Event generation

a crash course

FROM INTEGRATION TO EVENT GENERATION

- Calculations of cross section or decay widths involve integrations over phase space of very complex functions

FROM INTEGRATION TO EVENT GENERATION

- Calculations of cross section or decay widths involve integrations over phase space of very complex functions

$$\sigma = \frac{1}{2s} \int |\mathcal{M}|^2 d\Phi(n)$$

FROM INTEGRATION TO EVENT GENERATION

- Calculations of cross section or decay widths involve integrations over phase space of very complex functions

$$Dim[\Phi(n)] \sim 3n$$

$$\sigma = \frac{1}{2s} \int |\mathcal{M}|^2 d\Phi(n)$$

FROM INTEGRATION TO EVENT GENERATION

- Calculations of cross section or decay widths involve integrations over phase space of very complex functions

$$Dim[\Phi(n)] \sim 3n$$

$$\sigma = \frac{1}{2s} \int |\mathcal{M}|^2 d\Phi(n)$$

General and flexible method is needed

Phase Space

Phase Space

$$d\Phi_n = \left[\Pi_{i=1}^{n} \frac{d^3 p_i}{(2\pi)^3 (2E_i)}\right] (2\pi)^4 \delta^{(4)}(p_0 - \sum_{i=1}^{n} p_i)$$

Phase Space

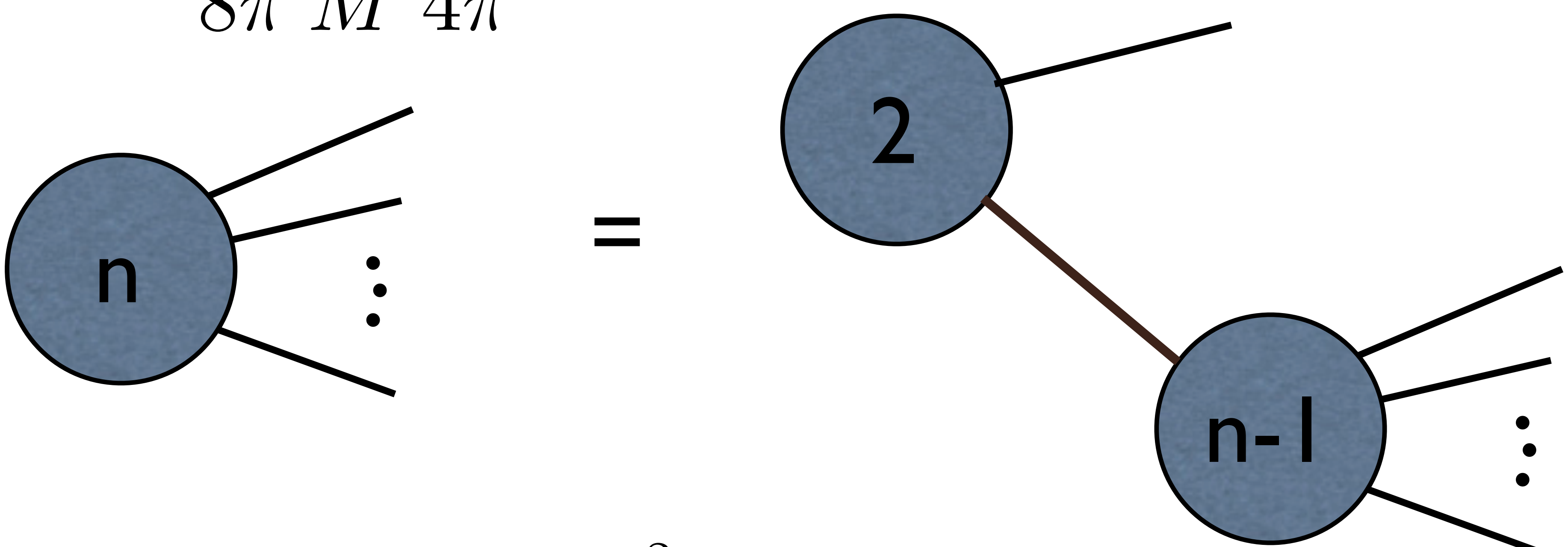
$$d\Phi_n = \left[\Pi_{i=1}^n \frac{d^3 p_i}{(2\pi)^3 (2E_i)}\right] (2\pi)^4 \delta^{(4)}(p_0 - \sum_{i=1}^n p_i)$$

$$d\Phi_2(M) = \frac{1}{8\pi} \frac{2p}{M} \frac{d\Omega}{4\pi}$$

Phase Space

$$d\Phi_n = \left[\Pi_{i=1}^n \frac{d^3 p_i}{(2\pi)^3 (2E_i)}\right] (2\pi)^4 \delta^{(4)}(p_0 - \sum_{i=1}^n p_i)$$

$$d\Phi_2(M) = \frac{1}{8\pi} \frac{2p}{M} \frac{d\Omega}{4\pi}$$



$$d\Phi_n(M) = \frac{1}{2\pi} \int_0^{(M-\mu)^2} d\mu^2 d\Phi_2(M) d\Phi_{n-1}(\mu)$$

INTEGRALS AS AVERAGES

Integrals as averages

$$I = \int_{x_1}^{x_2} f(x)dx \quad \Rightarrow \quad I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \quad \Rightarrow \quad V_N = (x_2 - x_1)^2 \frac{1}{N}\sum_{i=1}^{N} [f(x)]^2 - I_N^2$$

Integrals as averages

$$I = \int_{x_1}^{x_2} f(x)dx \quad \Rightarrow \quad I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \quad \Rightarrow \quad V_N = (x_2 - x_1)^2 \frac{1}{N}\sum_{i=1}^{N} [f(x)]^2 - I_N^2$$

$$I = I_N \pm \sqrt{V_N/N}$$

INTEGRALS AS AVERAGES

$$I = \int_{x_1}^{x_2} f(x)dx \Rightarrow I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

