Adaptive Quasi-Monte Carlo Integration

June 4th, 2002

Rudolf Schrer

The Problem

Given a function

$$f\colon C_s \to \mathbb{R},$$

with $C_s = [0,1]^s \subset \mathbb{R}^s$ denoting the s-dimensional unit cube.

Calculate an approximation $\mathrm{Q}f$ for the multi-variate integral

$$\mathrm{I}f := \int_{C_s} f(\boldsymbol{x})\,d\boldsymbol{x}.$$

$\mathrm{Q}f$ has to be based on f-evaluations at n points $\boldsymbol{x}_i \in C_s$ which can be chosen arbitrarily by the integration routine.
Therefore, $\mathrm{Q}f$ will be of the form

$$\mathrm{Q}_n f = \sum_{i=1}^{n} w_i\, f(\boldsymbol{x}_i).$$

Monte Carlo Integration

$$\mathrm{Q}_n f := \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i)$$

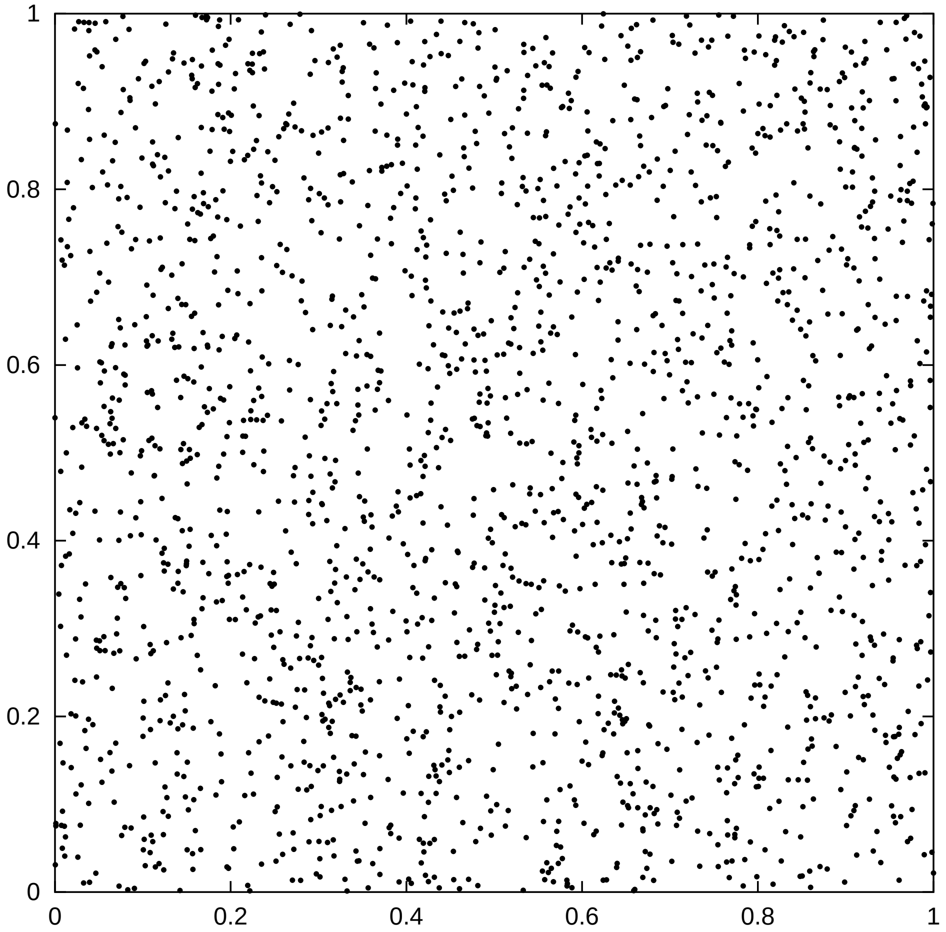
with $\boldsymbol{x}_i$ random samples uniformly distributed in C_s.

- $|\mathrm{I}f - \mathrm{Q}_n f| \approx \frac{\sigma(f)}{\sqrt{n}} = \sqrt{\frac{Var(f)}{n}} = \mathcal{O}(n^{-1/2})$
- Independent of dimension s
- $\sigma(f)$ behaves well for a huge class of integrands
- We can even estimate the accuracy:

$$|\mathrm{I}f - \mathrm{Q}_n f| \approx \sqrt{\frac{Var(f)}{n}} \approx \sqrt{\frac{\sum f^2(\boldsymbol{x}_i) - \frac{1}{n}\left(\sum f(\boldsymbol{x}_i)\right)^2}{n(n-1)}}$$

- $\Rightarrow$ MC integration is a pretty foolproof way to estimate an integral

Can we do better?



