted Wald interval or Wilson score interval, since it was first discussed by E. B. Wilson [Wil27]. This method performs better than the Wald interval, however, it is

an approximate result and carries no guarantee.

Clopper and Pearson [CP34] suggested a tail method to calculate the exact confidence interval for a given sample size n and confidence level $1-\alpha$. Sterne [Ste54], Crow [Cro56], Blyth and Still [BS83] and Blaker [Bla00] proposed different ways to improve the exact confidence interval, however, all of them were only tested on small sample sizes, n. Moreover, there is no clue how to determine n with fixed half-width ε_a is given.

Hoeffding's Inequality in Theorem 2.7.1 provides a way to confidently estimate the probability p given the random variable is a Bernoulli one. We will use this inequality to construct the fixed width confidence interval for estimating Bernoulli random variables in the following sections.

In this chapter, we will present the ways to reliably construct a fixed width confidence interval for p with a guaranteed confidence level. That is, given an absolute error tolerance, ε_a, or a relative error tolerance, ε_r, and confidence level, $1-\alpha$, the algorithm determines the sample size, n, needed to compute the sample mean, $\hat{p}_n$, that satisfies the fixed width confidence interval condition $\Pr(|p-\hat{p}_n| \le \varepsilon_a) \ge 1-\alpha$ or $\Pr(|p-\hat{p}_n|/p \le \varepsilon_r) \ge 1-\alpha$.

5.1 Absolute error criterion

Theorem 2.7.1 motivates the following algorithm for constructing a fixed width confidence interval for an unknown p in terms of IID samples of $Y \sim \mathrm{Ber}(p)$.

Algorithm 5.1.1 (`meanMCBer_g` with absolute error criterion)**.** *Given an absolute error tolerance $\varepsilon_a > 0$ and an uncertainty level $\alpha \in (0,1)$, one generates*

$$n = N_{\mathrm{Hoeff}}(\varepsilon_a, \alpha) := \left\lceil \frac{\log(2/\alpha)}{2\varepsilon_a^2} \right\rceil \tag{5.1}$$

IID Bernoulli random samples of $Y \sim \mathrm{Ber}(p)$, and uses them to compute sample

mean:

$$\hat{p}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i. \tag{5.2}$$

Return $\hat{p}_n$ as the answer.

The following theorem ensures the success of the algorithm.

Theorem 5.1.2. *Algorithm 5.1.1 returns an answer, $\hat{p}_n$, that satisfies*

$$\Pr\left(|\hat{p}_n - p| \leq \varepsilon_a\right) \geq 1 - \alpha. \tag{5.3}$$

at a computational cost of $N_{\text{Hoeff}}(\varepsilon_a, \alpha)$ samples.

Proof. The definition of the sample size in (5.1) implies that

$$n = \left\lceil \frac{\log(2/\alpha)}{2\varepsilon_a^2} \right\rceil \geq \frac{\log(2/\alpha)}{2\varepsilon_a^2} \Rightarrow 1 - 2e^{-2n\varepsilon_a^2} \geq 1 - \alpha.$$

Applying Hoeffding's Inequality in Theorem 2.7.1 leads directly to (5.3). □

5.2 Relative error criterion

Given the relative error tolerance, ε_r, we are looking into constructing the fixed width confidence interval for p with the relative error criterion. By applying inequality (2.16) in Theorem 2.7.1, we have

$$\Pr\left(\frac{|\hat{p}_n - p|}{p} \leq \varepsilon_r\right) \geq 1 - 2\exp\left(-2np^2\varepsilon_r^2\right).$$

In order to make the right hand side of the inequality at least $1 - \alpha$, i.e.:

$$1 - 2\exp\left(-2np^2\varepsilon_r^2\right) \geq 1 - \alpha$$

the sample size n should satisfy the following condition:

$$n \geq -\frac{\log(\alpha/2)}{2p^2\varepsilon_r^2}.$$

Since p is unknown, the lower bound is needed to ensure n is large enough to bound the relative error. By applying one-side Hoeffding's Inequality (2.15) in Theorem 2.7.1, the following inequality holds:

$$\Pr\left(p \geq \hat{p}_n - \varepsilon\right) \geq 1 - \exp(-2n\varepsilon^2). \tag{5.4}$$

If the sample size

$$n \geq -\frac{\log(\alpha)}{2\varepsilon^2},$$

p has a conservative lower bound

$$\Pr(p \geq \hat{p}_n - \varepsilon) \geq 1 - \alpha.$$

Based on the argument above, an algorithm that estimates the means of Bernoulli random variables to a prescribed relative error tolerance is developed. The basic idea is to find a reliable lower bound of p at first, then estimate the p to the prescribed relative error tolerance.