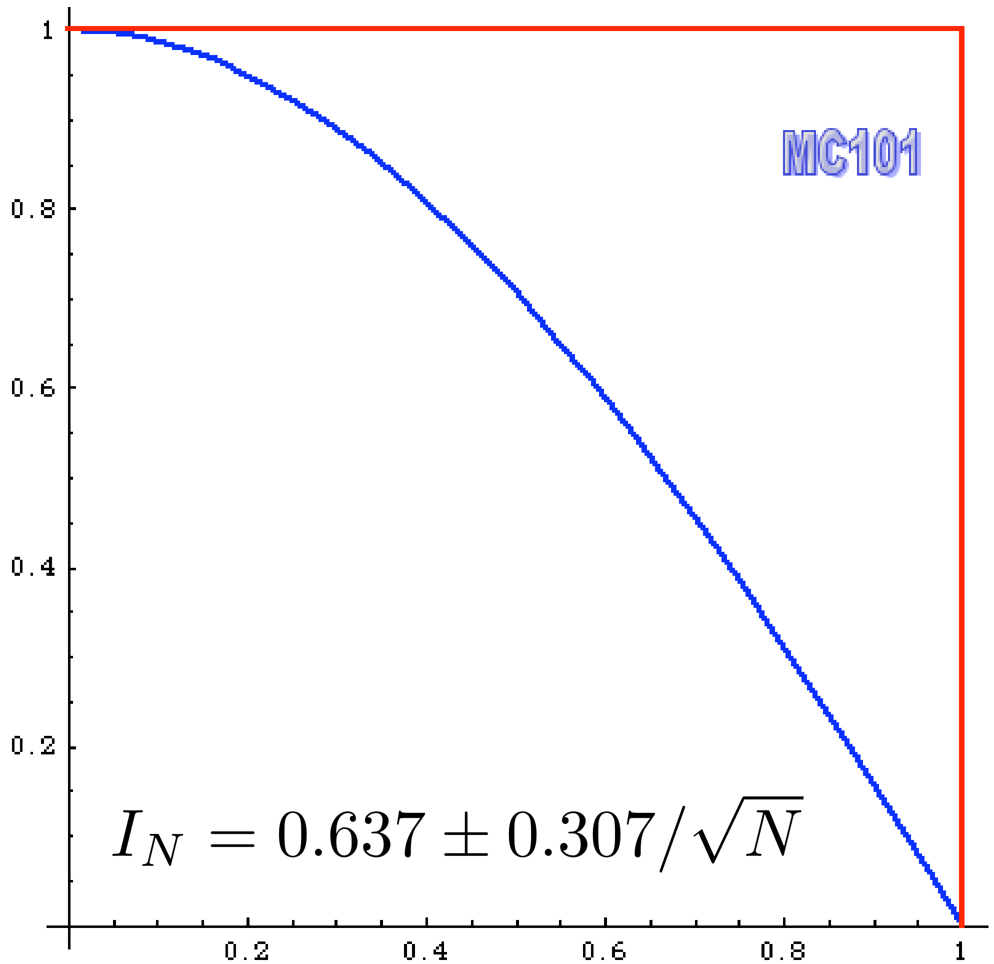
$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \Rightarrow V_N = (x_2 - x_1)^2\frac{1}{N}\sum_{i=1}^{N}[f(x)]^2 - I_N^2$$

$$I = I_N \pm \sqrt{V_N/N}$$

- ☞ Convergence is slow but it can be estimated easily
- ☞ Error does not depend on # of dimensions!
- ☞ Improvement by minimizing V_N.
- ☞ Optimal/Ideal case: $f(x)=C \Rightarrow V_N=0$

