

# Monte Carlo Algorithms Where the Integrand Size is Unknown

Fred J. Hickernell

Department of Applied Mathematics  
Illinois Institute of Technology  
Email: [hickernell@iit.edu](mailto:hickernell@iit.edu), Web: [www.iit.edu/~hickernell](http://www.iit.edu/~hickernell)

Joint work with Lan Jiang, Yuewei Liu, and Art Owen

Thanks to Ian Sloan, Frances Kuo, Josef Dick, and Gareth Peters for inviting me.  
This work is partially supported by NSF-DMS-1115392 and NSF-DMS-1135257 grants.

Feb. 16, 2012



# Hypothetical Conversation

Practitioner

You, the Expert

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.



# Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .



# Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  **$n$** ?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .



# Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  **$n$** ?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .

As large as your computational budget allows.



# Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  $n$  to obtain  $|\mu - \hat{\mu}| \leq \varepsilon$ ?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .

The **Central Limit Theorem** says

$$n = \left\lceil \left( \frac{1.96\sigma}{\varepsilon} \right)^2 \right\rceil$$

where  $\sigma^2$  is the variance of the integrand.



# Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  $n$  to obtain  $|\mu - \hat{\mu}| \leq \varepsilon$ ?

How do I find  $\sigma^2$ ?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .

The **Central Limit Theorem** says

$$n = \left\lceil \left( \frac{1.96\sigma}{\varepsilon} \right)^2 \right\rceil$$

where  $\sigma^2$  is the variance of the integrand.



# Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  $n$  to obtain  $|\mu - \hat{\mu}| \leq \varepsilon$ ?

How do I find  $\sigma^2$ ?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i+n_\sigma}),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .

The **Central Limit Theorem** says

$$n = \left\lceil \left( \frac{1.96 \hat{\sigma}}{\varepsilon} \right)^2 \right\rceil$$

Try the **sample variance** times a **variance inflation factor**:

$$\hat{\sigma}^2 = \frac{n_\sigma}{n_\sigma - 1} \sum_{i=1}^{n_\sigma} [f(\mathbf{X}_i) - \hat{\mu}_\sigma]^2.$$





# Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  $n$  to obtain  $|\mu - \hat{\mu}| \leq \varepsilon$ ?

How do I find  $\sigma^2$ ?

Does theory **guarantee** that this algorithm works (at least 95% of the time)?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i+n_\sigma}),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .

The **Central Limit Theorem** says

$$n = \left\lceil \left( \frac{1.96\hat{\sigma}}{\varepsilon} \right)^2 \right\rceil$$

Try the **sample variance** times a **variance inflation factor**:

$$\hat{\sigma}^2 = \frac{n}{n-1} \sum_{i=1}^n [f(\mathbf{X}_i) - \hat{\mu}]^2.$$



## Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  $n$  to obtain  $|\mu - \hat{\mu}| \leq \varepsilon$ ?

How do I find  $\sigma^2$ ?

Does theory **guarantee** that this algorithm works (at least 95% of the time)?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i+n_\sigma}),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .

The **Central Limit Theorem** says

$$n = \left\lceil \left( \frac{1.96\hat{\sigma}}{\varepsilon} \right)^2 \right\rceil$$

Try the **sample variance** times a **variance inflation factor**:

$$\hat{\sigma}^2 = \frac{n}{n-1} \sum_{i=1}^n [f(\mathbf{X}_i) - \hat{\mu}]^2.$$

**Yes!** This algorithm, with minor modifications, carries a **limited warranty**. ☰ ↺ ↻ ↶ ↷



## Three Perspectives

$$\mu = E[f(\mathbf{X})] = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x} = ?$$

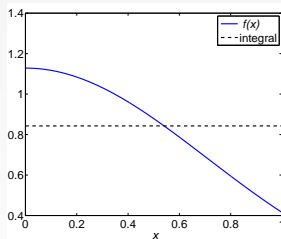
**Algorithm Design** Construct an automatic multivariate integrator analogous to MATLAB's **quad** for univariate integrals.

**Information-Based Complexity** Construct an algorithm,  $A$ , satisfying  $|\mu - A(f)| \leq \epsilon$  (definitely, with high probability, or on average) with  $\text{cost}(\epsilon, A, f)$  depending reasonably on  $\epsilon$  and the **unknown**  $\text{size}(f)$ .

**Statistics** Find a nonparametric confidence interval of **prescribed half-width**  $\epsilon$  for  $\mu$  from a reasonable number of samples  $Y_i = f(\mathbf{X}_i)$ .

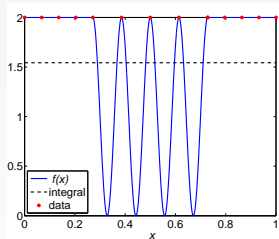


# MATLAB's Quadrature Routine quad Works Well, *but It Can Be Fooled*



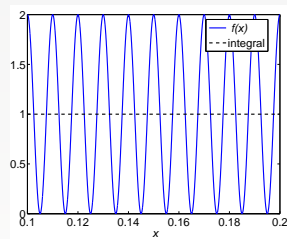
$$\frac{2}{\sqrt{\pi}} \int_0^1 e^{-x^2} dx = 0.8427007929497149$$

quad →  
0.8427007929497149 in  
0.160521 seconds.



$$\int_0^1 f(x) dx = 1.5436$$

but quad →  
2  
in 0.007092 seconds.



$$\int_0^1 [1 + \cos(200\pi x)] dx = 1$$

but quad →  
0.7636784919876782  
in 0.205272 seconds.