Algorithm 5.2.1 (`meanMCber_g` with relative error criterion)**.** *The algorithm requires two inputs:*

- *the relative error tolerance,* $\varepsilon_r > 0$,
- *uncertainty level,* $\alpha \in (0, 1)$.

For $i = 1, 2, \cdots$, *do the following:*

1. *Compute the sample average* $\hat{p}_{n_i}$ *using*

$$n_i = \left\lceil \frac{-4^i \log\left(\alpha_i\right)}{2\varepsilon_r^2} \right\rceil \tag{5.5}$$

IID Bernoulli random samples, where

$$\alpha_i = 1 - (1 - \alpha/2)^{2^{-i}}. \tag{5.6}$$

2. (a) If

$$p_{n_i} \geq 3\varepsilon_r 2^{-i}, \tag{5.7}$$

set $\tau = i$ *and compute*

$$\hat{p}_L = \hat{p}_{n_\tau} - \varepsilon_r 2^{-\tau}, \tag{5.8}$$

a highly probable lower bound on p*. Go to step 3.*

(b) Else, set $i = i + 1$*, and go back to step 1.*

3. Compute $\hat{p} = \hat{p}_n$ *using*

$$n = \left\lceil \frac{\log(4/\alpha)}{2(\hat{p}_L \varepsilon_r)^2} \right\rceil \tag{5.9}$$

IID Bernoulli random samples that are independent of those used to compute $\hat{p}_{n_1}, \cdots, \hat{p}_{n_\tau}$*. Terminate the algorithm.*

If the algorithm terminates, then the relative error criterion is satisfied. The success of the algorithm is guaranteed in the following theorem.

Theorem 5.2.2. *Let* Y *be a Bernoulli random variable with mean* p*, it follows that Algorithm 5.2.1 using sample size* (5.9) *yields an estimate* $\hat{p}_n$ *that satisfies the fixed width confidence interval condition:*

$$\Pr\left(\frac{|\hat{p} - p|}{p} \leq \varepsilon_r\right) \geq 1 - \alpha. \tag{5.10}$$

Proof. At each step of iteration, the sample size is given in (5.5), i.e.

$$n_i \geq -\frac{4^i \log(\alpha_i)}{2\varepsilon_r^2},$$

which is equivalent to:

$$1 - \exp\left(-2n_i 4^{-i} \varepsilon_r^2\right) \geq 1 - \alpha_i. \tag{5.11}$$

Thus, by applying inequality (5.4), the following inequality holds:

$$\Pr\left(p \geq \hat{p}_{n_i} - 2^{-i}\varepsilon_r\right) \geq 1 - \exp\left(-2n_i 4^{-i}\varepsilon_r^2\right). \tag{5.12}$$

Combining (5.11) and (5.12) we get

$$\Pr\left(p \geq \hat{p}_{n_i} - 2^{-i}\varepsilon_r\right) \geq 1 - \alpha_i. \tag{5.13}$$

Similarly, the definition of sample size in (5.9) leads to the following inequality:

$$1 - 2\exp\left(-2(\hat{p}_L\varepsilon_r)^2\right) \geq 1 - \alpha/2. \tag{5.14}$$

Applying (2.16) in Theorem 2.7.1 yields:

$$\Pr\left(|\hat{p}_n - p| \leq \hat{p}_L\varepsilon_r\right) \geq 1 - 2\exp\left(-2(\hat{p}_L\varepsilon_r)^2\right). \tag{5.15}$$

Combining (5.14) and (5.15) yields:

$$\Pr\left(|\hat{p}_n - p| \leq \hat{p}_L\varepsilon_r\right) \geq 1 - \alpha/2. \tag{5.16}$$

