Monte Carlo Methods

Simon Plätzer

DESY Theory Group

Outline and Disclaimer

What the lecture will cover:

- Monte Carlo methods: What and how.
- From integrals to integrands and back: integration and sampling.
- Some more advanced topics, modulo time constraints.

What the lecture certainly not covers:

- The art of (pseudo-) random number generation.
- Technical details of available methods/codes.
- MC methods outside HEP, in particular in statistical physics.

A Travel Guide to Monte Carlo

If there are any questions, don't hesitate to interrupt me.

A Travel Guide to Monte Carlo

Getting started:

- Why go to Monte Carlo at all?

A first tour:

- Sampling by inversion.
- Dealing with many variables: hit-and-miss.
- From hits to weights and integrals.

For experienced visitors:

- Loading the dice: variance reduction.
- A first glimpse on VEGAS.

Additional tours for longer stays:

- MC methods for NLO & parton showers.

Expectations to the Journey.

The Task, Physics Wise.

Given some cross section differential in all momentum components ...

- Calculate the total cross section σ.
- With arbitrary acceptance criteria ('cuts').
- Produce a sample of events $(p_1, ..., p_n)$ with probability density

$$\frac{1}{\sigma}\mathrm{d}\sigma(p_1, ..., p_n) \ .$$

- Book histograms for arbitrary observables, and compare to data

The Task, Technically.

Given a function $p(x_1, ..., x_n)$,
and a volume $V = \{(x_1, ..., x_n) \in \mathbb{R}^n | v(x_1, ..., x_n) = 0\}$ ...

- Calculate
$$N(p, V) = \int_V p(x_1, ..., x_n) \mathrm{d}^n x \ .$$
- Produce a sample of events $(x_1, ..., x_n) \in V$ with probability density
$$\frac{1}{N(p, V)} p(x_1, ..., x_n) \mathrm{d}^n x \ .$$
- Book histograms for arbitrary functions $O(x_1, ..., x_n)$.

Why go to Monte Carlo at all?

Why go to Monte Carlo at all?

Well, we could just do numerical integrations,

$$N(p, V) = \int_V^{\text{Gauss},\ldots} p(x_1, \ldots, x_n)\mathrm{d}^n x \ .$$

Don't even need 'events', nor histograms, but calculate

$$\frac{\mathrm{d}p}{\mathrm{d}O} = \int_V p(x_1, \ldots, x_n)\delta(O - O(x_1, \ldots, x_n))\mathrm{d}^n x$$

the same way, and just plot into measured histograms.

Why go to Monte Carlo at all?

Well, we could just do numerical integrations,

$$N(p,V) = \int_V^{\text{Gauss},\dots} p(x_1,\dots,x_n)\mathrm{d}^n x \ .$$

Don't even need 'events', nor histograms, but calculate

$$\frac{\mathrm{d}p}{\mathrm{d}O} = \int_V p(x_1,\dots,x_n)\delta(O - O(x_1,\dots,x_n))\mathrm{d}^n x$$

the same way, and just plot into measured histograms.

So what?

Why go to Monte Carlo at all?

There are $n = 3k - 4$ variables for k outgoing particles.

- Watch out for convergence of numerical integrations for $n \gtrsim 2$.

Why go to Monte Carlo at all?

There are $n = 3k - 4$ variables for k outgoing particles.

- Watch out for convergence of numerical integrations for $n \gtrsim 2$.

Flexibility: Easily add or change observables O and cut definitions v.

- Would have to adapt the numerical integrations each time.

Why go to Monte Carlo at all?

There are $n = 3k - 4$ variables for k outgoing particles.

- Watch out for convergence of numerical integrations for $n \gtrsim 2$.

Flexibility: Easily add or change observables O and cut definitions v.

- Would have to adapt the numerical integrations each time.

MC methods are the only feasible way to achieve the task.

We'll see how ...

Getting into Touch with MC Methods.

Getting into Touch with MC Methods.

We'll start off with drawing random variates in a single variable.

May seem unrelated to the problems we want to solve, yet:

- Gives a first feeling for what is going on.
- Often needed as helper for more efficient algorithms.

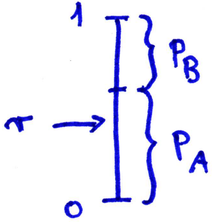
How to do things with some probability?

How to do things with some probability?

Choose between two outcomes A and B with probabilities $P_{A,B}$.
How to implement an algorithm selecting either one according to $P_{A,B}$?

Have **rnd**() to return equally distributed random numbers $r \in [0, 1]$.

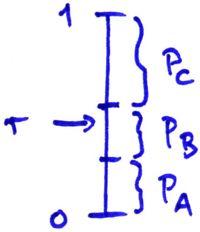
$r \leftarrow$ **rnd**()
if $r < P_A$ **then**
  return A
else
  return B
end if



How to do things with some probability?

Choose between three outcomes A, B and C with probabilities $P_{A,B,C}$.

$r \leftarrow$ **rnd**()
if $r < P_A$ **then**
  return A
else if $r < P_A + P_B$ **then**
  return B
else
  return C
end if



Etc.

Sampling by Inversion.

Suppose we got $p(x) \geq 0$ and $V = [a, b]$ to define a probability density

$$P(x)\mathrm{d}x = \theta(b-x)\theta(x-a)\frac{p(x)\mathrm{d}x}{\int_a^b p(z)\mathrm{d}z}$$

from which we are to draw random variates.

We'll assume that p is sufficiently simple such that

- we can calculate the integral of p, and
- we can solve

$$\int_a^x p(z)\mathrm{d}z = r \int_a^b p(z)\mathrm{d}z$$

for x as a function of $r \in [0, 1]$.

Sampling by Inversion.

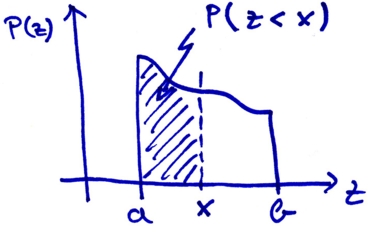
The algorithm generating events according to $P(x)$ is simple:

$r \leftarrow$ **rnd**()
$x \leftarrow$ solution of
$\int_a^x p(z)\mathrm{d}z = r \;\; \int_a^b p(z)\mathrm{d}z$
return x

Sampling by Inversion.