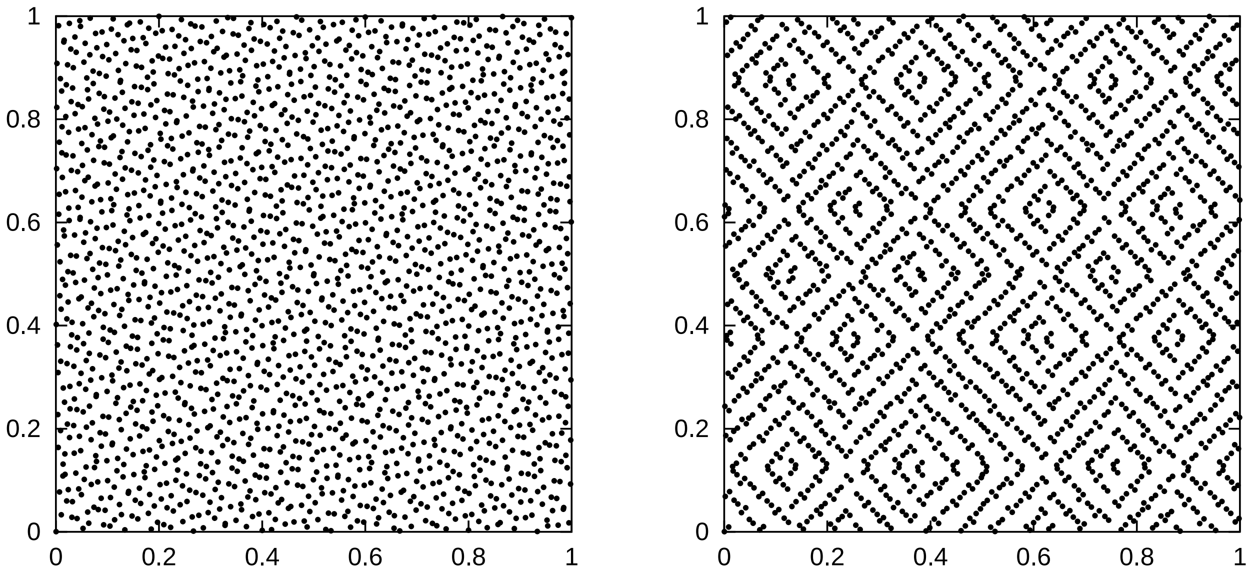
- Random points are evenly distributed in any dimension

- However, random clusters and gaps appear

- Are there high-dimensional, evenly distributed, but regular point sets?

Quasi-Monte Carlo

- Instead of drawing random samples, use low discrepancy point-sets like (t, m, s)-nets!



The first 2048 points from the Sobol sequence

$x_1 x_3$-projection (left), $x_{37} x_{40}$-projection (right)