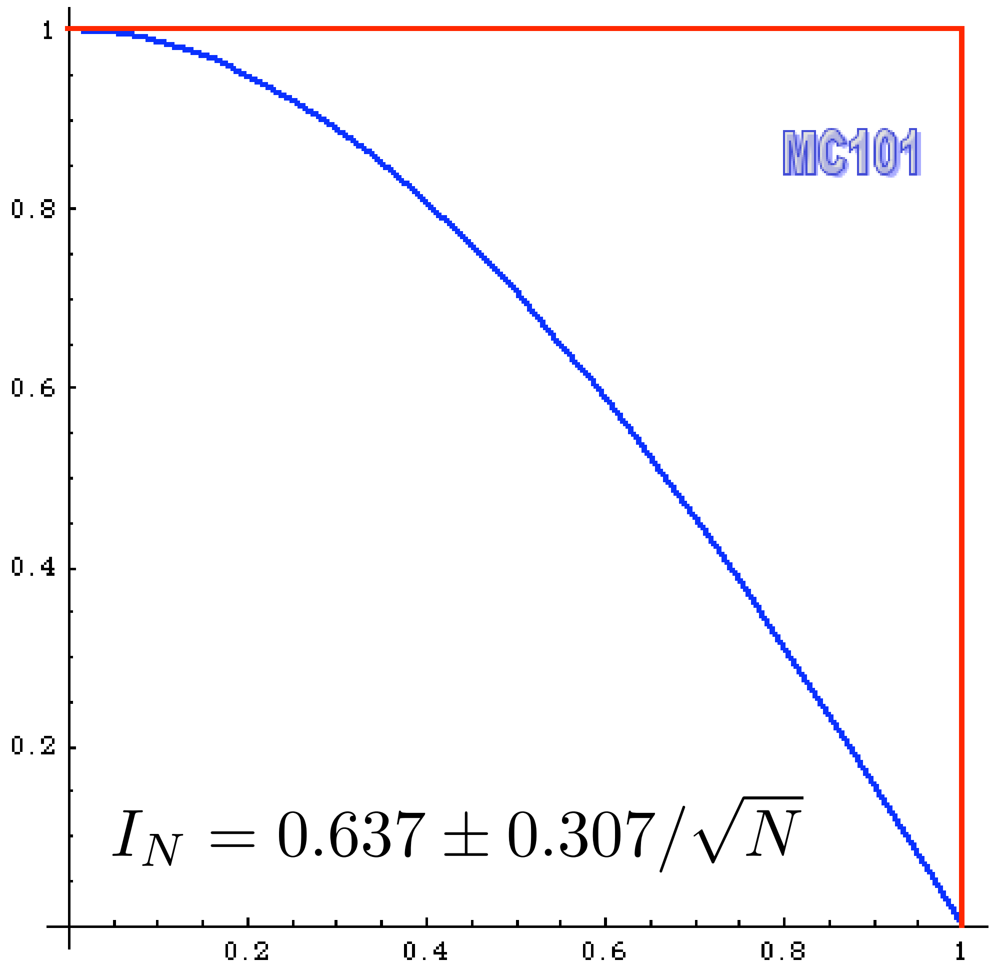
Importance Sampling

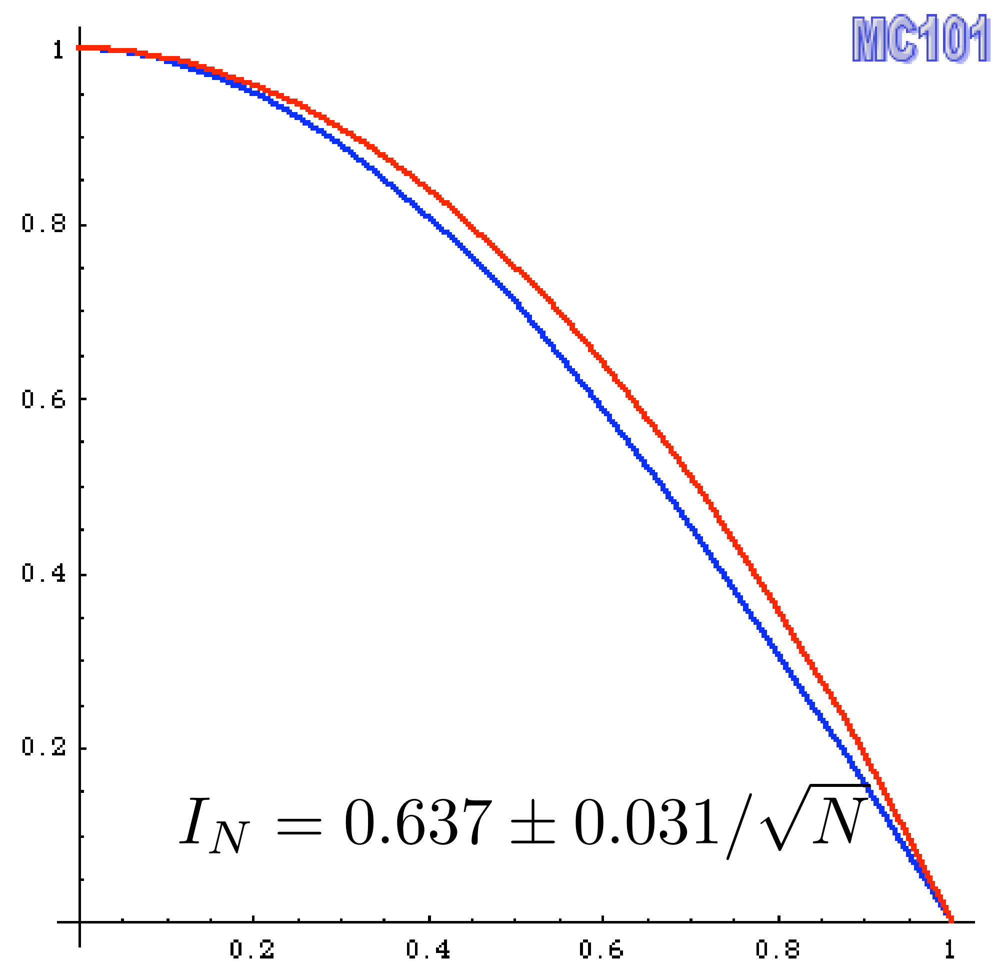
Importance Sampling



$$I = \int_0^1 dx \cos\frac{\pi}{2}x$$

Importance Sampling

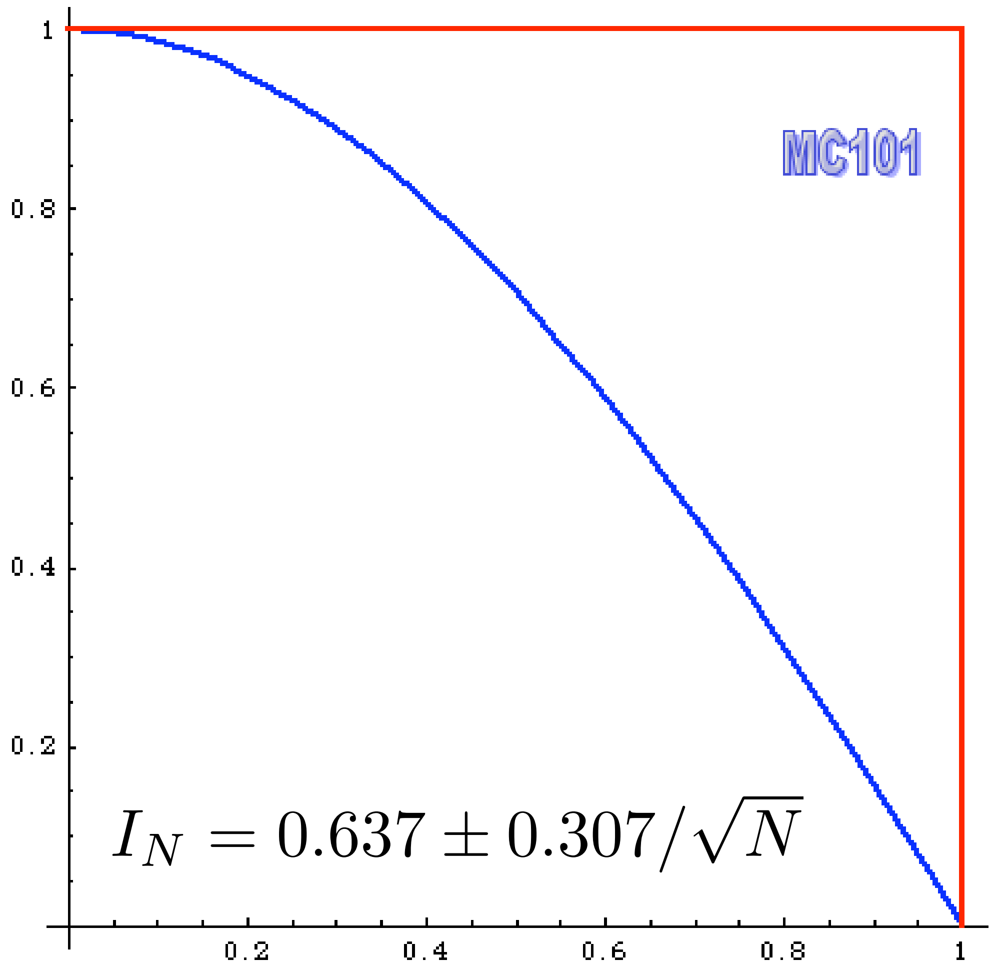