## Can We Have a **Guarantee** Like This?

For nice integrands,  $f$ , **quad** will provide  $\int_a^b f(x) dx$  with an error  $\leq \varepsilon$  in a reasonable amount of time, or your money back.

A nice integrand,  $f$ , satisfies the following conditions:

- ▶ ..., i.e., **quad** won't be fooled,
- ▶ ..., i.e., the number of function values required is moderate.

If  $f$  is not nice (nasty), then this guarantee is void, and **quad** may return an incorrect answer.



## An Impractical Guarantee

For integrands,  $f$ , satisfying  $\|f''\|_{\infty} \leq M$ , a trapezoidal rule with  $n = \sqrt{(b-a)^3 M / (12\varepsilon)}$  trapezoids will provide  $\int_a^b f(x) dx$  with an absolute error  $\leq \varepsilon$ .

To apply this guarantee, one must know  $M$  in advance, which is impractical. This is why **quad** (adaptive recursive Simpson's rule) estimates the error and **adaptively** determines the number of function evaluations,  $n$ .

If the algorithm works for  $f$ , it should normally work for  $cf$ .



## Recall Our Hypothetical Conversation

Practitioner

I need to evaluate **integrals**

$$\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x},$$

for many different  $f$ , where  $\rho$  is a given probability density function.

How large should I make  $n$  to obtain  $|\mu - \hat{\mu}| \leq \varepsilon$ ?

How do I find  $\sigma^2$ ?

Does theory **guarantee** that this algorithm works (at least 95% of the time)?

You, the Expert

Try a **sample average**,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i+n_\sigma}),$$

where the  $\mathbf{X}_i$  are i.i.d.  $\sim \rho$ .

The **Central Limit Theorem** says

$$n = \left\lceil \left( \frac{1.96\hat{\sigma}}{\varepsilon} \right)^2 \right\rceil$$

Try the **sample variance** times a **variance inflation factor**:

$$\hat{\sigma}^2 = \frac{n}{n-1} \sum_{i=1}^n [f(\mathbf{X}_i) - \hat{\mu}]^2.$$

**Yes!** This algorithm, with minor modifications, carries a **limited warranty**.



## Guarantee the Variance

The **sample variance**,  $v$  is an unbiased estimate of  $\sigma^2 = \int_{\mathbb{R}^d} [f(\mathbf{x}) - \mu]^2 \rho(\mathbf{x}) \, d\mathbf{x}$ .

$$v = \frac{1}{n_\sigma - 1} \sum_{i=1}^{n_\sigma} [f(\mathbf{X}_i) - \hat{\mu}_{n_\sigma}]^2, \quad \hat{\mu}_{n_\sigma} = \frac{1}{n_\sigma} \sum_{i=1}^{n_\sigma} f(\mathbf{X}_i), \quad \mathbf{X}_1, \mathbf{X}_2, \dots \text{ i.i.d. } \sim \rho$$

$$E[v] = \sigma^2, \quad \text{var}(v) = \frac{\sigma^4}{n_\sigma} \left( \kappa - \frac{n_\sigma - 3}{n_\sigma - 1} \right), \quad \kappa := \frac{\int_{\mathbb{R}^d} [f(\mathbf{x}) - \mu]^4 \rho(\mathbf{x}) \, d\mathbf{x}}{\sigma^4}$$

**Cantelli's Inequality** (Lin and Bai, 2010, 6.1e) guarantees that an inflated sample variance bounds the variance from above with uncertainty  $\tilde{\alpha}$ ,

$$\hat{\sigma}^2 := \mathfrak{C}^2 v, \quad \text{Prob}(\hat{\sigma}^2 \geq \sigma^2) \geq 1 - \tilde{\alpha}, \quad \mathfrak{C} > 1$$

provided that the **kurtosis** of the integrand,  $\kappa$ , is not too large, i.e.,

$$\kappa \leq \frac{n_\sigma - 3}{n_\sigma - 1} + \left( \frac{\tilde{\alpha} n_\sigma}{1 - \tilde{\alpha}} \right) \left( 1 - \frac{1}{\mathfrak{C}^2} \right)^2 =: \kappa_{\max}(\tilde{\alpha}, n_\sigma, \mathfrak{C}).$$