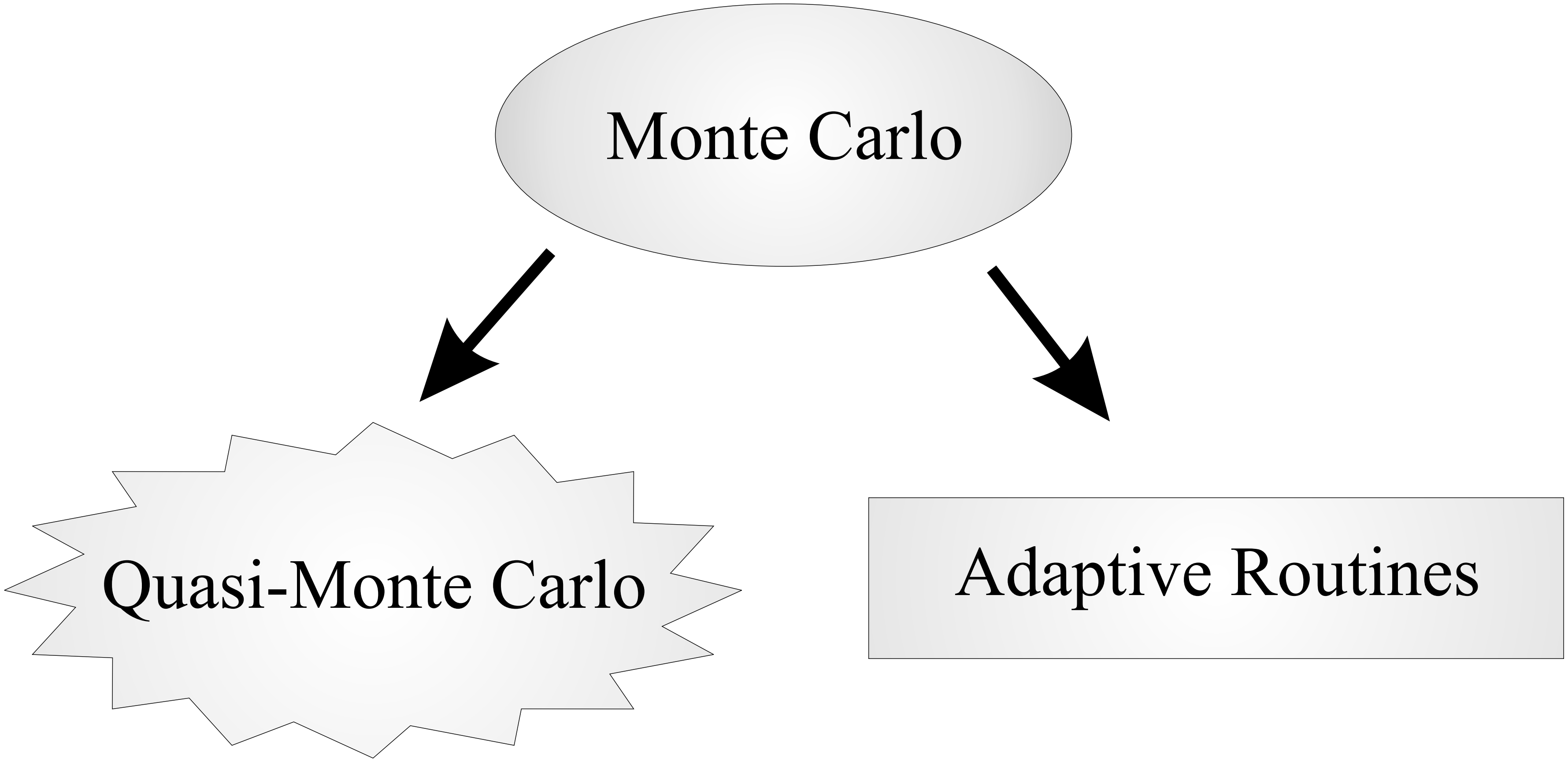
Performance of Quasi-Monte Carlo

- Koksma-Hlawka inequality:

$$|\mathrm{I}f - \mathrm{Q}_n f| \leq V(f) \cdot D_n^* \leq c \, \frac{\log^s n}{n}$$

  - Only an upper bound, no estimator
  - $V(f) = \infty$ even for simple integrands
  - No general method for estimating $V(f)$
  - $\log^s n$ is huge for affordable n

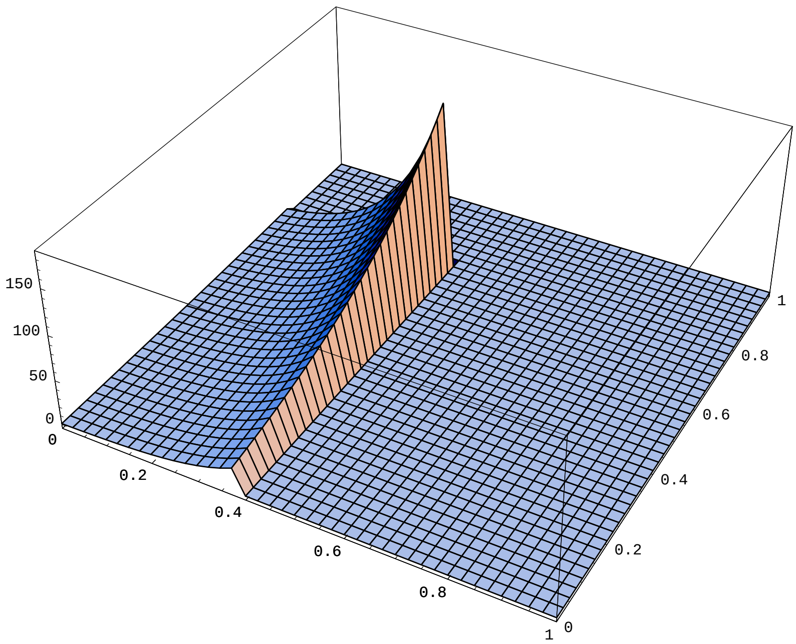
- However, it works quite well in practice: $|\mathrm{I}f - \mathrm{Q}_n f| \approx \mathcal{O}(n^{-1})$ is usually obtained!

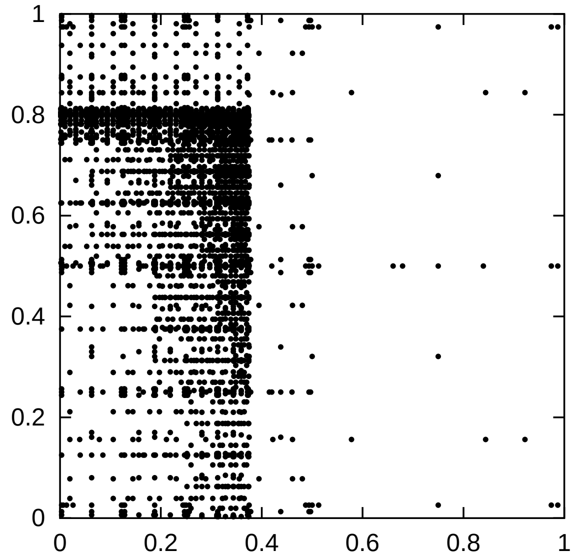

Monte Carlo
Quasi-Monte Carlo
Adaptive Routines

Adaptive Integration

Algorithm 1 Adaptive Integration

Put C_s into region collection
while estimated error too large **do**
  Choose subregion with large error
  Split region
  Apply basic rule
  Store new regions in region collection
end while





Stratified Sampling

Taking $n/2$ samples from two halfs of C_s is always better than sampling C_s with n points!