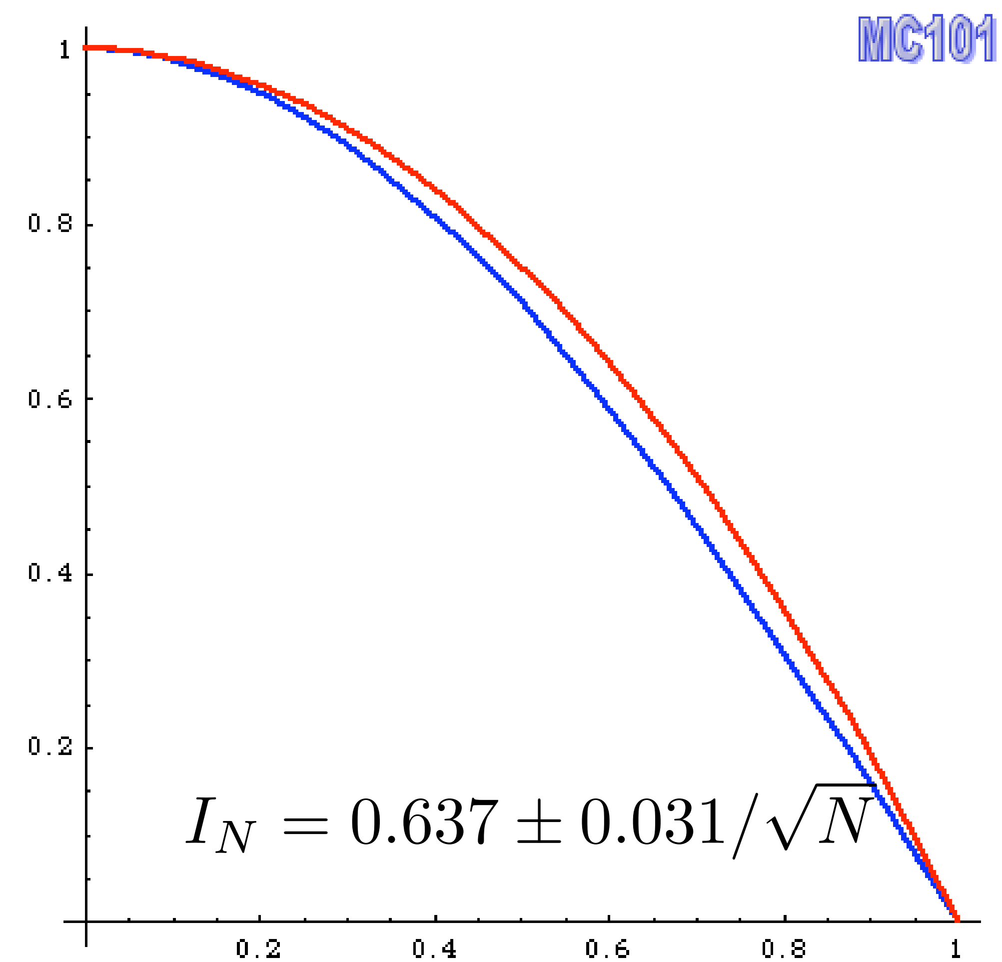


$$I = \int_0^1 dx \cos\frac{\pi}{2}x$$

$$I = \int_0^1 dx(1-x^2)\frac{\cos\frac{\pi}{2}x}{1-x^2}$$

Importance Sampling

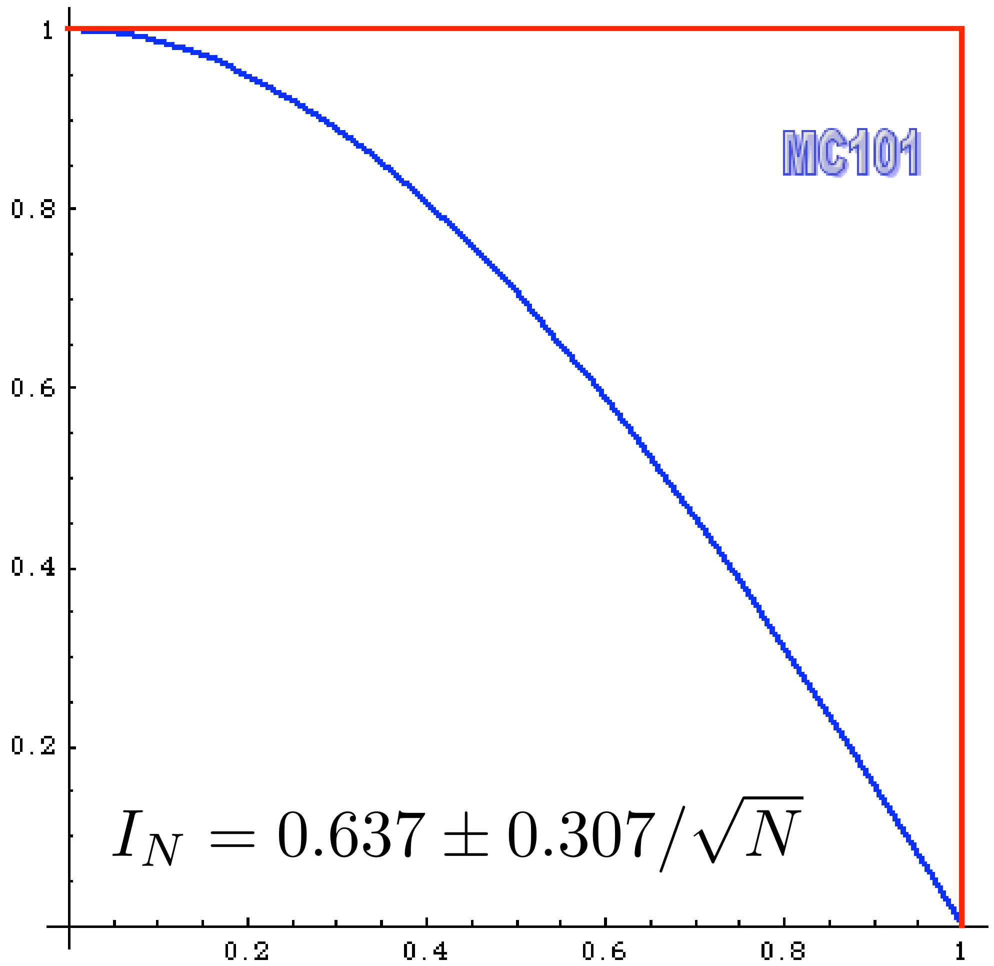