Suppose the Algorithm 5.2.1 terminates at time τ, in which step, the relative error criterion

$$\Pr\left(\frac{|\hat{p} - p|}{p} \leq \varepsilon_r\right) \geq \Pr\left(|\hat{p}_n - p| \leq \hat{p}_L\varepsilon_r \;\;\&\;\; \hat{p}_L \leq p\right) \tag{5.17}$$

$$\geq \Pr\left(|\hat{p}_n - p| \leq \hat{p}_L\varepsilon_r \;\;\&\;\; \hat{p}_{n_i} - 2^{-i}\varepsilon_r \leq p \;\forall i \leq \tau\right) \tag{5.18}$$

$$= \Pr\left(|\hat{p}_n - p| \leq \hat{p}_L\varepsilon_r\right) + \prod_{i=1}^{\tau} \Pr\left(\hat{p}_{n_i} - 2^{-i}\varepsilon_r \leq p\right) - 1 \tag{5.19}$$

$$\geq (1 - \alpha/2) + \prod_{i=1}^{\tau}(1 - \alpha_i) - 1 \tag{5.20}$$

$$\geq (1 - \alpha/2) + \prod_{i=1}^{\infty}(1 - \alpha_i) - 1 \tag{5.21}$$

$$= 1 - \alpha. \tag{5.22}$$

The inequality (5.17) is true by adding an extra condition of a middle term $\hat{p}_L$, which makes the probability smaller. (5.18) is true by enlarging the set, letting all

the previous steps satisfying the lower bound condition. (5.19) is true by using the inequality $\Pr(A+B) \geq \Pr(A) + \Pr(B) - 1$ and the property of independence. (5.20) is true by applying the inequality (5.13) and (5.16). The final result is obtained by the way we construct α_i in(5.6). □

Next, we want to prove the computational cost of Algorithm 5.2.1 has an upper bound. As shown in Algorithm 5.2.1, the total cost is combined by two parts:

- For step 1 and 2, the cost is $n_i = \lceil -4^i \log(\alpha_i)/(2\varepsilon_r^2) \rceil$, for $i = 1 \cdots \tau$.
- For step 3, the cost is $n = \lceil -\log(\alpha/4)/(2(\hat{p}_L \varepsilon_r)^2) \rceil$.

Thus, the total cost can be written as:

$$n_{\text{tot}} = \sum_{i=1}^{\tau} \left\lceil -\frac{4^i \log(\alpha_i)}{2\varepsilon_r^2} \right\rceil + \left\lceil -\frac{\log(\alpha/4)}{2(\hat{p}_L \varepsilon_r)^2} \right\rceil, \tag{5.23}$$

where $\hat{p}_L$ is defined in (5.8). There are two random parts in n_{tot}, the stopping time τ and $\hat{p}_L$. In order to find the upper bound on cost, we need the upper bound on τ and lower bound on $\hat{p}_L$. Next, define the upper bound on the stopping time as

$$\bar{\tau} := \min\{i : p - \varepsilon_r 2^{-i} \geq 3\varepsilon_r 2^{-i}, i \in \mathbb{N}\} = \max(1, \lceil \log_2(4\varepsilon_r/p) \rceil), \tag{5.24}$$

and the upper bound on cost as:

$$n_{\text{up}} = \sum_{i=1}^{\bar{\tau}} \left\lceil -\frac{4^i \log(\alpha_i)}{2\varepsilon_r^2} \right\rceil + \left\lceil -\frac{\log(\alpha/4)}{2\varepsilon_r^4} \right\rceil. \tag{5.25}$$

Thus, we have the following theorem.

Theorem 5.2.3. *The cost of Algorithm 5.2.1, n_{tot}, has a probabilistic bound, n_{up}, that satisfying the following inequality*

$$\Pr(n_{\text{tot}} < n_{\text{up}}) \geq 1 - \alpha/2. \tag{5.26}$$

Proof. As the stopping criterion is:

$$\tau = \min\left\{i : \hat{p}_{n_i} \geq 3\varepsilon_r 2^{-i}, i \in \mathbb{N}\right\},$$

by the definition of $\bar{\tau}$ in (5.24), and applying one-side Hoeffding's Inequality, we have

$$\Pr(\tau \leq \bar{\tau}) \geq \Pr(\hat{p}_{n_i} \geq p - \varepsilon_r 2^{-i}, i \leq \tau) \geq \prod_{i=1}^{\infty}(1-\alpha_i) = 1 - \alpha/2. \tag{5.27}$$