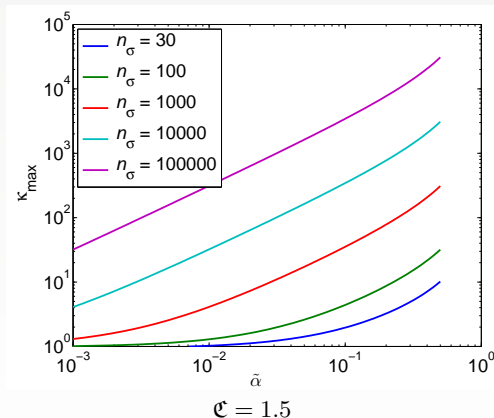


## Guarantee the Variance

$$\hat{\sigma}^2 = \frac{\mathfrak{C}^2}{n_\sigma - 1} \sum_{i=1}^{n_\sigma} [f(\mathbf{X}_i) - \hat{\mu}_{n_\sigma}]^2,$$

$$\text{Prob}(\hat{\sigma}^2 \geq \sigma^2) \geq 1 - \tilde{\alpha}$$

if  $\kappa \leq \kappa_{\max}(\tilde{\alpha}, n_\sigma, \mathfrak{C})$



## Guarantee the Integral (Mean)

The **Central Limit Theorem** gives an asymptotic result for fixed  $z \geq 0$ :

$$\text{Prob} \left[ \left| \underbrace{\int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x}}_{\mu} - \underbrace{\frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i+n_\sigma})}_{\hat{\mu}} \right| \leq \frac{z\sigma}{\sqrt{n}} \right] \rightarrow 1 - 2\Phi(-z) \quad \text{as } n \rightarrow \infty$$

A non-uniform **Berry-Esseen Inequality** (Petrov, 1995, Theorem 5.16, p. 168) gives a hard upper bound:

$$\text{Prob} \left[ |\mu - \hat{\mu}| \leq \frac{z\sigma}{\sqrt{n}} \right] \geq 1 - 2 \left( \Phi(-z) + \frac{0.56\kappa^{3/4}}{\sqrt{n}} (1 + |z|)^{-3} \right)$$

This guarantees that  $\text{Prob} [|\mu - \hat{\mu}| \leq \varepsilon] \geq 1 - \tilde{\alpha}$  if the sample size is large enough:

$$n \geq N_B(\varepsilon/\sigma, \tilde{\alpha}, \kappa) := \min \left\{ m \in \mathbb{N} : \Phi(-\varepsilon\sqrt{m}/\sigma) + \frac{0.56\kappa^{3/4}}{\sqrt{m}(1 + \varepsilon\sqrt{m}/\sigma)^3} \leq \frac{\tilde{\alpha}}{2} \right\}$$
$$\asymp \frac{\sigma^2}{\varepsilon^2} \quad \text{as } \frac{\varepsilon}{\sigma} \rightarrow 0$$



## cubMC

To evaluate  $\mu = \int_{\mathbb{R}^d} f(\mathbf{x}) \rho(\mathbf{x}) \mathrm{d}\mathbf{x}$

given input  $f, \rho, \varepsilon, \alpha, n_\sigma, \mathfrak{C}$ , and  $N_{\max}$ :

- ▶ Compute  $\tilde{\alpha} = 1 - \sqrt{1 - \alpha}$ , and the maximum kurtosis allowed,  $\kappa_{\max}(\tilde{\alpha}, n_\sigma, \mathfrak{C})$ .
- ▶ Overestimate the variance:  $\hat{\sigma}^2 = \frac{\mathfrak{C}^2}{n_\sigma - 1} \sum_{i=1}^{n_\sigma} [f(\mathbf{X}_i) - \hat{\mu}_{n_\sigma}]^2$ .
- ▶ Choose the new sample size,  $n = \min(\max(n_\sigma, N_B(\varepsilon/\hat{\sigma}, \tilde{\alpha}, \kappa_{\max})), N_{\max})$ , for the sample mean.
- ▶ Finally, compute the sample mean:  $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i+n_\sigma})$ .

Then  $\text{Prob}[|\mu - \hat{\mu}| \leq \varepsilon] \geq 1 - \alpha$  provided  $\kappa \leq \kappa_{\max}$  and  $n < N_{\max}$ .



## Guarantee the Time (Sample Size)

Cantelli's inequality also tells us that the estimated variance,  $\hat{\sigma}^2$ , will not overestimate the true variance,  $\sigma^2$ , by much, and so the number of function values needed is not unnecessarily large:

$$\begin{aligned}\text{cost}(\varepsilon, \text{cubMC}, \sigma) &= \sup_{f: \kappa \leq \kappa_{\max}} \min_N \{ \text{Prob}[n_\sigma + n \leq N] \geq 1 - \beta \} \\ &\leq n_\sigma + \max(n_\sigma, N_B(\varepsilon/(\sigma\gamma), \tilde{\alpha}, \kappa_{\max}^{3/4})) \asymp \frac{\sigma^2}{\varepsilon^2}, \\ \gamma &:= \mathfrak{C} \left\{ 1 + \sqrt{\left(\frac{\tilde{\alpha}}{1 - \tilde{\alpha}}\right) \left(\frac{1 - \beta}{\beta}\right) \left(1 - \frac{1}{\mathfrak{C}^2}\right)^2} \right\}^{1/2}.\end{aligned}$$

Cost depends on  $\sigma^2 = \text{var}(f)$ , but the algorithm **does not need to know**  $\sigma^2$ .



## Guarantee for cubMC

For nice integrands **cubMC** will provide the value of  $\mu = \int_{\mathbb{R}^d} f(x) \rho(x) dx$  with an absolute error of  $\leq \varepsilon$ , with probability  $1 - \alpha$ , in time  $\asymp (\sigma/\varepsilon)^2$  with probability  $1 - \beta$ , or your money back.