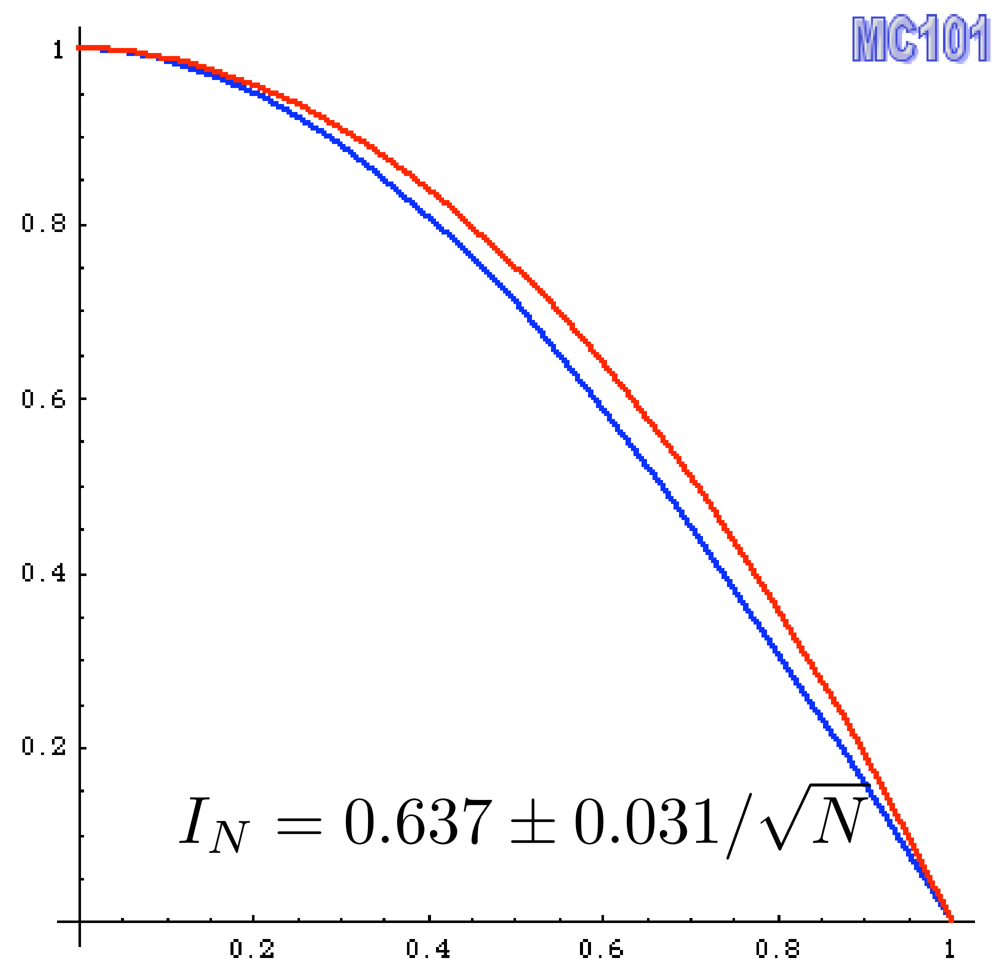


$$I = \int_0^1 dx \cos \frac{\pi}{2} x$$

$$I = \int_0^1 dx (1 - x^2) \frac{\cos \frac{\pi}{2} x}{1 - x^2}$$
$$= \int_{\xi_1}^{\xi_2} d\xi \frac{\cos \frac{\pi}{2} x[\xi]}{1 - x[\xi]^2}$$