Moreover, since the stopping time $\tau \geq 1$ and by the stopping criterion defined in (5.7), $\hat{p}_L$ has a lower bound, ε_r:

$$\Pr\left(\hat{p}_L \geq \varepsilon_r\right) = \Pr\left(\hat{p}_{n_\tau} \geq \varepsilon_r 2^{-\tau} + \varepsilon_r\right) \geq \Pr\left(\hat{p}_{n_\tau} \geq 3\varepsilon_r 2^{-\bar{\tau}}\right) = 1 \tag{5.28}$$

Combining (5.24) and (5.28), the bound is obtained,

$$\Pr(n_{\text{tot}} \leq n_{\text{up}}) \geq \Pr(\tau \leq \bar{\tau}, \hat{p}_L \geq \varepsilon_r) = \Pr(\tau \leq \bar{\tau}) = 1 - \alpha/2.$$

□

5.3 Numerical example for `meanMCber_g`

Algorithm 5.1.1 has been implemented in MATLAB, under the name `meanMCber_g` which is one the guaranteed algorithms in Guaranteed Automatic Integration Library (GAIL) [CDH+15].

5.3.1 Numerical example for `meanMCBer_g` with absolute error tolerance. To demonstrate its performance the algorithm `meanMCber_g` was run for 500 replications, each with a different p and ε_a, and with a fixed confidence level $1 - \alpha = 95\%$. The logarithms of p and ε_a were chosen independently and uniformly, namely

$$\log_{10} p \sim U[-3, -1], \qquad \log_{10} \varepsilon_a \sim U[-5, -2].$$

For each replication, the inputs ε_a and $\alpha = 5\%$ were provided to `meanMCber_g`, along with a Ber(p) random number generator, and the answer $\hat{p}_n$ was returned.

Figure 5.1 shows the ratio of the true error to the absolute error tolerance, $|p - \hat{p}_n| / \varepsilon_a$, for each of the 500 replications, plotted against p. All of the replications resulted in $|p - \hat{p}_n| \le \varepsilon_a$, which is better than guaranteed 95% confidence level. For some of these replications, `meanMCber_g` was asked to exceed its sample budget of 10^{10}, namely when $\varepsilon_a \le \sqrt{\log(2/0.05)} \times 10^{-5} = 3.69 \times 10^{-5}$.

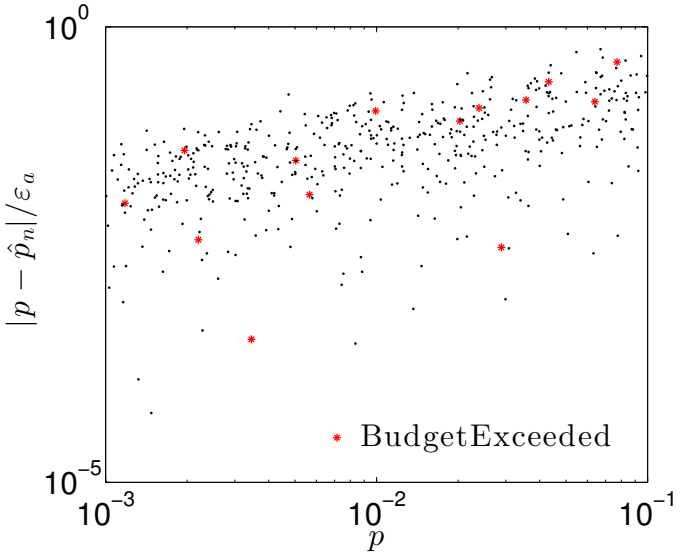

Figure 5.1. Ratio of the actual absolute error to the absolute error tolerance from `meanMCber_g` versus p, for different random samples of Ber(p) random variables.

While it is encouraging to see that `meanMCber_g` provides the correct answer in all cases, it is concerning that `meanMCber_g` is rather conservative for small p. This is due to the fact that the error of $\hat{p}_n$ using n samples is expected to be proportional to $\sqrt{\text{var}(Y)/n} = \sqrt{p(1-p)/n}$. Even though the error is small for small p, our algorithm does not take advantage of that fact. To do so will require at least a loose lower bound on p at the same time that the algorithm is trying to determine the sample size needed to estimate p carefully.