A nice integrand,  $f$ , satisfies the following conditions:

- ▶ the kurtosis is not too large, i.e.,  $\kappa \leq \kappa_{\max}(\tilde{\alpha}, n_{\sigma}, \mathcal{C})$ , and
- ▶ the variance is not overwhelming, i.e.,  $\sigma^2 \leq c\varepsilon^2 N_{\max}/d$ , where  $N_{\max}$  is the maximum number of scalar samples.

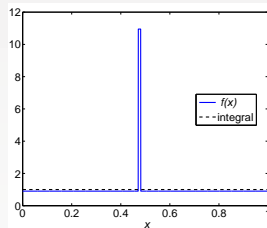
If  $f$  is not nice (nasty), **cubMC** may return the wrong answer.



## Peak Function — cubMC

$$f(x) = \begin{cases} 1 + \sigma \sqrt{\frac{1-p}{p}}, & 0 \leq x - z \pmod{1} \leq p, \\ 1 - \sigma \sqrt{\frac{p}{1-p}}, & p < x - z \pmod{1} \leq 1, \end{cases}$$

$$\mu = 1, \quad \kappa = \frac{1}{p(1-p)} - 3$$



$$z \sim U(0,1)$$

$$p \in [10^{-5}, 1/2], \quad \sigma \in [0.1, 10]$$

$$\log(p), \log(\sigma) \sim \text{Uniform}$$

$$\alpha = 5\%, \quad \mathfrak{C} = 1.5,$$

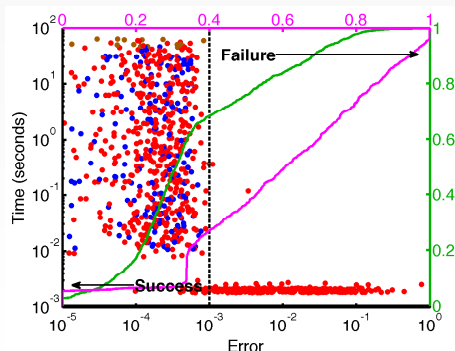
$$n_{\sigma} = 1024, \quad \varepsilon = 0.001$$

$$\kappa_{\max} = 9.2, \quad N_{\max} = 10^9$$

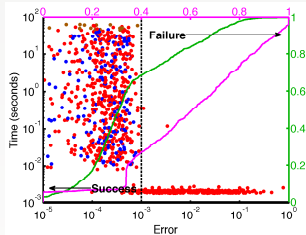
covered by guarantee

kurtosis too large

truncated sample



## Peak Function — cubMC vs. quad & quadgk



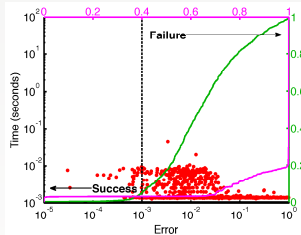
my cubMC

$\varepsilon = 0.001$

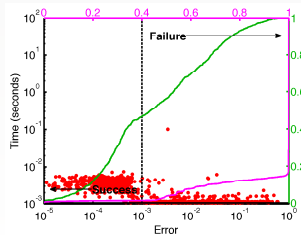
covered by guarantee

kurtosis too large

truncated sample



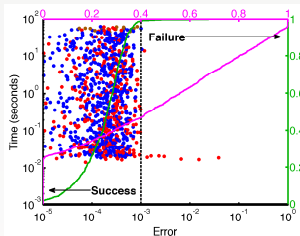
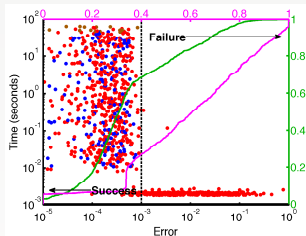
{ MATLAB's quad  
 $\varepsilon = 0.001$   
fast  
tolerance rarely met  
no guarantee



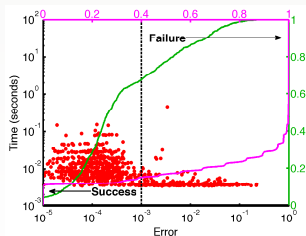
{ MATLAB's quadgk  
 $\varepsilon = 0.001$   
fast  
tolerance rarely met  
no guarantee



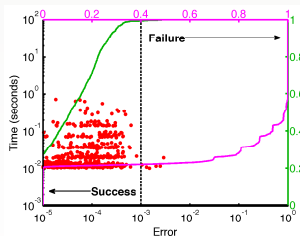
# Peak Function — cubMC i.i.d. vs. Sobol', Plus More Robustness



i.i.d. sampling  
covered by guarantee  
kurtosis too large  
truncated sample



$$n_{\sigma} = 1024, \kappa_{\max} = 9.2$$



Sobol' sampling  
no guarantee yet  
faster

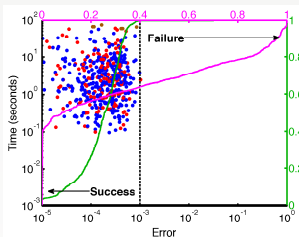
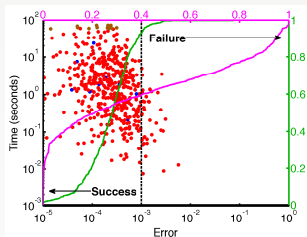
$$n_{\sigma} = 131072, \kappa_{\max} = 1050$$

$$\varepsilon = 0.001$$

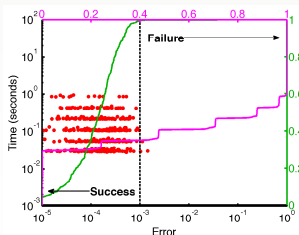
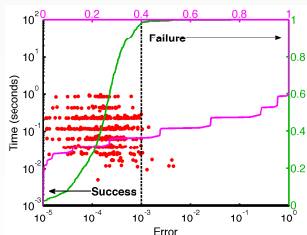