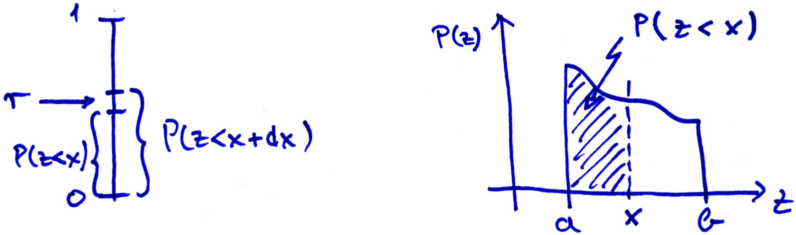
The algorithm generating events according to $P(x)$ is simple:

$r \leftarrow$ **rnd**()
$x \leftarrow$ solution of
$\int_a^x p(z)\mathrm{d}z = r \int_a^b p(z)\mathrm{d}z$
return x



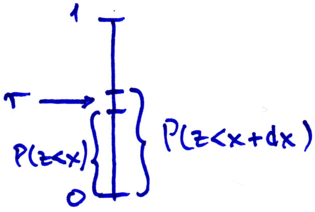
Sampling by Inversion.

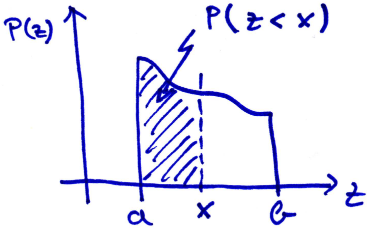
The algorithm generating events according to $P(x)$ is simple:



Sampling by Inversion.

The algorithm generating events according to $P(x)$ is simple:





$$\mathrm{d}r = P(z)\mathrm{d}z$$

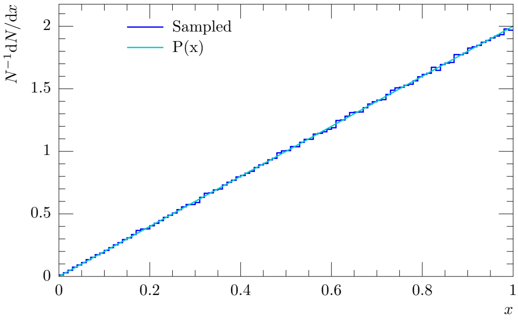
We only solved a change of variables.

Sampling by Inversion: Example.

Suppose we have $p(x) = x$ on $[0, 1]$. Then solve $\frac{x^2}{2} = r\ \frac{1}{2}$.

Sampling by Inversion: Example.

Suppose we have $p(x) = x$ on $[0, 1]$. Then solve $\frac{x^2}{2} = r\ \frac{1}{2}$.



Dealing with Many Variables: Hit-and-Miss.

Dealing with Many Variables: Hit-and-Miss.

Before trying many variables: What if we cannot invert the integral?

Dealing with Many Variables: Hit-and-Miss.

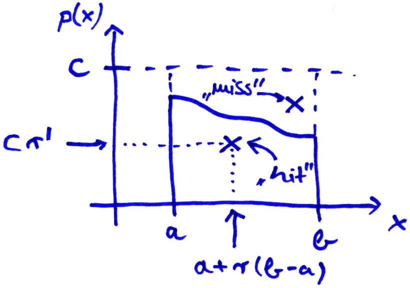
Before trying many variables: What if we cannot invert the integral? Suppose we know $c \geq p(x)$.

```
loop
  r ← rnd()
  x ← a + r(b − a)
  r′ ← rnd()
  if r′ < p(x)/c then
    return x
  end if
end loop
```

Dealing with Many Variables: Hit-and-Miss.

Before trying many variables: What if we cannot invert the integral? Suppose we know $c \geq p(x)$.

loop
  $r \leftarrow$ **rnd**()
  $x \leftarrow a + r(b - a)$
  $r' \leftarrow$ **rnd**()
  if $r' < p(x)/c$ **then**
    return x
  end if
end loop

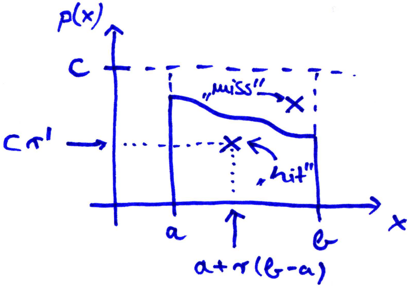


Dealing with Many Variables: Hit-and-Miss.

Before trying many variables: What if we cannot invert the integral?
Suppose we know $c \geq p(x)$.

---

loop
  $r \leftarrow$ **rnd**()
  $x \leftarrow a + r(b - a)$
  $r' \leftarrow$ **rnd**()
  if $r' < p(x)/c$ **then**
    return x
  end if
end loop

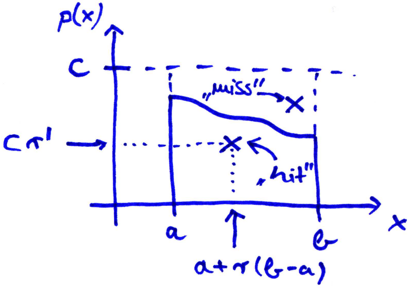


The frequency of hits in $[x, x + \mathrm{d}x]$ is directly proportional to $p(x)$.

Dealing with Many Variables: Hit-and-Miss.

Before trying many variables: What if we cannot invert the integral?
Suppose we know $c \geq p(x)$.

```
loop
  r ← rnd()
  x ← a + r(b − a)
  r' ← rnd()
  if r' < p(x)/c then
    return x
  end if
end loop
```



Note that we did not have to know the normalization!

Dealing with Many Variables: Hit-and-Miss.

Given a function $p(x_1, ..., x_n)$,
and a volume $V = \{(x_1, ..., x_n) \in \mathbb{R}^n | v(x_1, ..., x_n) = 0\}$ ...

Dealing with Many Variables: Hit-and-Miss.