5.3.2 Numerical example for `meanMCBer_g` with absolute error tolerance. Using the sample parameters as in the previous example, we test the `meanMCBer_g` with the relative error tolerance, ε_r, which is random chosen with the distribution

$\log_{10}\varepsilon_r \sim U[-2,-1]$. The plot is given in Figure 5.2. Similar to the Figure 5.1,

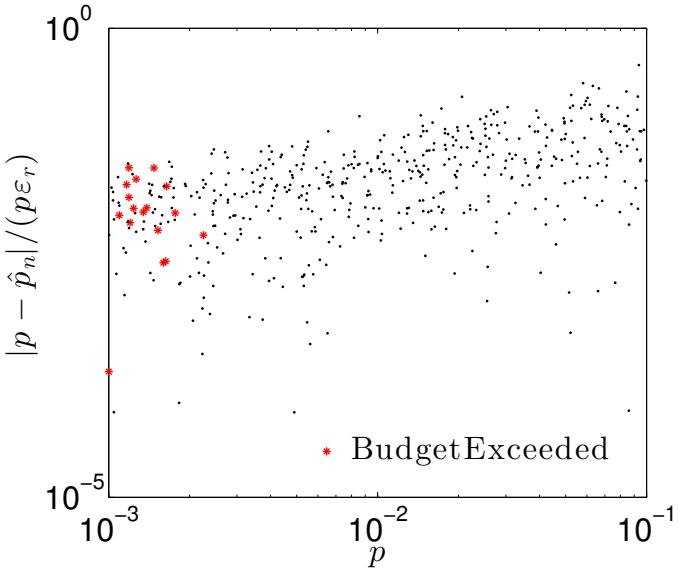

Figure 5.2. Ratio of the actual relative error to the relative error tolerance from `meanMCber_g` versus p, for different random samples of Ber(p) random variables.

100% of the points satisfy the relative error criterion although the confidence level is only 95%. Our algorithm is conservative and tends to use more samples. When p gets smaller, more points exceed the sample budget. The reason is that our algorithm determines the sample size based on the lower bound estimation of p, when p is small, this information makes the algorithm require more samples, hence, it is more likely to get over budget.

5.3.3 CLT & Hoeffding's Inequality Confidence Interval Cost Comparison.

By using Hoeffding's Inequality to construct guaranteed fixed-width confidence interval, we definitely incur additional cost compared to an approximate CLT confidence interval. The ratio of this cost is

$$\frac{N_{\text{Hoeff}}}{N_{\text{CLT}}} = \frac{\lceil \log(2/\alpha)/2\varepsilon_a^2 \rceil}{\lceil \Phi^{-1}(1-\alpha/2)/4\varepsilon_a^2 \rceil} \approx \frac{2\log(2/\alpha)}{\Phi^{-1}(1-\alpha/2)}. \tag{5.29}$$

This ratio essentially depends on the uncertainty level α and is plotted in Figure 5.3. For α between 0.01% to 10% this ratio is between 3.64 to 5.09, which we believe is a

reasonable price to pay for the added certainty of `meanMCber_g`.

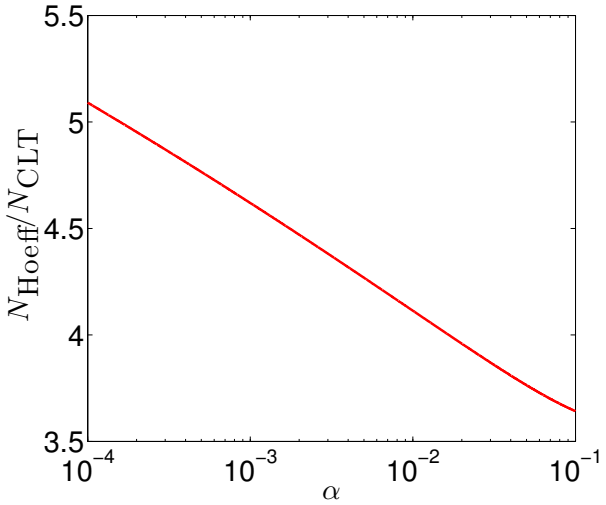

Figure 5.3. The computational cost ratio of using Hoeffding's Inequality and the CLT to construct a fixed-width confidence interval.