## Peak Function for $d = 3$



i.i.d. sampling  
covered by guarantee  
kurtosis too large  
truncated sample



Sobol' sampling  
quasi-standard error  
no guarantee yet  
faster

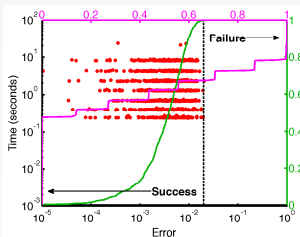
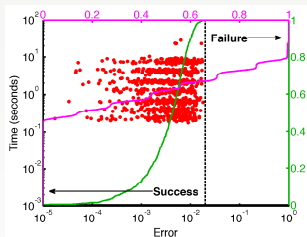
$$n_{\sigma} = 1024, \kappa_{\max} = 9.2$$

$$n_{\sigma} = 131072, \kappa_{\max} = 1050$$

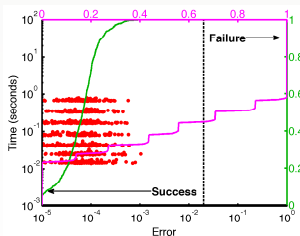
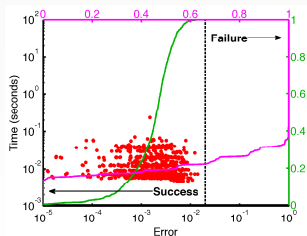
$$\varepsilon = 0.001$$



# Asian Geometric Mean Call, $d = 1, 2, 4, \dots, 64$



{ i.i.d. sampling



{ Sobol' sampling

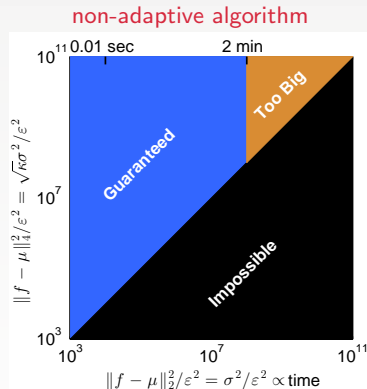
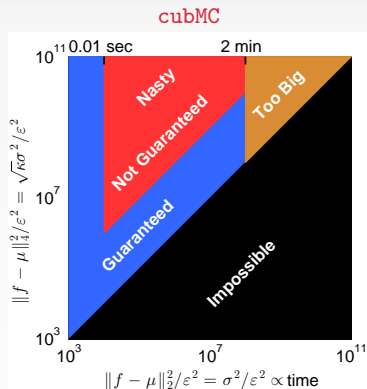
$$n_{\sigma} = 1024, \kappa_{\max} = 9.2$$

$$n_{\sigma} = 131072, \kappa_{\max} = 1050$$

$$\varepsilon = 0.02$$



## Why an Adaptive Algorithm?



- ▶ Cost depends on size of integrand
- ▶ Algorithm parameters determine robustness to nasty integrands
- ▶ Tiny integrands handled regardless
- ▶ Huge integrands cannot be handled

- ▶ Cost is fixed and high if you want it reach tolerance for lots of integrands
- ▶ Huge integrands cannot be handled



## Why Make cubMC Dependent on Kurtosis?

$$\sigma = \text{difficulty} \quad \kappa = \text{nastiness}$$

The kurtosis of a random variable or function,

$$\kappa = \frac{E[(Y - \mu)^4]}{\underbrace{\{E[(Y - \mu)^2]\}^2}_{\sigma^4}} \quad \text{or} \quad \kappa := \frac{\int_{\mathbb{R}^d} [f(\mathbf{x}) - \mu]^4 \rho(\mathbf{x}) \, d\mathbf{x}}{\underbrace{\left\{ \int_{\mathbb{R}^d} [f(\mathbf{x}) - \mu]^2 \rho(\mathbf{x}) \, d\mathbf{x} \right\}^2}_{\sigma^4}}$$

is difficult to estimate. Why should cubMC's guarantee depend on bounded  $\kappa$ ?