$$\tilde{Q}_n f = \frac{1}{2}\left(Q_{n/2} f_\alpha + Q_{n/2} f_\alpha\right)$$

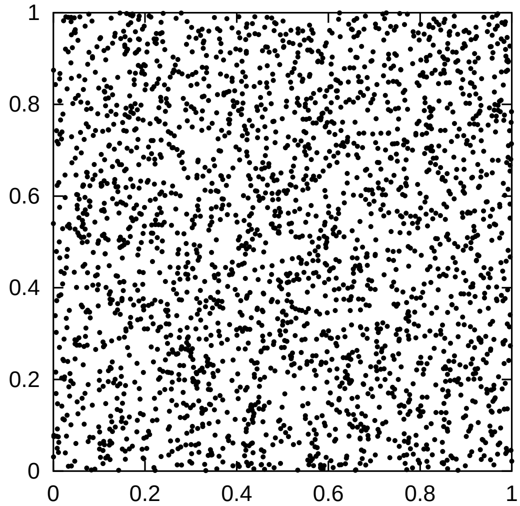
with f_α and f_β denoting f restricted to the left and right subcube.

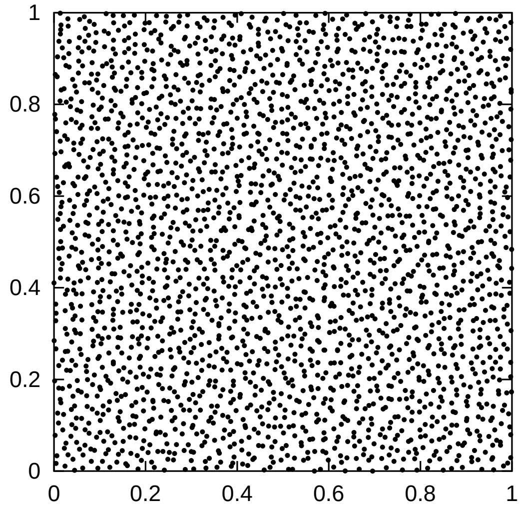
Variance of this estimator:

$$\begin{aligned}
Var(\tilde{Q}_n f) &= \frac{1}{4}\left(Var\left(Q_{n/2} f_\alpha\right) + Var\left(Q_{n/2} f_\beta\right)\right) \\
&\approx \frac{1}{4}\left(\frac{Var(f_\alpha)}{n/2} + \frac{Var(f_\beta)}{n/2}\right) \\
&= \frac{1}{2n}\left(Var(f_\alpha) + Var(f_\beta)\right) \\
&= \frac{1}{n} \cdot \frac{Var(f_\alpha) + Var(f_\beta)}{2} \\
&\leq \frac{1}{n} \cdot Var(f) = Var(Q_n f)
\end{aligned}$$

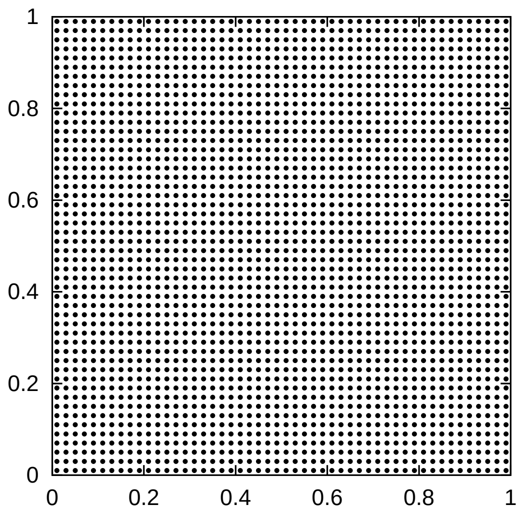
Recursive Stratified Sampling

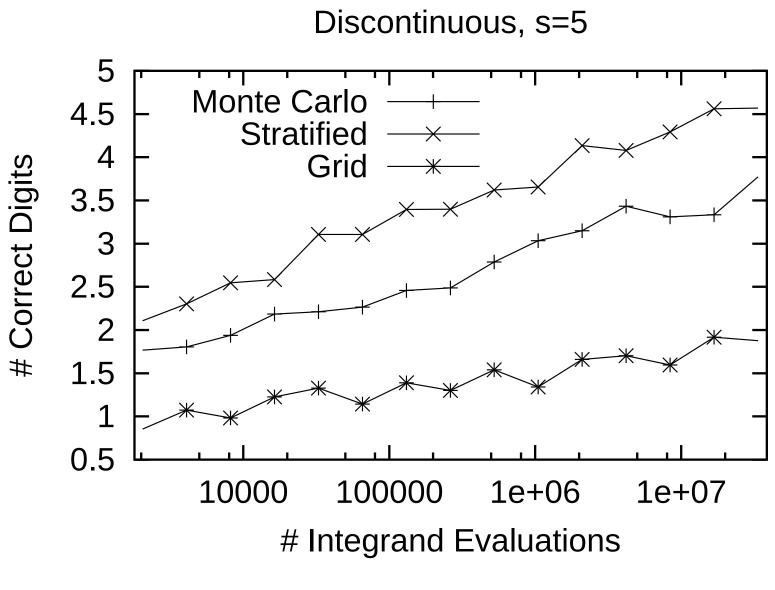
Stratification can be done recursively, leading to n subcubes with one random point in each of them.