CHAPTER 6

GUARANTEED AUTOMATIC INTEGRATION LIBRARY

Guaranteed Automatic Integration Library (GAIL) [CDH+15] is a MATLAB toolbox created and developed by a group of researchers including Sou-Cheng T. Choi, Yuhan Ding, Fred J. Hickernell, Lan Jiang, Lluís Antoni Jiménez Rugama, Xin Tong, Yizhi Zhang and Xuan Zhou. It is a suite of algorithms including automatic and adaptive univariate function approximation (`funappx_g`), function minimization using linear splines (`funmin_g`), one-dimensional numerical integration using trapezoidal rule (`integral_g`) and mean estimation and numerical integration using Monte Carlo methods (`meanMC_g`,`cubMC_g` and `meanMCBer_g`). This purpose is to create a reliable and robust open source MATLAB package, following the philosophy of reproducible research championed by Claerbout [Cla10] and Donoho[BD95] and developing supportable scientific software that promote reliable reproducible research. In the next section, we will describe the structure of GAIL and the procedures by which we have been developing it.

6.1 Introduction to GAIL

Figure 6.1 shows the structure and hierarchy of GAIL. It contains several levels of directories.

- Documentations: It includes the scripts that generate the HTMLs that support and explain the algorithms.
- Papers: It includes the published papers related to GAIL.
- Algorithms: It is the core part of GAIL, which contains all the algorithms shown in Figure 6.1.
- Tests: It includes doctest and unittest for all the algorithms.

- Workouts: The practical examples and test functions used to test GAIL.
- Output Files: The output files generated from workouts and papers.
- Third Party: The useful third party tools including doctest package and chebfun package.

6.1.1 Downloads. GAIL can be downloaded from

`http://code.google.com/p/gail/`

Alternatively, you can get a local copy of the GAIL repository with this command:

```
git clone https://github.com/GailGithub/GAIL_Dev.git
```

You will need to install MATLAB 7 or a later version.

6.1.2 Documentation. Detailed documentations are available at GAIL_ Matlab/-Documentation directory.

6.1.3 Algorithms. GAIL Version 2.1 [CDH+15] includes the following eight algorithms:

- `funappx_g`: 1-D function approximation on a bounded interval with local adaption.
- `funmin_g`: Global univariate function minimization on a bounded interval.
- `integral_g`: 1-D numerical integration on a bounded interval based on Trapezoidal rule.
- `cubLattice_g`: Numerical integration via quasi-Monte Carlo method using rank-1 Lattices.

- `cubSobol_g`: Numerical integration via quasi-Monte Carlo method using Sobol points.
- `meanMC_g`: Monte Carlo method for estimating the mean of a random variable with a generalized error tolerance.
- `cubMC_g`: Monte Carlo method for the numerical integration with a generalized error tolerance.
- `meanMCBer_g`: Monte Carlo method for estimating the mean of a Bernoulli random variable.

6.1.4 Installation Instruction.

1. Unzip the contents of the zip file to a directory and maintain the existing directory and sub-directory structure. (Please note: If you install into the toolbox sub-directory of the MATLAB program hierarchy, you will need to click the button "Update toolbox path cache" from the File/Preferences... dialog in MATLAB.)
2. In MATLAB, add the GAIL directory to your path. This can be done by running `GAIL_Install.m`. Alternatively, this can be done by selecting "File/Set Path..." from the main or Command window menus, or with the command path tool. We recommend that you select the "Save" button on this dialog so that GAIL is on the path automatically in future MATLAB sessions.
3. To check if you have installed GAIL successfully, type `help meanMC_g` to see if its documentation shows up.

Alternatively, you could do this:

1. Download `DownloadInstallGail 2_1.m` and put it where you want GAIL to be installed.

2. Execute it in MATLAB.

To uninstall GAIL, execute `GAIL_Uninstall`.

6.1.5 Tests. Two tests are provided here, doctest and unittest, for each of the algorithms.

- To run doctests, simply call `doctest meanMC_g`
- To run unittest, call `run(ut_meanMC_g)`. Please note, unittest is only supported for MATLAB version 8 or later.