Given a function $p(x_1, ..., x_n)$,
and a volume $V = \{(x_1, ..., x_n) \in \mathbb{R}^n | v(x_1, ..., x_n) = 0\}$ ...

Suppose we know $c \geq p(x_1, ..., x_n)$.

Dealing with Many Variables: Hit-and-Miss.

Given a function $p(x_1, ..., x_n)$,
and a volume $V = \{(x_1, ..., x_n) \in \mathbb{R}^n | v(x_1, ..., x_n) = 0\}$ ...

Suppose we know $c \geq p(x_1, ..., x_n)$.
And a hypercube $I = [a_1, b_1] \times \cdots \times [a_n, b_n]$ with $V \subset I$.

Dealing with Many Variables: Hit-and-Miss.

Given a function $p(x_1, ..., x_n)$,
and a volume $V = \{(x_1, ..., x_n) \in \mathbb{R}^n | v(x_1, ..., x_n) = 0\}$ ...

Suppose we know $c \geq p(x_1, ..., x_n)$.
And a hypercube $I = [a_1, b_1] \times \cdots \times [a_n, b_n]$ with $V \subset I$.

Define

$$p_V(x_1, ..., x_n) = \begin{cases} p(x_1, ..., x_n) & : \quad v(x_1, ..., x_n) = 0 \\ 0 & : \quad \text{otherwise} \end{cases} \quad .$$

Dealing with Many Variables: Hit-and-Miss.

$$p_V(x_1, ..., x_n) = \begin{cases} p(x_1, ..., x_n) & : \quad v(x_1, ..., x_n) = 0 \\ 0 & : \quad \text{otherwise} \end{cases}$$

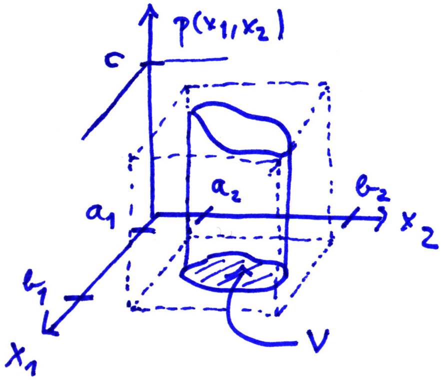
---

loop
  for $i = 1..n$ **do**
    $r \leftarrow$ **rnd**()
    $x_i \leftarrow a_i + r(b_i - a_i)$
  end for
  $r' \leftarrow$ **rnd**()
  if $r' < p_V(x_1, ..., x_n)/c$ **then**
    return $(x_1, ..., x_n)$
  end if
end loop

Dealing with Many Variables: Hit-and-Miss.

$$p_V(x_1, ..., x_n) = \begin{cases} p(x_1, ..., x_n) & : \quad v(x_1, ..., x_n) = 0 \\ 0 & : \quad \text{otherwise} \end{cases}$$

```
loop
  for i = 1..n do
    r ← rnd()
    x_i ← a_i + r(b_i − a_i)
  end for
  r' ← rnd()
  if r' < p_V(x_1, ..., x_n)/c then
    return (x_1, ..., x_n)
  end if
end loop
```



Remark.

Unless stated otherwise: Back to one variable.

Generalizations should be obvious now.

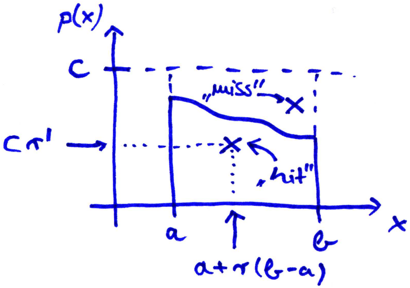
If not: Please ask!

From Hits to Weights and Integrals.

From Hits to Weights and Integrals.

Before trying many variables: What if we cannot invert the integral?
Suppose we know $c \geq p(x)$.

```
loop
    r ← rnd()
    x ← a + r(b − a)
    r' ← rnd()
    if r' < p(x)/c then
        return x
    end if
end loop
```

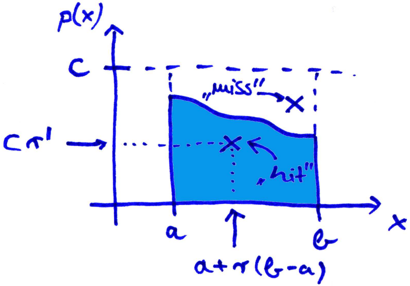


Note that we did not have to know the normalization!

From Hits to Weights and Integrals.

We actually *estimated* the normalization, if we were counting hits:

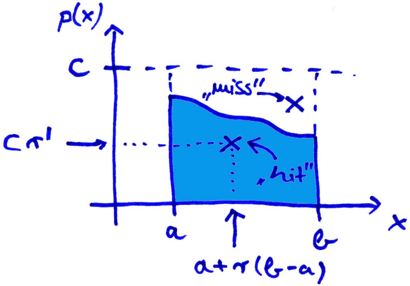
$$\int_a^b p(x)\mathrm{d}x \approx \frac{\#\text{hits}}{\#\text{hits} + \#\text{misses}} \times c(a-b)\ .$$



From Hits to Weights and Integrals.

We actually *estimated* the normalization, if we were counting hits:

$$\int_a^b p(x)\mathrm{d}x \approx \frac{\#\text{hits}}{\#\text{hits} + \#\text{misses}} \times c(a-b)\ .$$



In other words: We have just (approximately) calculated an integral!

Weights, Averages, and Variances.

Let's put the estimate onto a more waterproof ground.

Weights, Averages, and Variances.

Let's put the estimate onto a more waterproof ground.

Averaging p over $[a, b]$ is connected to its integral,

$$\langle p \rangle = \frac{1}{b-a} \int_a^b p(x) \, \mathrm{d}x \ .$$

Weights, Averages, and Variances.

Let's put the estimate onto a more waterproof ground.

Averaging p over $[a, b]$ is connected to its integral,

$$\langle p \rangle = \frac{1}{b-a} \int_a^b p(x) \, \mathrm{d}x \; .$$

Now estimate the average by

- recording p's value at random points x_i,
- for a total of N points:

$$\langle p \rangle_{\text{estimate}} = \frac{1}{N} \sum_{i=1}^{N} p(x_i) \; .$$

Weights, Averages, and Variances.