- ▶ Practically, we need  $\kappa$  bounded to justify the estimates of  $\sigma^2$ .
- ▶ Bounded  $\kappa$  yields sets of probability distributions or functions that are **non-convex**.
  - ▶ Nonparametric confidence intervals are **impossible** for **convex** sets of distributions (Bahadur and Savage, 1956, Corollary 2). ▶ How we break convexity
  - ▶ **Adaptive information does not help** for **convex** sets of integrands in the worst case and probabilistic settings (Traub et al, 1988, Chapter 4, Theorem 5.2.1; Chapter 8, Corollary 5.3.1). ▶ How we break convexity

## Quasi-Standard Error (Internal Replications) for Quasi-Random Sequences

Let  $\mathbf{X}_1, \mathbf{X}_2, \dots$  be a (random or deterministic) sequence, let  $r$  be fixed, and let

$$\hat{\mu}_m = \frac{1}{2^m} \sum_{i=1}^{2^m} f(\mathbf{X}_i) = \frac{1}{2^r} \sum_{j=1}^{2^r} \hat{\mu}_{m,j}, \quad \hat{\mu}_{m,j} = \frac{1}{2^{m-r}} \sum_{i=1}^{2^{m-r}} f(\mathbf{X}_{(j-1)2^{m-r}+i})$$

The **quasi-standard error** (Owen, 1997) measures the variation of among the means of parts of the whole sample

$$\text{qse}_m = \sqrt{\frac{1}{2^r(2^r - 1)} \sum_{j=1}^{2^r} (\hat{\mu}_{m,j} - \hat{\mu}_m)^2}$$

Given error tolerance,  $\varepsilon$ , and parameters  $r \in \mathbb{N}$ ,  $m_1 \geq r$ , and  $\mathfrak{C} > 1$ , for  $m = m_1, m_1 + 1, \dots$ ,

- ▶ Compute  $f(\mathbf{X}_{2^{m-1}+1}), \dots, f(\mathbf{X}_{2^m})$ , and  $\hat{\mu}_{m,1}, \dots, \hat{\mu}_{m,2^r}, \hat{\mu}_m$ .
- ▶ If  $\mathfrak{C} \text{qse}_m \leq \varepsilon$ , then stop. Else continue.



## Further Work

**Quasi-Monte Carlo Sampling** — What is a good measure of an integrand being nasty or nice?

**Variance Reduction Techniques** — Can we preserve the guarantee?

**Different Error Criteria** — Worst case? Randomized?

**Lower Bounds on Cost** — The typical fooling functions are nasty (high kurtosis). Does assuming moderate kurtosis make the problem easier?

**Relative Error Tolerances** — Both the variance and the mean are needed to determine the eventual sample size.

**Unbounded or Infinite  $d$**  — Can automatic integrators for finite  $d$  be used in multilevel methods to improve efficiency?

[▶ Look here](#)

**Other Problems** — Are there any guarantees for MATLAB's **quad**, or any other univariate adaptive quadrature routine that estimates error? What about guarantees for function approximation?



## References I

- Bahadur RR, Savage LJ (1956) The nonexistence of certain statistical procedures in nonparametric problems. *Ann Math Stat* 27:1115–1122
- Halton JH (2005) Quasi-probability: Why quasi-Monte-Carlo methods are statistically valid and how their errors can be estimated statistically. *Monte Carlo Methods and Appl* 11:203–350
- Lin Z, Bai Z (2010) *Probability Inequalities*. Science Press and Springer-Verlag, Beijing and Berlin
- Owen AB (1997) Scrambled net variance for integrals of smooth functions. *Ann Stat* 25:1541–1562
- Owen AB (2006) On the Warnock-Halton quasi-standard error. *Monte Carlo Methods and Appl* 12:47–54
- Petrov VV (1995) *Limit Theorems of Probability Theory: Sequences of Independent Random Variables*. Clarendon Press, Oxford
- Snyder WC (2000) Accuracy estimation for quasi-Monte Carlo simulations. *Math Comput Simulation* 54:131–143
- Traub JF, Wasilkowski GW, Woźniakowski H (1988) *Information-Based Complexity*. Academic Press, Boston



# A Set of Distributions with Bounded Kurtosis is Non-Convex

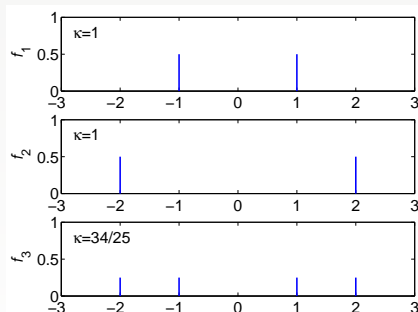
$$\text{Prob}(Y_1 = y) = 0.5, \quad y = \pm 1$$

$$\text{Prob}(Y_2 = y) = 0.5, \quad y = \pm 2$$

$$f_3 = \frac{1}{2}f_1 + \frac{1}{2}f_2$$

$$\text{Prob}(Y_3 = y) = 0.25, \quad y = \pm 1, \pm 2$$

$$\kappa_3 = \frac{34}{25} > 1 = \kappa_1 = \kappa_2$$

[Return](#)