This comes close to a grid.
However, randomization performs much better!





MISER – Adaptive Stratification

Stratification improves performance whenever

$$\sigma(f_\alpha) \neq \sigma(f_\beta).$$

The optimal performance can be achieved by allocating points such that

$$n_\alpha / n_\beta = \sigma(f_\alpha)/\sigma(f_\beta).$$

This lead directly to the following adaptive algorithm:

Algorithm 2 MISER

1: Allocate points for presampling
2: Estimate $\sigma_{\alpha i}$ and $\sigma_{\beta i}$ for all $i = 1, \ldots, s$ halfs
3: Choose split dimension
4: Assign point budgets N_α and N_β
5: Apply MISER to both subcubes
6: Calculate final estimate

Importance Sampling

- Integration error depends on $Var(f)$

- What if
  - We have positive-valued function p with
    $$\int_{C_s} p(\boldsymbol{x})\, d\boldsymbol{x} = 1,$$
    i. e. p is a probability density function
  - p mimics f such that $p \simeq |f|$

- Then
  - f/p has a very low variance
  - $$\int_{C_s} f(\boldsymbol{x})\, d\boldsymbol{x} = \int_{C_s} \frac{f(\boldsymbol{x})}{p(\boldsymbol{x})}\, dP(\boldsymbol{x}),$$
    i. e. the sample mean of f/p with density p equals sample mean of f with density 1.

Recipe:

Find a pdf p with

- $p \simeq |f|$
- We can generate p-distributed random numbers

Adaptive Importance Sampling

Algorithm 3 Adaptive Importance Sampling

```
Start with p ≡ 1/ vol C_s
for i = 1, ..., m do
    Sample f/p to refine p
end for
Use remaining points to sample f/p with density p
```

Algorithms differ by the available functions p and by the way they are estimated.

VEGAS

VEGAS uses a product of piecewise constant, one-dimensional functions.