We'll call $w_i = p(x_i)$ the *weight* of an event x_i.
w_i is a measure of how many hits we should expect in $[x_i, x_i + \mathrm{d}x]$.

Weights, Averages, and Variances.

We'll call $w_i = p(x_i)$ the *weight* of an event x_i.
w_i is a measure of how many hits we should expect in $[x_i, x_i + \mathrm{d}x]$.

$\langle p \rangle_{\text{estimate}}$ now got a well defined uncertainty:
We *measure* p at equally distributed, independent random points.

Weights, Averages, and Variances.

We'll call $w_i = p(x_i)$ the *weight* of an event x_i.
w_i is a measure of how many hits we should expect in $[x_i, x_i + \mathrm{d}x]$.

$\langle p \rangle_{\text{estimate}}$ now got a well defined uncertainty:
We *measure* p at equally distributed, independent random points.

The variance of $\langle p \rangle_{\text{estimate}}$ is

$$\sigma^2 \left[\langle p \rangle_{\text{estimate}}\right] = \frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N} w_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} w_i\right)^2\right) .$$

Monte Carlo Integrals.

By recording p's value at random points x_i, $i = 1, ..., N$ we can approximatley calculate its integral:

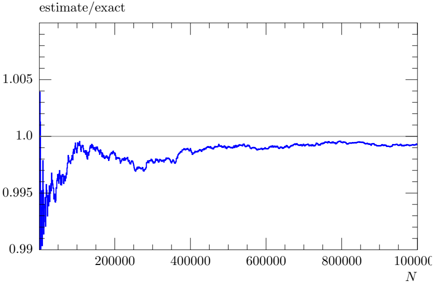
Monte Carlo Integrals.

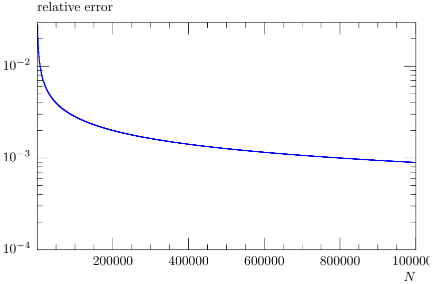
By recording p's value at random points x_i, $i = 1, ..., N$ we can approximatley calculate its integral:

$$\boxed{\int_a^b p(x)\,\mathrm{d}x = (b-a)\langle p\rangle_{\text{estimate}} \pm (b-a)\sigma\left[\langle p\rangle_{\text{estimate}}\right]}$$

Monte Carlo Integrals: Example.

Integrate $p(x) = x^2$ on $[0, 1]$.

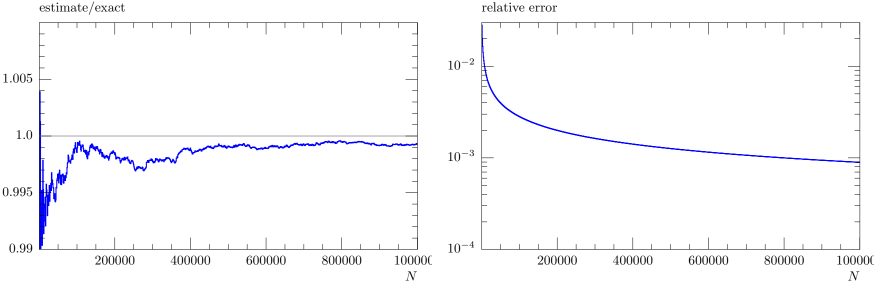




Uncertainty drops as $1/\sqrt{N}$. Mind the independent measurements.

Monte Carlo Integrals: Example.

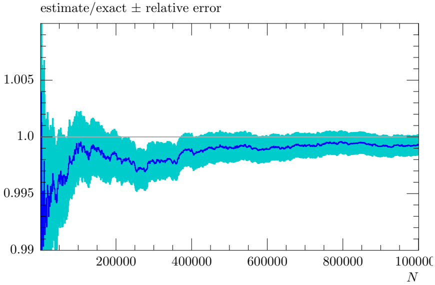
Integrate $p(x) = x^2$ on $[0, 1]$.



Doesn't really converge to the true value, right?

Monte Carlo Integrals: Example.

Integrate $p(x) = x^2$.



Even worse: Error band just scratches true value for large N.

Monte Carlo Integrals.

Mind the choice of your random number generator!

Monte Carlo Integrals.

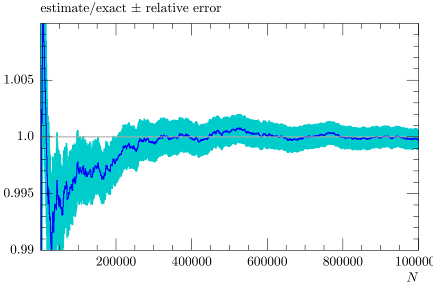
Mind the choice of your random number generator!

Never, ever use things like:

`rnd(), drand48() ...`

Monte Carlo Integrals: Example.

Integrate $p(x) = x^2$.



Same thing, better random number generator.

How does this connect to hits and misses?

How does this connect to hits and misses?

Assign weights

- $w_i = c$ to any 'hit' x_i, and
- $w_j = 0$ to any 'miss' x_j.

How does this connect to hits and misses?

Assign weights

- $w_i = c$ to any 'hit' x_i, and
- $w_j = 0$ to any 'miss' x_j.

Then

$$\int_a^b p(x)\mathrm{d}x \approx \frac{\#\text{hits}}{\#\text{hits} + \#\text{misses}} \times c(a-b)$$

as conjectured.

But now we know how accurate this estimate is.

How does this connect to hits and misses?

We still miss an explanation for $w_{\text{hit}} = c$.

How does this connect to hits and misses?

We still miss an explanation for $w_{\text{hit}} = c$.

We have actually 'measured' $p(x)$ in units of c...

- by accepting $N \times p(x)/c$ hits in $[x, x + \mathrm{d}x]$, thus
- recording the value of $p(x)/c$ by the number of hits.

How does this connect to hits and misses?

We still miss an explanation for $w_{\text{hit}} = c$.