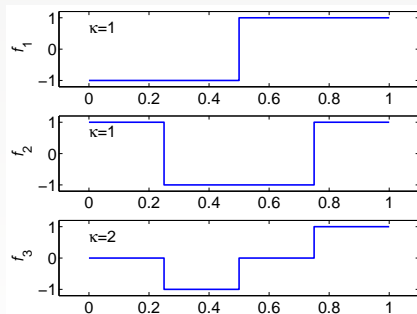
## A Set of Integrands with Bounded Kurtosis is Non-Convex

$$f_1(x) = \begin{cases} -1, & 0 \leq x < 1/2 \\ 1, & 1/2 \leq x \leq 1 \end{cases}$$

$$f_2(x) = \begin{cases} 1, & 0 \leq x < 1/4 \\ -1, & 1/4 \leq x \leq 3/4 \\ 1, & 3/4 \leq x \leq 1 \end{cases}$$

$$\begin{aligned} f_3(x) &= \frac{1}{2}f_1(x) + \frac{1}{2}f_2(x) \\ &= \begin{cases} 0, & 0 \leq x < 1/4, \\ -1, & 1/4 \leq x \leq 1/2, \\ 0, & 1/2 \leq x \leq 3/4, \\ 1, & 3/4 \leq x \leq 1, \end{cases} \end{aligned}$$

$$\kappa_3 = 2 > 1 = \kappa_1 = \kappa_2$$

[Return](#)

## When Does Quasi-Standard Error Work?

Quasi-standard error has been proposed by Warnock and studied by Owen (1997); Snyder (2000); Halton (2005); Owen (2006). Suppose that  $\mathbf{X}_1, \mathbf{X}_2, \dots$  is a scrambled digital  $(t, d)$ -sequence in base 2,  $P_m = \{\mathbf{X}_1, \dots, \mathbf{X}_{2^m}\}$ , and the integrand can be expanded in a Walsh series:

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) e^{\pi \sqrt{-1} \mathbf{k} \otimes \mathbf{x}}, \quad (S_B f)(\mathbf{x}) := \sum_{\mathbf{k} \in B} \hat{f}(\mathbf{k}) e^{\pi \sqrt{-1} \mathbf{k} \otimes \mathbf{x}} \quad (\text{filtered } f)$$

$$\mu = \int_{[0,1]^d} f(\mathbf{x}) \, d\mathbf{x} = S_{\{\mathbf{0}\}} f, \quad \hat{\mu}_m = \frac{1}{2^m} \sum_{i=1}^{2^m} f(\mathbf{X}_i) = (S_{P_m^\perp} f)(\mathbf{X}_1),$$

where  $\otimes$  is a bitwise dot product modulo 2, and  $P_m^\perp = \{\mathbf{k} \in \mathbb{N}_0^d : \mathbf{k} \otimes \mathbf{x} = 0 \, \forall \mathbf{x} \in P_m\}$  is the **dual net** (wavenumbers aliased with  $\mathbf{0}$ ). The **quasi-standard error** may be expressed as

$$\text{qse}_m = \sqrt{\frac{1}{2^r - 1} \sum_{\mathbf{l} \in (P_m^\perp - \mathbf{r} / P_m^\perp) \setminus \{\mathbf{0}\}} (S_{P_m^\perp \oplus \mathbf{l}} f)^2(\mathbf{X}_1)}$$

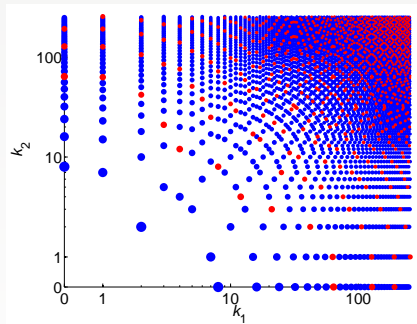
which is a surrogate for  $|\mu - \hat{\mu}_m| = |(S_{P_m^\perp \setminus \{\mathbf{0}\}} f)(\mathbf{X}_1)|$ .



# When Does Quasi-Standard Error Work?

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) e^{\pi \sqrt{-1} \mathbf{k} \otimes \mathbf{x}}$$

$$|\mu - \hat{\mu}_m| = \left| (S_{P_m^\perp \setminus \{\mathbf{0}\}} f)(\mathbf{X}_1) \right| \quad \bullet P_6^\perp \setminus \{\mathbf{0}\}$$



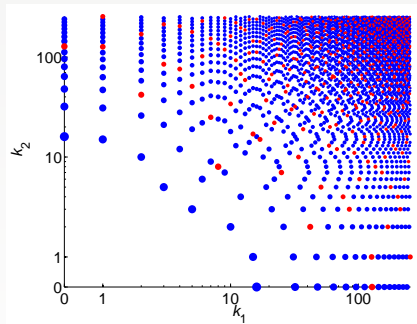
$$\text{qse}_m = \sqrt{\frac{1}{7} \sum_{\mathbf{l} \in (P_{m-3}^\perp / P_m^\perp) \setminus \{\mathbf{0}\}} (S_{P_m^\perp \oplus \mathbf{l}} f)^2(\mathbf{X}_1)} \quad \bullet P_3^\perp \setminus P_6^\perp$$



# When Does Quasi-Standard Error Work?

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) e^{\pi \sqrt{-1} \mathbf{k} \otimes \mathbf{x}}$$

$$|\mu - \hat{\mu}_m| = \left| (S_{P_m^\perp \setminus \{\mathbf{0}\}} f)(\mathbf{X}_1) \right| \quad \bullet P_7^\perp \setminus \{\mathbf{0}\}$$