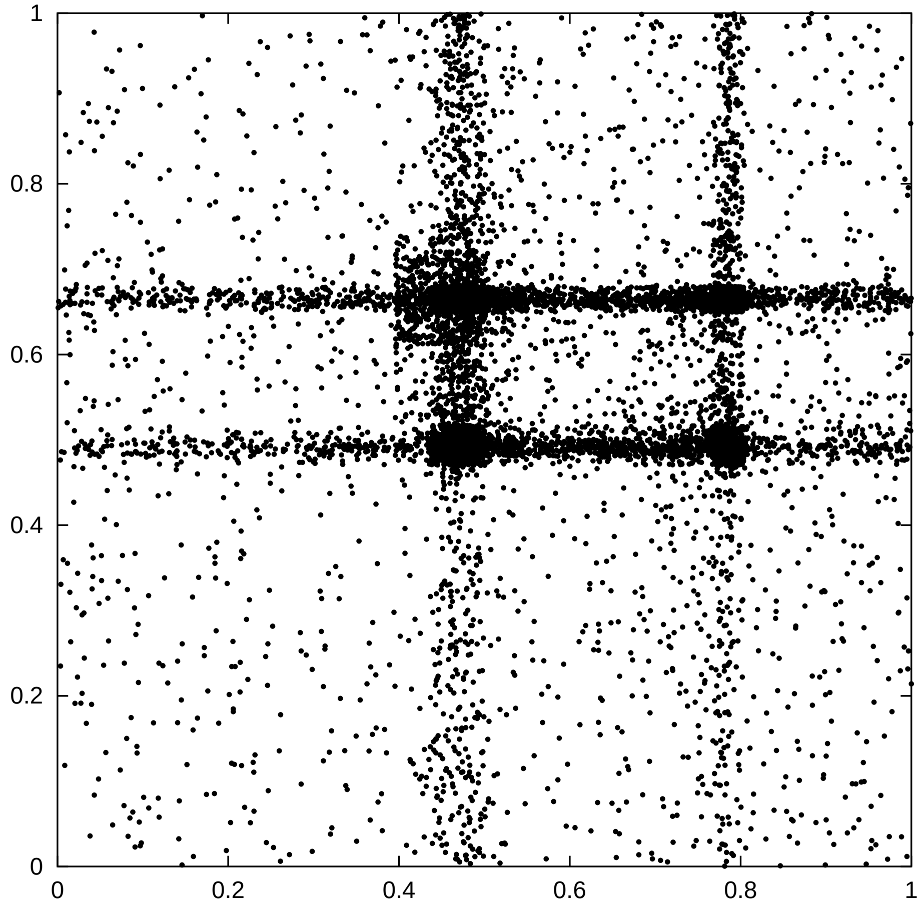
Control Variates

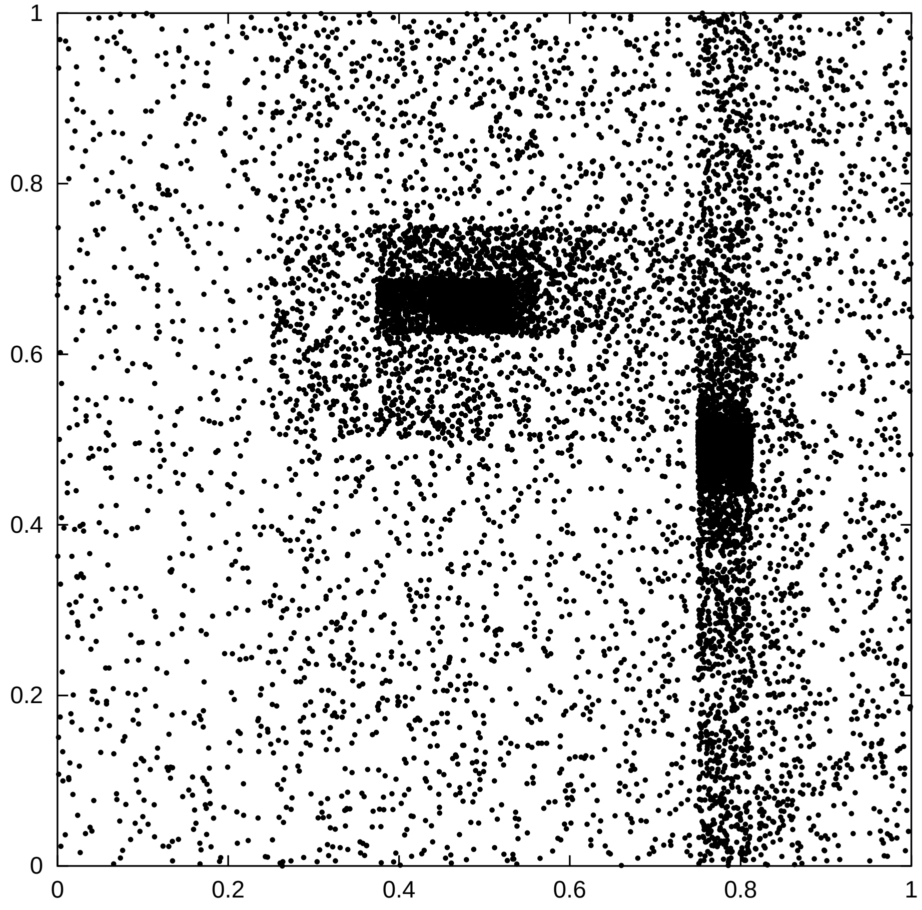
Break f into two parts φ and $(f - \varphi)$ such that

- $\mathrm{I}\varphi$ can be calculated analytically
- $Var(f - \varphi)$ is small
- $$\begin{aligned} \mathrm{Q}_n f &= \mathrm{I}\varphi + \frac{1}{n}\sum_{i=1}^{n}(f - \varphi)(\boldsymbol{x}_i) \\ &= \mathrm{I}\varphi + \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) + \frac{1}{n}\sum_{i=1}^{n}\varphi(\boldsymbol{x}_i) \\ &= \mathrm{I}\varphi + \tilde{f} + \tilde{\varphi} \end{aligned}$$
- $$Var(\mathrm{Q}_n f) = Var(\tilde{f}) + Var(\tilde{\varphi}) - 2Cov(\tilde{f}, \tilde{\varphi})$$