Importance Sampling



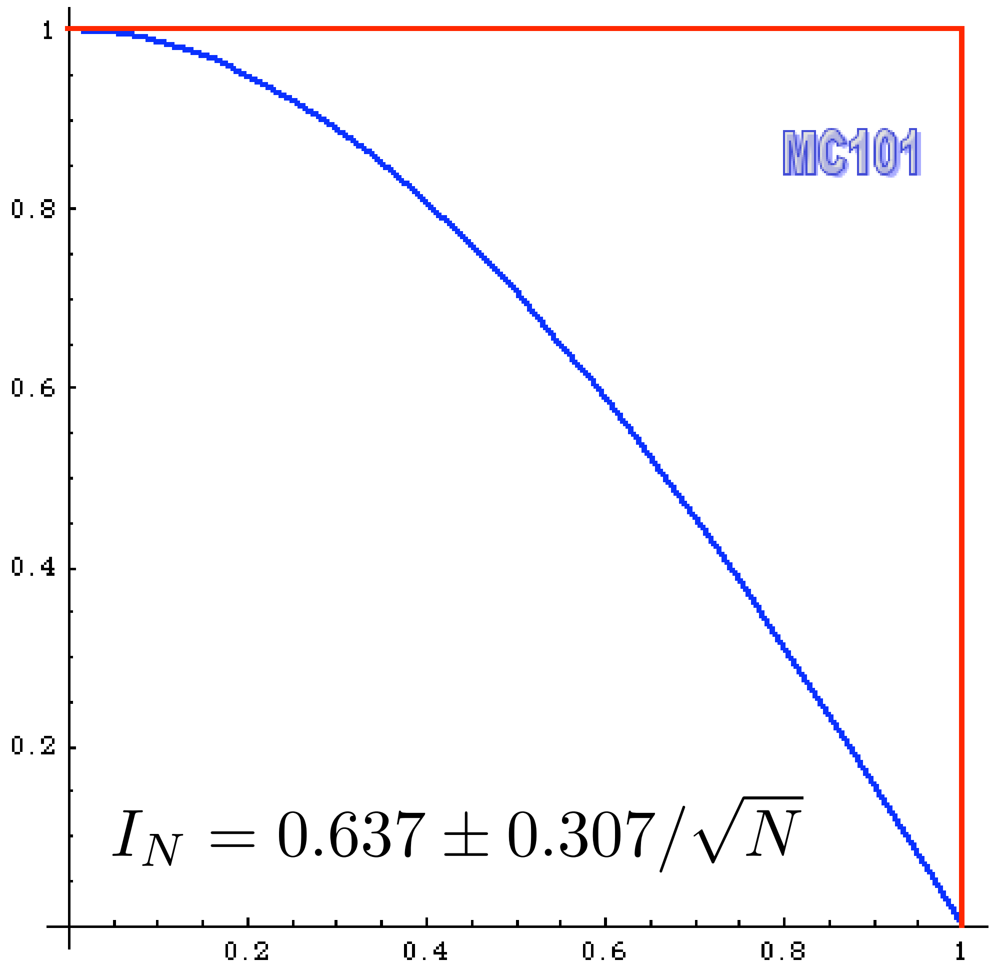


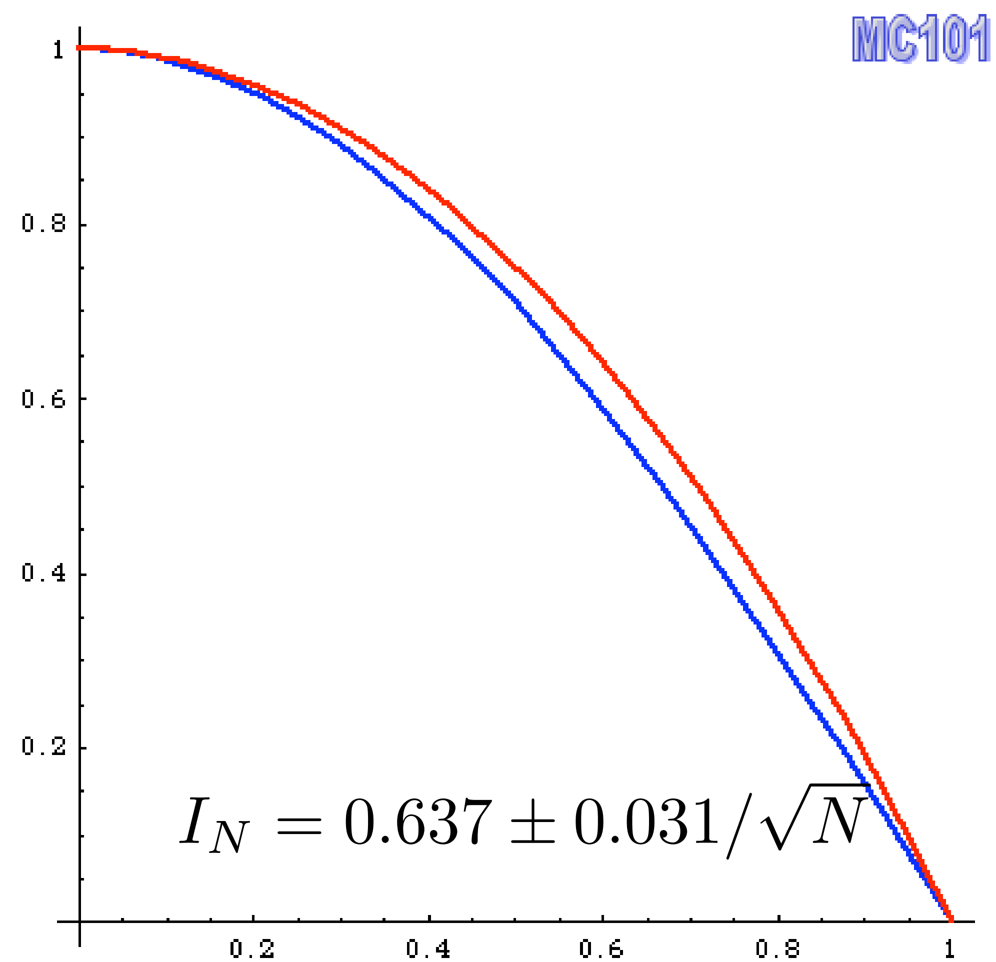
$$I = \int_0^1 dx \cos\frac{\pi}{2}x$$

$$I = \int_0^1 dx(1-x^2)\frac{\cos\frac{\pi}{2}x}{1-x^2}$$

$$= \int_{\xi_1}^{\xi_2} d\xi \frac{\cos\frac{\pi}{2}x[\xi]}{1-x[\xi]^2}$$

Importance Sampling





$$I = \int_0^1 dx \cos \frac{\pi}{2} x$$

$$I = \int_0^1 dx (1 - x^2) \frac{\cos \frac{\pi}{2} x}{1 - x^2}$$

$$= \int_{\xi_1}^{\xi_2} d\xi \frac{\cos \frac{\pi}{2} x[\xi]}{1 - x[\xi]^2} \Rightarrow \simeq 1$$

Importance Sampling

Importance Sampling

but... you need to know too much about f(x)!

Importance Sampling

but... you need to know too much about f(x)!