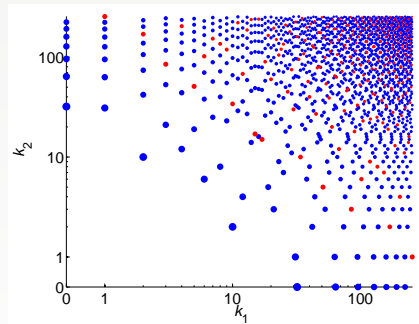


$$\text{qse}_m = \sqrt{\frac{1}{7} \sum_{\mathbf{l} \in (P_{m-3}^\perp / P_m^\perp) \setminus \{\mathbf{0}\}} (S_{P_m^\perp \oplus \mathbf{l}} f)^2(\mathbf{X}_1)} \quad \bullet P_4^\perp \setminus P_7^\perp$$

# When Does Quasi-Standard Error Work?

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) e^{\pi \sqrt{-1} \mathbf{k} \otimes \mathbf{x}}$$

$$|\mu - \hat{\mu}_m| = \left| (S_{P_m^\perp \setminus \{\mathbf{0}\}} f)(\mathbf{X}_1) \right| \quad \bullet P_8^\perp \setminus \{\mathbf{0}\}$$



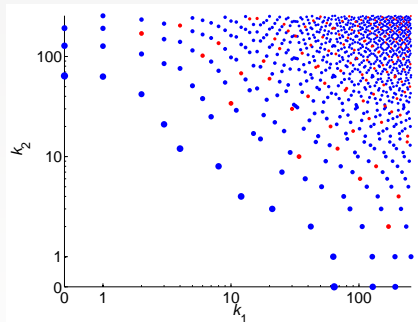
$$\text{qse}_m = \sqrt{\frac{1}{7} \sum_{\mathbf{l} \in (P_{m-3}^\perp / P_m^\perp) \setminus \{\mathbf{0}\}} (S_{P_m^\perp \oplus \mathbf{l}} f)^2(\mathbf{X}_1)} \quad \bullet P_5^\perp \setminus P_8^\perp$$



# When Does Quasi-Standard Error Work?

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) e^{\pi \sqrt{-1} \mathbf{k} \otimes \mathbf{x}}$$

$$|\mu - \hat{\mu}_m| = \left| (S_{P_m^\perp \setminus \{\mathbf{0}\}} f)(\mathbf{X}_1) \right| \quad \bullet P_9^\perp \setminus \{\mathbf{0}\}$$



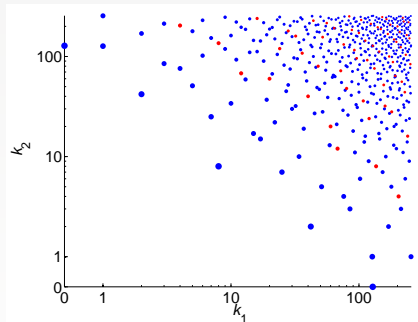
$$\text{qse}_m = \sqrt{\frac{1}{7} \sum_{\mathbf{l} \in (P_{m-3}^\perp / P_m^\perp) \setminus \{\mathbf{0}\}} (S_{P_m^\perp \oplus \mathbf{l}} f)^2(\mathbf{X}_1)} \quad \bullet P_6^\perp \setminus P_9^\perp$$

[Return](#)


# When Does Quasi-Standard Error Work?

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) e^{\pi \sqrt{-1} \mathbf{k} \otimes \mathbf{x}}$$

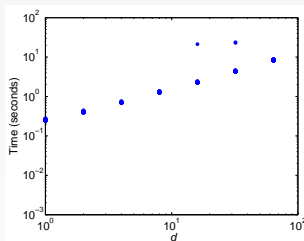
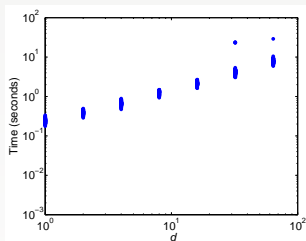
$$|\mu - \hat{\mu}_m| = \left| (S_{P_m^\perp \setminus \{\mathbf{0}\}} f)(\mathbf{X}_1) \right| \quad \bullet P_{10}^\perp \setminus \{\mathbf{0}\}$$



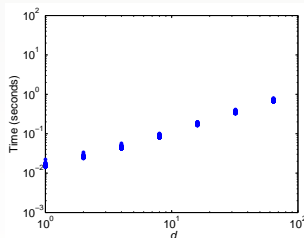
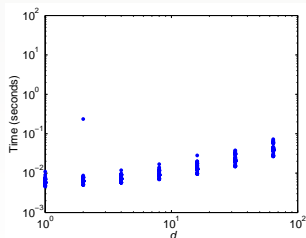
$$\text{qse}_m = \sqrt{\frac{1}{7} \sum_{\mathbf{l} \in (P_{m-3}^\perp / P_m^\perp) \setminus \{\mathbf{0}\}} (S_{P_m^\perp \oplus \mathbf{l}} f)^2(\mathbf{X}_1)} \quad \bullet P_7^\perp \setminus P_{10}^\perp$$

[Return](#)


## Asian Geometric Mean Call Execution Times



{ i.i.d. sampling



{ Sobol' sampling

 $n_\sigma = 1024, \kappa_{\max} = 9.2$  $n_\sigma = 131072, \kappa_{\max} = 1050$ 

Return