We have actually 'measured' $p(x)$ in units of c...

- by accepting $N \times p(x)/c$ hits in $[x, x + \mathrm{d}x]$, thus
- recording the value of $p(x)/c$ by the number of hits.

For any hit we therefore need to multiply by the unit c we've chosen.

How does this connect to hits and misses?

We still miss an explanation for $w_{\text{hit}} = c$.

We have actually 'measured' $p(x)$ in units of c...

- by accepting $N \times p(x)/c$ hits in $[x, x + \mathrm{d}x]$, thus
- recording the value of $p(x)/c$ by the number of hits.

For any hit we therefore need to multiply by the unit c we've chosen.

Just a scaling of variables:

$$\int_a^b p(x)\mathrm{d}x = c \int_a^b \frac{p(x)}{c}\mathrm{d}x$$

True *changes* of variables when trying to cheat in the casino ...

A Travel Guide to Monte Carlo

Time for questions.

Uncertainties, continued.

Mind the integral's uncertainty,

$$\sigma^2 \left[\langle p \rangle_{\text{estimate}} \right] = \langle \sigma^2 \left[p \right] \rangle_{\text{estimate}} \ .$$

If p has large variance, need a very large N for a reasonable uncertainty.

Uncertainties, continued.

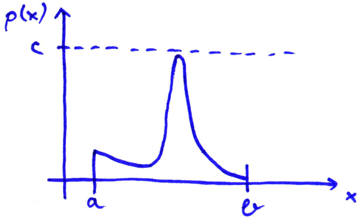
Mind the integral's uncertainty,

$$\sigma^2 \left[\langle p \rangle_{\text{estimate}}\right] = \langle \sigma^2 \left[p\right] \rangle_{\text{estimate}} \ .$$

If p has large variance, need a very large N for a reasonable uncertainty.

Connected to this is an inacceptable efficiency of hit-and-miss,

$$\epsilon = \frac{\#\text{hits}}{\#\text{hits} + \#\text{misses}} \ll 1 \ .$$



Loading the Dice: Variance Reduction.

Loading the Dice: Variance Reduction.

We've been honest gamblers, using equally distributed random numbers.

Loading the Dice: Variance Reduction.

We've been honest gamblers, using equally distributed random numbers.

Now we'll start to cheat.

Loading the Dice: Variance Reduction.

We've been honest gamblers, using equally distributed random numbers.

Now we'll start to cheat.

First set some notation,

$$\langle p \rangle \to \langle p \rangle_1 \ ,$$

where in general

$$\langle p \rangle_r = \int_a^b p(x) r(x) \mathrm{d}x \ .$$

Loading the Dice: Variance Reduction.

The basic ingredients to variance reduction:

- A constant function has zero variance.
- And we always have

$$\langle p \rangle_1 = \left\langle \frac{p}{r} \right\rangle_r \quad .$$

Loading the Dice: Variance Reduction.

The basic ingredients to variance reduction:

- A constant function has zero variance.
- And we always have

$$\langle p \rangle_1 = \left\langle \frac{p}{r} \right\rangle_r \ .$$

So, ideally

$$\langle p \rangle_1 = \langle 1 \rangle_p$$

with zero variance ???

Loading the Dice: Variance Reduction.

What does $\left\langle \frac{p}{r} \right\rangle_r$ actually mean?

Loading the Dice: Variance Reduction.

What does $\left\langle \frac{p}{r} \right\rangle_r$ actually mean?

– A change of variables,

$$p(x)\mathrm{d}x = p(x(R))\frac{\mathrm{d}x(R)}{\mathrm{d}R}\mathrm{d}R$$

with $r(x)\mathrm{d}x = \mathrm{d}R$.
Record p/r at points inside the transformed volume.

Loading the Dice: Variance Reduction.

What does $\left\langle \frac{p}{r} \right\rangle_r$ actually mean?

- A change of variables,

$$p(x)\mathrm{d}x = p(x(R))\frac{\mathrm{d}x(R)}{\mathrm{d}R}\mathrm{d}R$$

  with $r(x)\mathrm{d}x = \mathrm{d}R$.
  Record p/r at points inside the transformed volume.
- If $r(x)$ is normalized to define a probability density:
  Record p/r at points distributed with density r.

Loading the Dice: Variance Reduction.

What does $\left\langle \frac{p}{r} \right\rangle_r$ actually mean?

- A change of variables,

$$p(x)\mathrm{d}x = p(x(R))\frac{\mathrm{d}x(R)}{\mathrm{d}R}\mathrm{d}R$$

  with $r(x)\mathrm{d}x = \mathrm{d}R$.
  Record p/r at points inside the transformed volume.
- If $r(x)$ is normalized to define a probability density:
  Record p/r at points distributed with density r.

To arrive at $\langle 1 \rangle_p$ we would actually have to know the integral.
Then the uncertainty is – of course – zero.

Loading the Dice: Variance Reduction.

The best we can hope for is finding a r, which is very similar to p,

$$\frac{p(x)}{r(x)} \approx \text{constant} \ .$$

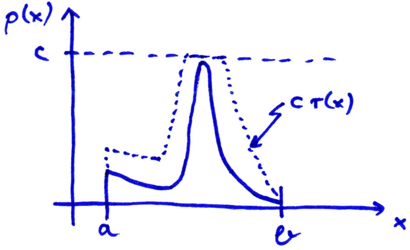
And sufficiently simple, such that we can distribute points with a probability density defined by r.

Loading the Dice: Variance Reduction.

This also helps with the hit-and-miss efficiency:

If we know c such that $c\ r(x) \geq p(x)$, we can

- Propose points with density defined by r, and
- accept a hit x with probability $\frac{p(x)}{c\ r(x)}$.



Loading the Dice: Variance Reduction.

Bottom line:

- Generate more points where p has large fluctuations.
- Generate less points where p is essentially constant.
- Divide out the bias introduced thereby.

Loading the Dice: Variance Reduction.

Bottom line:

- Generate more points where p has large fluctuations.
- Generate less points where p is essentially constant.
- Divide out the bias introduced thereby.

A bit of terminology:

What we got to know here is known as **'importance sampling'**.