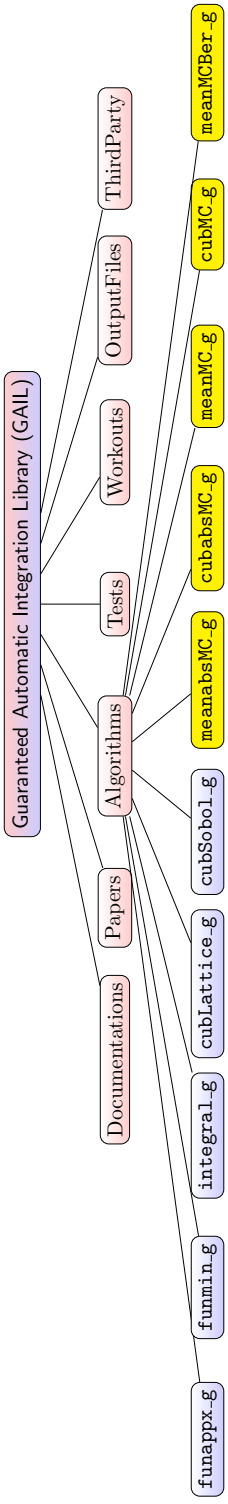

Figure 6.1. GAIL directory hierarchy

CHAPTER 7

CONCLUSION AND FUTURE WORK

Practitioners often use CLT based confidence intervals to construct Monte Carlo algorithms that estimating the mean of a random variable with the true variance been estimated by the sample variance, perhaps multiplied by some inflation factor. Often, this approach works, but it has no guarantee of success. This thesis suggests a trustworthy way to estimate the mean of a random variable to some absolute error tolerance, relative error tolerance, or some generalized error tolerance defined in Definition 2.9.2. Our algorithms guarantee that the answer will be true with probability at least 99% while CLT based algorithms do not carry such guarantee. Figure 3.3 compares the cost using the guaranteed algorithms and CLT based algorithm for the pure absolute error tolerance criterion. The cost ratio is the penalty for having a guaranteed algorithm.

Like any algorithm with guarantees, our algorithm does need to make assumptions about the random variable Y. We assume a known bound on the kurtosis of Y, either specified directly or implied by the user's choice of the sample size for estimating the variance, n_σ, and the SDIF, $\mathfrak{C}$. This is a philosophical choice. We prefer not to construct an algorithm that assumes a bound on the variance of Y, because such an algorithm would not be guaranteed for cY with $|c|$ large enough. If our algorithm works for Y, it will also work for cY, no matter how large $|c|$ is.

In practice the user may not know a priori if $\kappa \le \kappa_{\max}$ since it is even more difficult to estimate κ from a sample than it is to estimate σ^2. Thus, the choice of $\kappa_{\max}$ relies on the user's best judgment. Here are a few thoughts that might help. One might try a sample of typical problems for which one knows the answers and use these problems to suggest an appropriate $\kappa_{\max}$. Alternatively, one may think of $\kappa_{\max}$ not as a parameter to be prescribed, but as a reflection of the robustness of one's

Monte Carlo algorithm having chosen α_σ, n_σ and $\mathfrak{C}$.

There are a couple of areas that suggest themselves for further investigation. For the algorithms `meanMC_g`, the cost upper bound has been derived, however, it is loose. An tighter upper bound could be derived. Also, the lower bound derivation is another possible research area.

While `meanMCBer_g` in Algorithm 5.2.1 satisfies a relative error tolerance. One would expect the number of samples required to be proportional to $\mathrm{var}(Y)/(p\varepsilon_r)^2 \sim 1/(p\varepsilon_r^2)$ as $p\varepsilon_r \to 0$. Our attempts so far at using Hoeffding's inequality results in an algorithm with computational cost proportional to $1/(p\varepsilon_r)^2$ as $p\varepsilon_r \to 0$. Although, the algorithm for estimating the parameter p to some specified relative error tolerance is successful, the computational cost is extravagant for small p. The literature mentioned in the beginning of Chapter 5 may provide a clue to an algorithm with optimal cost. If this problem can be solved, then a natural extension would be to construct confidence intervals that satisfy either an absolute or relative error criterion, i.e., confidence intervals of the form $\Pr(|p - \hat{p}|\,/p \le \max(\varepsilon_a, \varepsilon_r p)) \ge 1 - \alpha$.

BIBLIOGRAPHY

[Agr02] Alan Agresti. *Categorical Data Analysis*. Wiley-Interscience, 2002.

[BD95] Jonathan B. Buckheit and David L. Donoho. *Wavelab and reproducible research*. Springer, 1995.