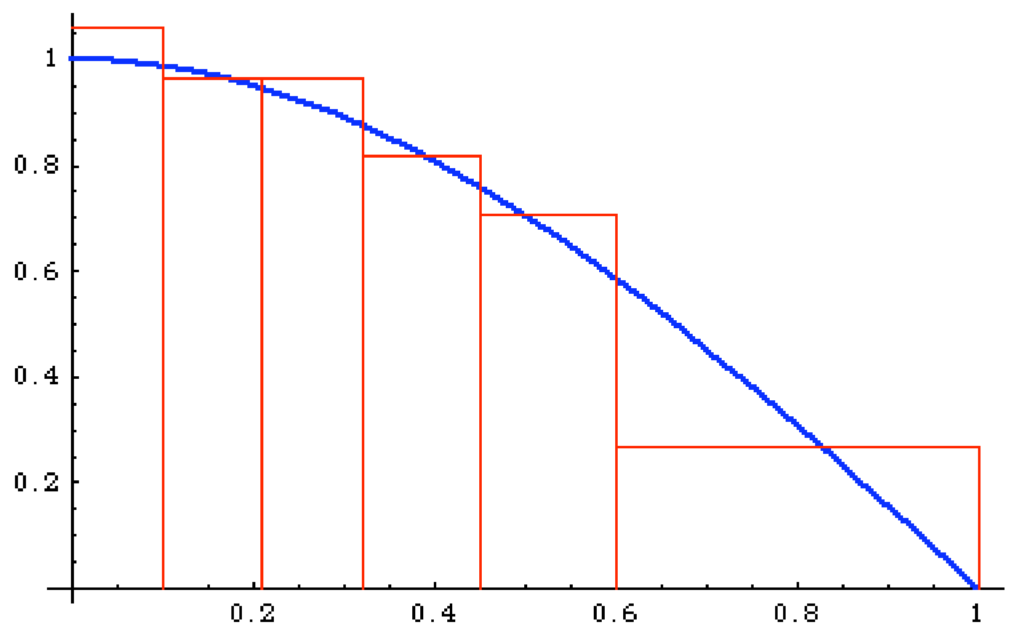
idea: learn during the run and build a step-function approximation p(x) of f(x) ⇒ VEGAS

Importance Sampling

but... you need to know too much about f(x)!

idea: learn during the run and build a step-function approximation p(x) of f(x) ⇒ VEGAS

MC101

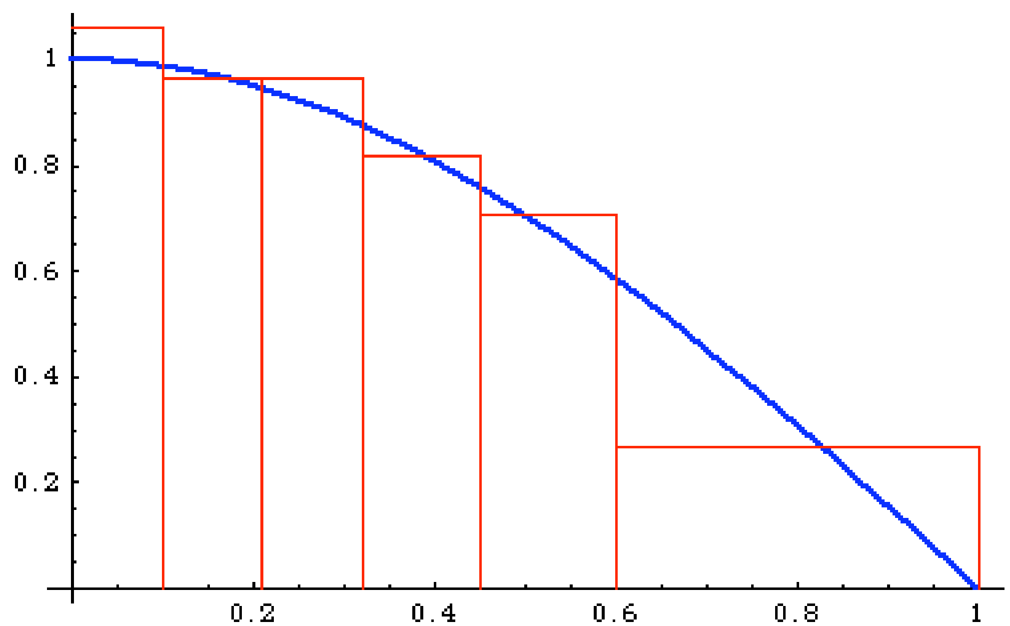


Importance Sampling

but... you need to know too much about f(x)!

idea: learn during the run and build a step-function approximation p(x) of f(x) ⟶ VEGAS

MC101



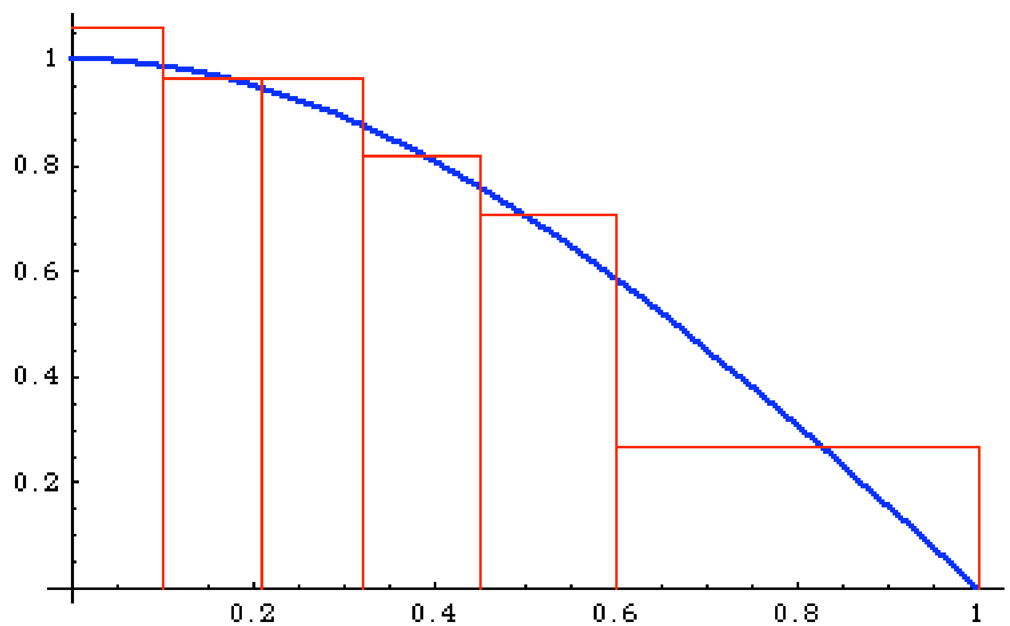
many bins where f(x) is large

Importance Sampling

but... you need to know too much about f(x)!