Loading the Dice: Variance Reduction.

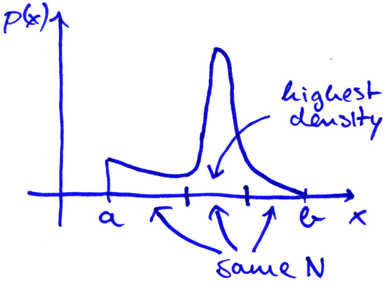
A bit of terminology:

There is also **‘stratified sampling’**.

Loading the Dice: Variance Reduction.

A bit of terminology:

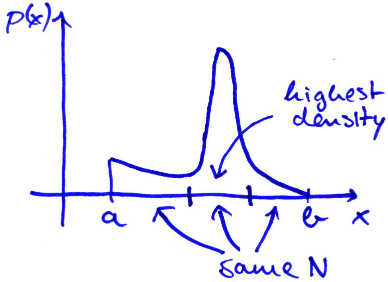
There is also **‘stratified sampling’**.



Loading the Dice: Variance Reduction.

A bit of terminology:

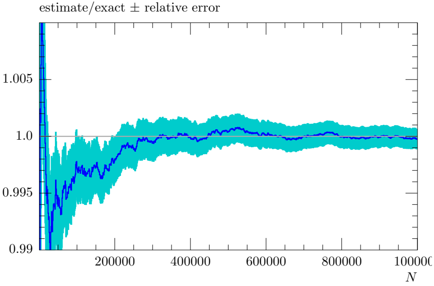
There is also **'stratified sampling'**.

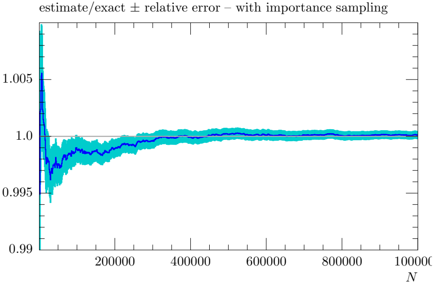


This is just another way of implementing a $r(x)$ made up of step functions.

Variance Reduction: Example.

Integrate $p(x) = x^2$ on $[0, 1]$. Importance sampling with $r(x) = x$.





Detour: From Integrals and Weights to Hits.

Detour: From Integrals and Weights to Hits.

From MC integration we obtain a sample of *weighted* events, (x_i, w_i).

For event generation, we are interested in *unweighted* events, (x_i, c).
Recap that w_i is a measure of the frequency of events in $[x_i, x_i + \mathrm{d}x]$.

Detour: From Integrals and Weights to Hits.

From MC integration we obtain a sample of *weighted* events, (x_i, w_i).

For event generation, we are interested in *unweighted* events, (x_i, c).
Recap that w_i is a measure of the frequency of events in $[x_i, x_i + \mathrm{d}x]$.

To get to unweighted events,

- find the maximum weight w_{max},
- keep each weighted event (x_i, w_i) with probability w_i/w_{max}, and
- assign common weight $c = N(p, V)/N_{\text{uw}}$ to N_{uw} accepted events.

Detour: From Integrals and Weights to Hits.

From MC integration we obtain a sample of *weighted* events, (x_i, w_i).

For event generation, we are interested in *unweighted* events, (x_i, c).
Recap that w_i is a measure of the frequency of events in $[x_i, x_i + \mathrm{d}x]$.

To get to unweighted events,

- find the maximum weight $w_{\max}$,
- keep each weighted event (x_i, w_i) with probability $w_i/w_{\max}$, and
- assign common weight $c = N(p, V)/N_{\mathrm{uw}}$ to N_{uw} accepted events.

NB: For a proper variance reduction, the weight's variance is small compared to the average weight: efficient 'unweighting'.

A First Glimpse on VEGAS.

A First Glimpse on VEGAS.

For this part we'll get back to many variables.

A First Glimpse on VEGAS.

For this part we'll get back to many variables.

What is VEGAS? [G.P. Lepage, J.Comput.Phys.27:192,1978]

A First Glimpse on VEGAS.

For this part we'll get back to many variables.

What is VEGAS? [G.P. Lepage, J.Comput.Phys.27:192,1978]

- An algorithm for stratified sampling.
- In a very clever way to deal with a huge number of variables.
- Adaptive: Adjust $r(x)$ in each *iteration* to get smaller variance.

A First Glimpse on VEGAS.

For this part we'll get back to many variables.

What is VEGAS? [G.P. Lepage, J.Comput.Phys.27:192,1978]

- An algorithm for stratified sampling.
- In a very clever way to deal with a huge number of variables.
- Adaptive: Adjust $r(x)$ in each *iteration* to get smaller variance.

Implementations: Lepage's listing, MONACO, GSL, dVegas, ...

A First Glimpse on VEGAS.

The troubles with many variables.

A First Glimpse on VEGAS.

The troubles with many variables.

Suppose we want to make up $r(x_1, ..., x_n)$ of step functions.
Divide each variable range $[a_i, b_i]$ into k intervals

$$[a_i = x_{i,0}, x_{i,1}], [x_{i,1}, x_{i,2}], ..., [x_{i,k-1}, b_i = x_{i,k}] \ ,$$

A First Glimpse on VEGAS.

The troubles with many variables.

Suppose we want to make up $r(x_1, \ldots, x_n)$ of step functions.
Divide each variable range $[a_i, b_i]$ into k intervals

$$[a_i = x_{i,0}, x_{i,1}], [x_{i,1}, x_{i,2}], \ldots, [x_{i,k-1}, b_i = x_{i,k}] \ ,$$

and let

$$r(x_1, \ldots, x_n) =$$
$$\sum_{i_1, \ldots, i_n = 1}^{k} r_{i_1, \ldots, i_n} \theta(x_{1,i_1} - x_1)\theta(x_1 - x_{1,i_1-1}) \cdots \theta(x_{n,i_n} - x_n)\theta(x_n - x_{n,i_n-1}) \ .$$

A First Glimpse on VEGAS.

The troubles with many variables.

We would have to keep track of k^n values $r_{i_1,\ldots,i_n}$.

A First Glimpse on VEGAS.