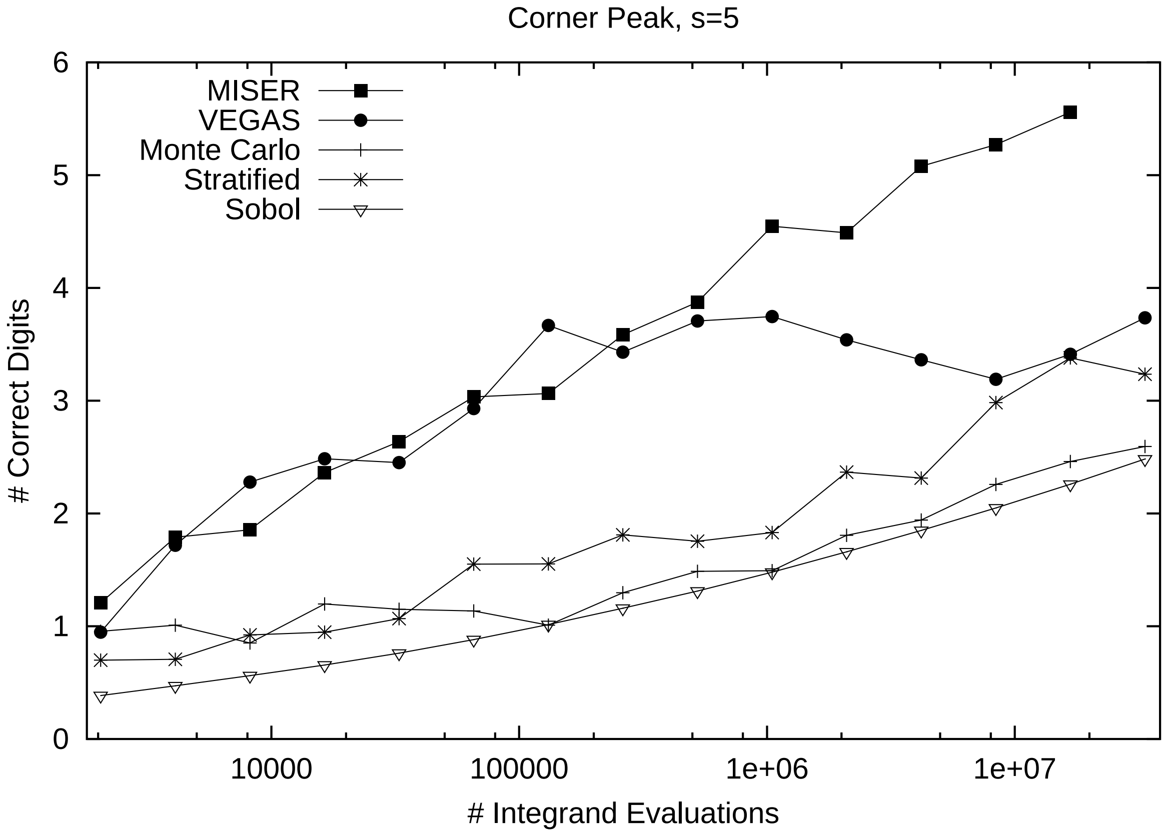
How can be find φ?

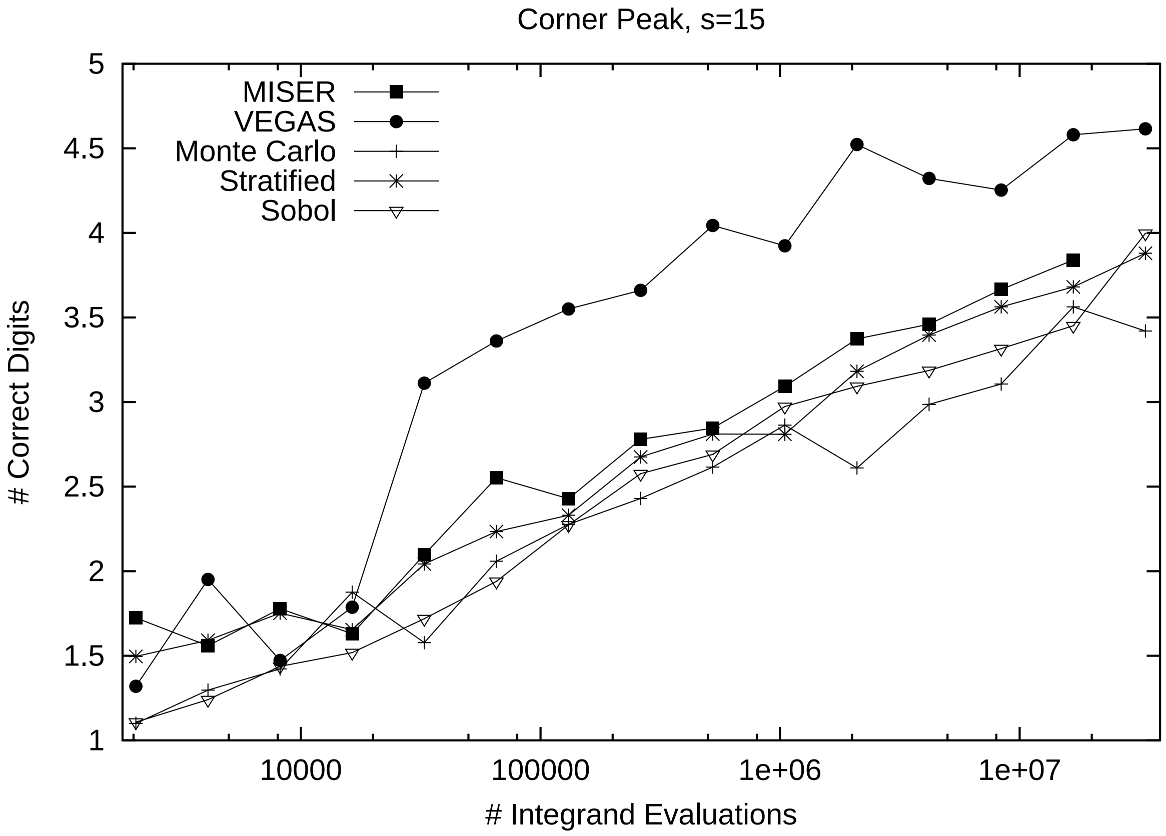
- "Parallel Simulation"
- Adaptive?