[Bla00] Helge Blaker. Confidence curves and improved exact confidence intervals for discrete distributions. *The Canadian Journal of Statistics*, 28(4):783–798, 2000.

[BS83] Colin R. Blyth and Harold A. Still. Binomial confidence intervals. *Journal of the American Statistical Association*, 78(381):108–116, 1983.

[CDH+13] N. Clancy, Y. Ding, C. Hamilton, F. J. Hickernell, and Y. Zhang. The cost of deterministic, adaptive, automatic algorithms: Cones, not balls. *Journal of complexity*, pages 21–45, 2013.

[CDH+15] Sou-Cheng T. Choi, Yuhan Ding, Fred J. Hickernell, Lan Jiang, Lluís Antoni Jiménez Rugama, Xin Tong, Yizhi Zhang, and Xuan Zhou. Gail: Guaranteed automatic integration library (version 2.1), matlab software, 2015.

[Che52] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[Cla10] Jon Claerbout. Reproducible computational research: A history of hurdles, mostly overcome, 2010.

[CP34] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 1934.

[CR65] Y. S. Chow and Herbert Robbins. On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *The Annals of Mathematical Statistics*, 36(2):457–462, 1965.

[Cro56] E.L. Crow. Confidence intervals for a proportion. *Biometrika*, 1956.

[DHT14] T. A Driscoll, N. Hale, and L. N. Trefethen. Chebfun guide, 2014.

[Fré35] Maurice Fréchet. Généralisation du théorème des probabilités totales. *Fundamenta Mathematicae*, 25(1):379–387, 1935.

[GB02] Alan Genz and Frank Bretz. Methods for the computation of multivariate t-probabilities. *Journal of Computational and Graphical Statistics*, 11(4), 2002.

[Gen92] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150, 1992.

[Hic14] Fred J. Hickernell. Koksma-hlawka inequality. *Wiley StatsRef: Statistics Reference Online*, 2014.

[HJ14] F. J. Hickernell and Ll. A. Jiménez Rugama. Reliable adaptive cubature using digital sequences, 2014. submitted for publication.

[HJLO14] F. J. Hickernell, L. Jiang, Y. Liu, and A. B. Owen. Guaranteed conservative fixed width confidence intervals via Monte Carlo sampling. *Monte Carlo and Quasi Monte Carlo Methods 2012*, pages 105–128, 2014.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, Mar 1963.

[JH14] Ll. A. Jiménez Rugama and F. J. Hickernell. Adaptive multidimensional integration based on rank-1 lattices, 2014. in preparation.

[JP04] Jean Jacod and Philip Protter. *Probability Essentials*. Springer, 2004.

[Kei96] B.D. Keister. Multidimensional quadrature algorithms. *Computers in Physics*, pages 119–122, 1996.

[LB10] Zhengyan Lin and Zhidong Bai. *Probability Inequalities*. Springer, 2010.

[MD96] Nitis Mukhopadhyay and Sujay Datta. On sequential fixed-width confidence intervals for the mean and second-order expansions of the associated coverage probabilities. *Annals of the Institute of Statistical Mathematics*, 48(3), 1996.

[NS13] Yu. S. Nefedova and I. G. Shevtsova. On nonuniform convergence rate estimates in the central limit theorem. *Theory of Probability and Its Applications*, 57(1):28–59, 2013.

[Sha08] Lawrence F. Shampine. Vectorized adaptive quadrature in Matlab. *Journal of Computational and Applied Mathematics*, 211:131–140, 2008.

[She14] I. G. Shevtsova. On absolute constants in inequalities of Berry-Esseen type. *Dokl. Akad. Nauk*, 456(6):650–654, 2014.

[Sie85] David Siegmund. *Sequential analysis: Tests and confidence intervals*. Springer, 1985.

[Ste45] Charles Stein. A two-sample test for a linear hypothesis whose power is independent of the variance. *The Annals of Mathematical Statistics*, 16(3):243–258, 1945.

[Ste54] T. E Sterne. Some remarks on confidence or fiducial limits. *Biometrika*, 1954.

[SW76] R. J. Serfling and D. D. Wackerly. Asymptotic theory of sequential fixed-width confidence interval procedures. *Journal of the American Statistical Association*, 71(356), 1976.

[Wil27] Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.