The troubles with many variables.

We would have to keep track of k^n values $r_{i_1,\ldots,i_n}$.

Suppose we have a $2 \to 4$ scattering at a hadron collider:
$n = 3 \times 4 - 4 + 2 = 10$.

A First Glimpse on VEGAS.

The troubles with many variables.

We would have to keep track of k^n values $r_{i_1,\ldots,i_n}$.

Suppose we have a $2 \to 4$ scattering at a hadron collider:
$n = 3 \times 4 - 4 + 2 = 10$.

And figure out that $k \sim 4$ looks reasonable (mostly way too small!).

A First Glimpse on VEGAS.

The troubles with many variables.

We would have to keep track of k^n values $r_{i_1,\ldots,i_n}$.

Suppose we have a $2 \to 4$ scattering at a hadron collider:
$n = 3 \times 4 - 4 + 2 = 10$.

And figure out that $k \sim 4$ looks reasonable (mostly way too small!).

A `double` is 64 $\mathrm{Bits} = 8\ \mathrm{Bytes}$:

$$4^{10} \times 8\ \mathrm{Bytes} = 8\ \mathrm{GBytes}$$

Not an option.

A First Glimpse on VEGAS.

Suppose we want to make up $r(x_1, ..., x_n)$ of step functions.
Divide each variable range $[a_i, b_i]$ into k intervals

$$[a_i = x_{i,0}, x_{i,1}], [x_{i,1}, x_{i,2}], ..., [x_{i,k-1}, b_i = x_{i,k}] \ ,$$

A First Glimpse on VEGAS.

Suppose we want to make up $r(x_1, ..., x_n)$ of step functions.
Divide each variable range $[a_i, b_i]$ into k intervals

$$[a_i = x_{i,0}, x_{i,1}], [x_{i,1}, x_{i,2}], ..., [x_{i,k-1}, b_i = x_{i,k}] \ ,$$

and let

$$r(x_1, ..., x_n) = r_1(x_1) \cdots r_n(x_n)$$

with

$$r_i(x_i) = \sum_{j=1}^{k} r_{i,j} \theta(x_{i,j} - x_i) \theta(x_i - x_{i,j-1}) \ .$$

A First Glimpse on VEGAS.

Suppose we want to make up $r(x_1, ..., x_n)$ of step functions.
Divide each variable range $[a_i, b_i]$ into k intervals

$$[a_i = x_{i,0}, x_{i,1}], [x_{i,1}, x_{i,2}], ..., [x_{i,k-1}, b_i = x_{i,k}] \ ,$$

and let

$$r(x_1, ..., x_n) = r_1(x_1) \cdots r_n(x_n)$$

with

$$r_i(x_i) = \sum_{j=1}^{k} r_{i,j} \theta(x_{i,j} - x_i) \theta(x_i - x_{i,j-1}) \ .$$

$n \times k$ values to store instead of k^n.
Now need 320 Bytes instead of 8 GBytes!

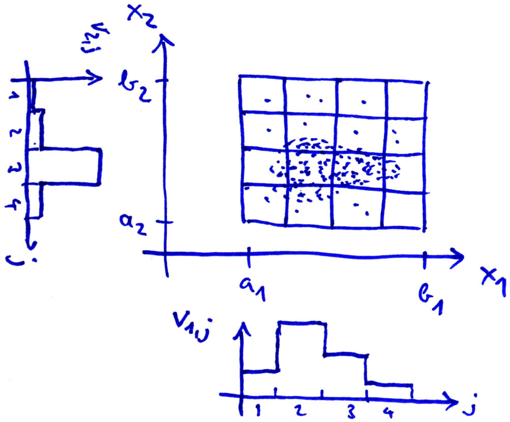
A First Glimpse on VEGAS.

Keep number of points per interval fixed.
Trade of $r_{i,j}$ to keep track of

$$v_{i,j} = \frac{1}{(b_1 - a_1)\cdots(x_{i,j} - x_{i,j-1})\cdots(b_n - a_n)} \times \int_{a_1}^{b_1} \mathrm{d}x_1 \cdots \int_{x_{i,j-1}}^{x_{i,j}} \mathrm{d}x_i \cdots \int_{a_n}^{b_n} \mathrm{d}x_n \; p(x_1, ..., x_n) \; ,$$

i.e. the average weight in $[x_{i,j-1}, x_{i,j}]$ *projected* onto variable i.

A First Glimpse on VEGAS.

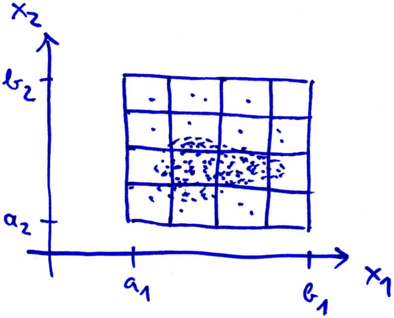


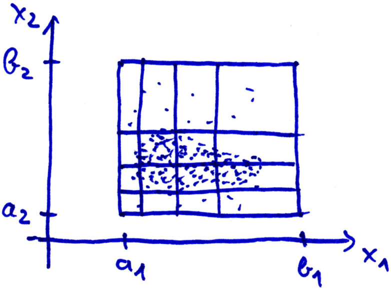
A First Glimpse on VEGAS.

The adaptive algorithm:

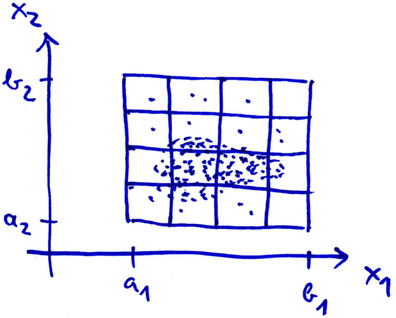
- Make a first run ('iteration') with N points.
- Adjust the $x_{i,j}$ such as to have higher density, where $v_{i,j}$ are large.
- Repeat for M iterations.