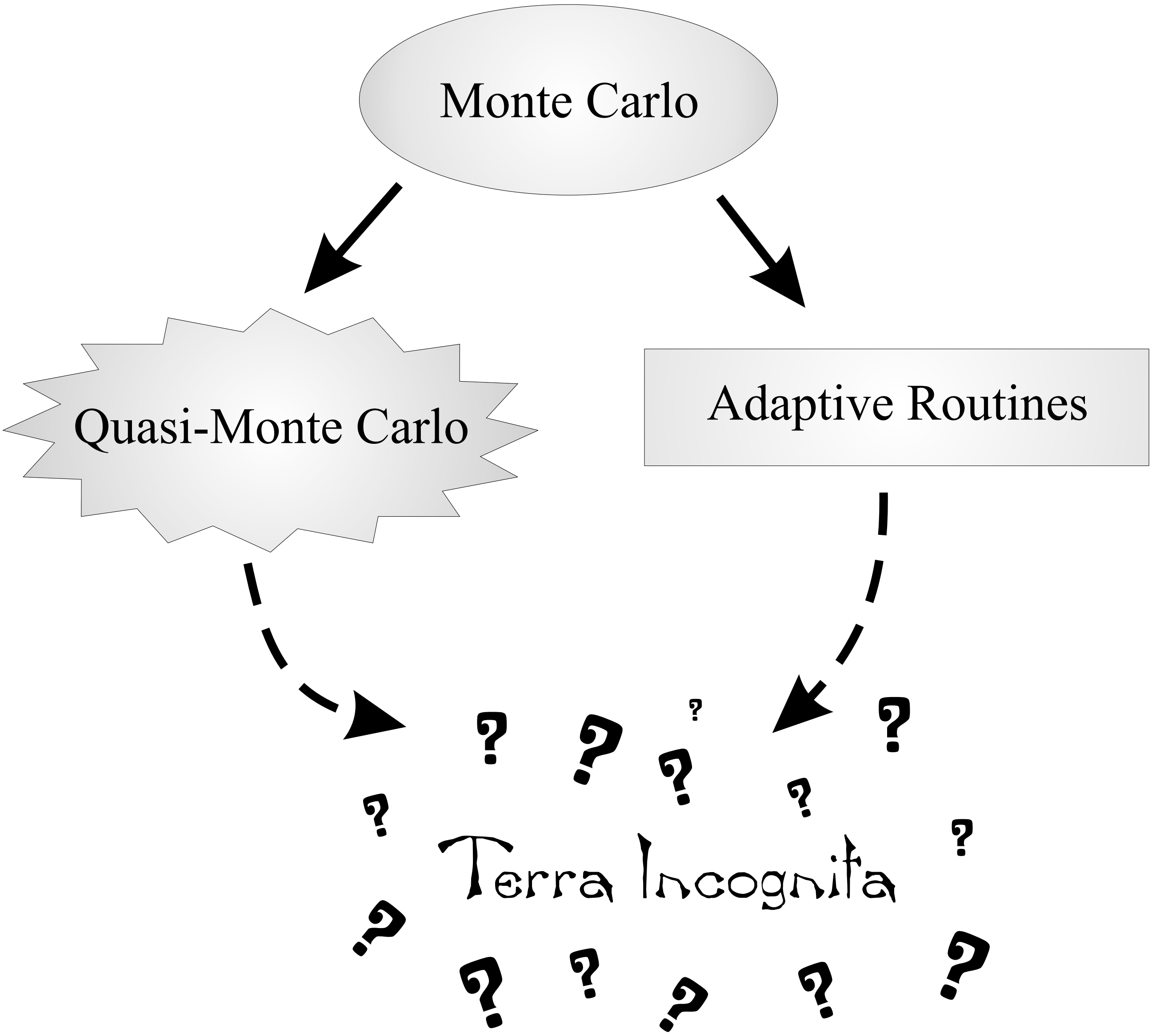

1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

Results


Corner Peak, s=5
MISER
VEGAS
Monte Carlo
Stratified
Sobol
Correct Digits
Integrand Evaluations
0
1
2
3
4
5
6
10000
100000
1e+06
1e+07


Corner Peak, s=15
MISER
VEGAS
Monte Carlo
Stratified
Sobol
Correct Digits
Integrand Evaluations
1
1.5
2
2.5
3
3.5
4
4.5
5
10000
100000
1e+06
1e+07


Monte Carlo
Quasi-Monte Carlo
Adaptive Routines
Terra Incognita

Can we extend these results to QMC?

- Key Concept "Variance Reduction"
  - $Var(f)$ can be estimated easily. However, there is no direct correlation between $Var(f)$ and the integration error.
  - Knowing $V(f)$ would give an upper bound. However, it can neither be estimated nor is the inequality sharp.
  - There are empirical results suggesting that $|\mathrm{Q}_n f - \mathrm{I} f| \approx \frac{Var(f)}{n}$ for many integrands
  - Integration error can be estimated using randomized QMC (e. g.: Owen Scrambling)
- Generating arbitrary distributions is possible, at least if an explicit transformation function is available.
- Applying two nets of size $n/2$ to two halfs of C_s definitely *decreases* performance
- QMC integration has a high rate of convergence by itself. Therefore, improving on it will be harder.