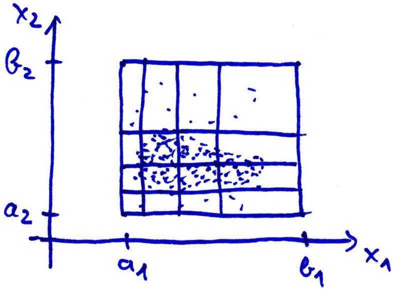
A First Glimpse on VEGAS.





A First Glimpse on VEGAS.





- Adjust the $x_{i,j}$ such as to have higher density, where $v_{i,j}$ are large.

This criterion is actually not unique, and rather complicated.

A Travel Guide to Monte Carlo

Time for questions.

MC Methods for NLO & Parton Showers.

MC Methods for NLO & Parton Showers.

Negative 'probability densities':

- subtraction terms.

MC Methods for NLO & Parton Showers.

Negative ‘probability densities’:

- subtraction terms.

Markov processes:

- next parton shower emission.

Negative ‘Probability Densities’

Negative 'Probability Densities'

Parts of a NLO differential cross section can be negative.

Can we still make sense of this in terms of MC methods?

Negative ‘Probability Densities’

Parts of a NLO differential cross section can be negative.

Can we still make sense of this in terms of MC methods?

If we just want to calculate an integral, there’s no problem.
We just have that some of the weights w_i are negative.

Negative ‘Probability Densities’

Parts of a NLO differential cross section can be negative.

Can we still make sense of this in terms of MC methods?

If we just want to calculate an integral, there’s no problem.
We just have that some of the weights w_i are negative.

But what about ‘events’?

Negative ‘Probability Densities’

There’s a very simple way out of this:

Negative 'Probability Densities'

There's a very simple way out of this:

If $p(x)$ goes negative for some values of x ...

- Generate events according to $|p(x)|$.
- For x with $p(x) > 0$ add an event to histogram bin.
- For x with $p(x) < 0$ subtract an event from histogram bin.

Negative 'Probability Densities'

There's a very simple way out of this:

If $p(x)$ goes negative for some values of x ...

- Generate events according to $|p(x)|$.
- For x with $p(x) > 0$ add an event to histogram bin.
- For x with $p(x) < 0$ subtract an event from histogram bin.

In other words:

Unweight from w_i to $c \times \mathrm{sign}(w_i)$ with
acceptance probability $|w_i|/\max|w_i|$.

Markov Processes.

Markov Processes.

Technically, parton showers are Markov processes.

Markov Processes.

Technically, parton showers are Markov processes.

Will not go into details, just state the problem:

Markov Processes.

Technically, parton showers are Markov processes.

Will not go into details, just state the problem:

Draw events from a probability density

$$\frac{\mathrm{d}S_p(\mu, q|Q)}{\mathrm{d}q} = \Delta_p(\mu|Q)\delta(q-\mu) + p(q)\Delta_p(q|Q)\theta(Q-q)\theta(q-\mu)$$

where

$$\Delta_p(q|Q) = \exp\left(-\int_q^Q P(t)\mathrm{d}t\right) \ .$$

Markov Processes.

How to achieve this?

Markov Processes.

How to achieve this?

First note, that we're truly facing a probability density,

$$\int_{\mu}^{Q} \frac{\mathrm{d}S_p(\mu, q|Q)}{\mathrm{d}q} \mathrm{d}q = 1 \ .$$

Markov Processes.

How to achieve this?

First note, that we're truly facing a probability density,

$$\int_{\mu}^{Q} \frac{\mathrm{d}S_p(\mu, q|Q)}{\mathrm{d}q} \mathrm{d}q = 1 \ .$$

We just use sampling by inversion, solving for q in

$$\int_{\mu}^{q} \frac{\mathrm{d}S_p(\mu, t|Q)}{\mathrm{d}t} \mathrm{d}t = \Delta_p(q|Q) = \mathbf{rnd}() \ .$$

Markov Processes.

There's a caveat:

$$\Delta_p(q|Q) = \mathbf{rnd}()$$

has no solution if **rnd**() returned a value smaller than $\Delta_p(\mu|Q)$.

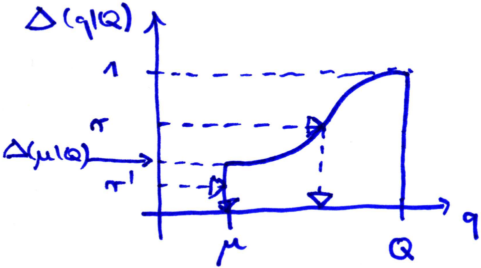
Markov Processes.

There's a caveat:

$$\Delta_p(q|Q) = \textbf{rnd}()$$

has no solution if **rnd**() returned a value smaller than $\Delta_p(\mu|Q)$.

This is precisely giving us the contribution multiplying the δ-function.



Markov Processes.

What if we can't solve $\Delta_p(q|Q) = \mathbf{rnd}()$ for q?

Markov Processes.

What if we can't solve $\Delta_p(q|Q) = \mathbf{rnd}()$ for q?

There's something like a hit-and-miss algorithm,
known as the 'Sudakov veto algorithm'.

Unfortunately, we'll have to leave Monte Carlo now ...

Memories from Monte Carlo

differential cross sections

events

probability densities

hits and misses

weights

total cross sections

integrals

Hands on a MC Event Generator

Hands on a MC Event Generator

Everything outlined in the lecture more or less feeds into a Monte Carlo event generator.

Hands on a MC Event Generator

Everything outlined in the lecture more or less feeds into a Monte Carlo event generator.

We'll now get to know one of these: Herwig++.

[M. Bähr, S. Gieseke, M.A. Gigg, D. Grellscheid, K. Hamilton, O. Latunde-Dada, SP, P. Richardson, M.H. Seymour, A. Sherstnev, B.R. Webber, Eur.Phys.J.C58:639-707,2008]

Hands on a MC Event Generator

Everything outlined in the lecture more or less feeds into a Monte Carlo event generator.

We'll now get to know one of these: Herwig++.

[M. Bähr, S. Gieseke, M.A. Gigg, D. Grellscheid, K. Hamilton, O. Latunde-Dada, SP, P. Richardson, M.H. Seymour, A. Sherstnev, B.R. Webber, Eur.Phys.J.C58:639-707,2008]

(After coffee...)