idea: learn during the run and build a step-function approximation p(x) of f(x) → VEGAS

MC101



many bins where f(x) is large

$$p(x) = \frac{1}{N_b \Delta x_i}, \quad x_i - \Delta x_i < x < x_i$$

Importance Sampling

Importance Sampling

can be generalized to n dimensions:

$$p(\vec{x}) = p(x) \bullet p(y) \bullet p(z) \dots$$

Importance Sampling

can be generalized to n dimensions:

$$p(\vec{x}) = p(x) \cdot p(y) \cdot p(z) \ldots$$

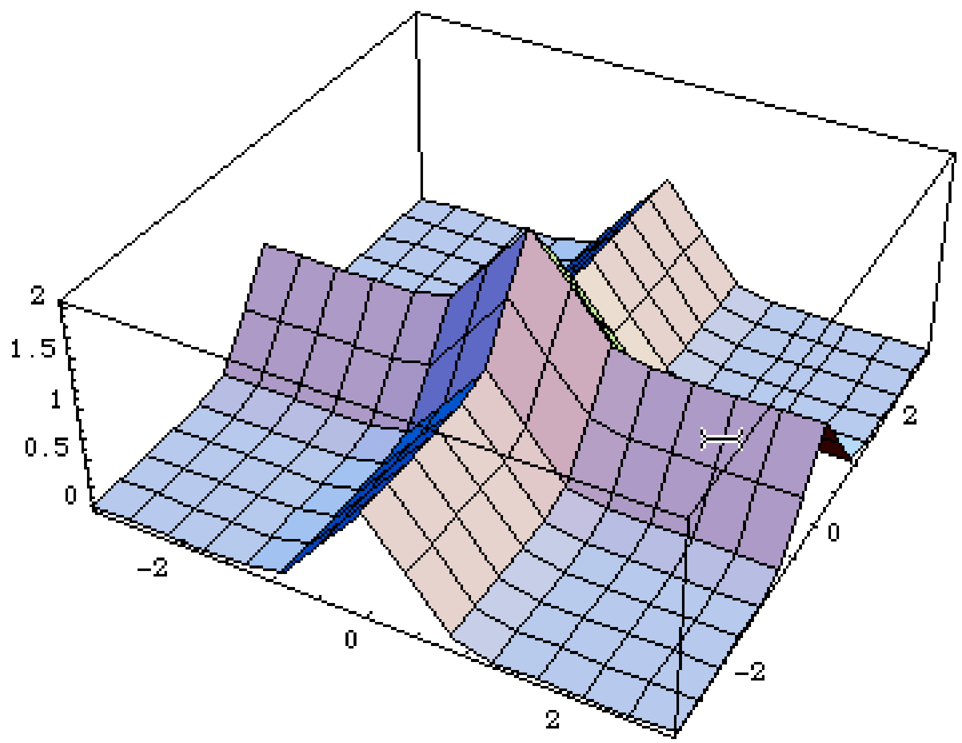
but the peaks of $f(\vec{x})$ need to be "aligned" to the axis!

Importance Sampling

can be generalized to n dimensions:

$$p(\vec{x}) = p(x) \bullet p(y) \bullet p(z) \ldots$$

but the peaks of $f(\vec{x})$ need to be "aligned" to the axis!



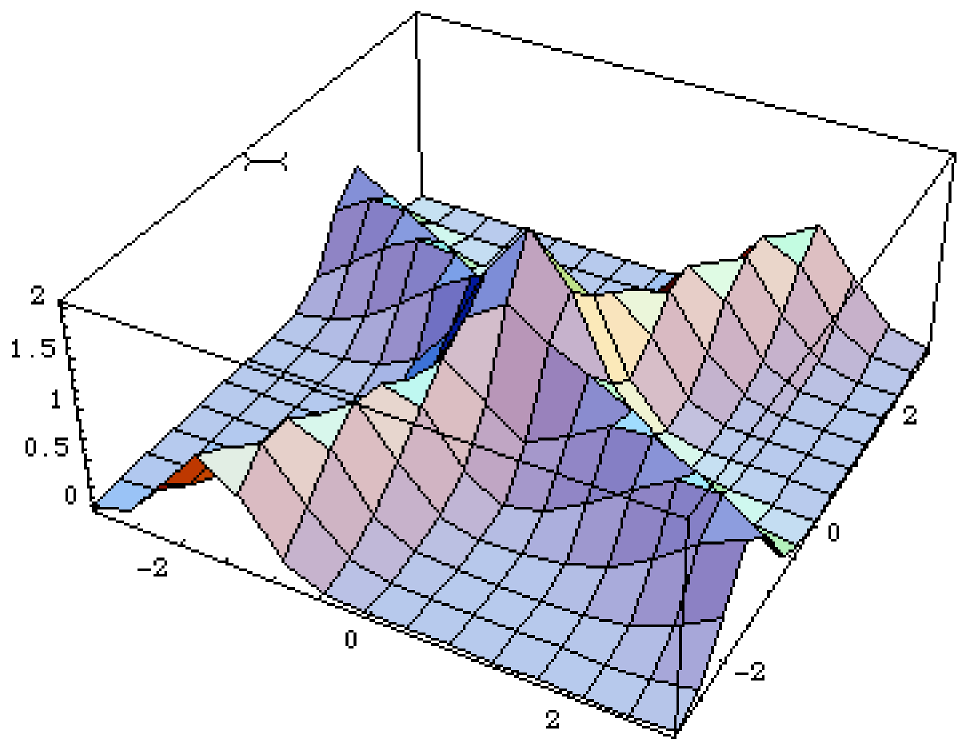
This is ok...

Importance Sampling

can be generalized to n dimensions:

$$p(\vec{x}) = p(x) \bullet p(y) \bullet p(z) \ldots$$

but the peaks of $f(\vec{x})$ need to be "aligned" to the axis!



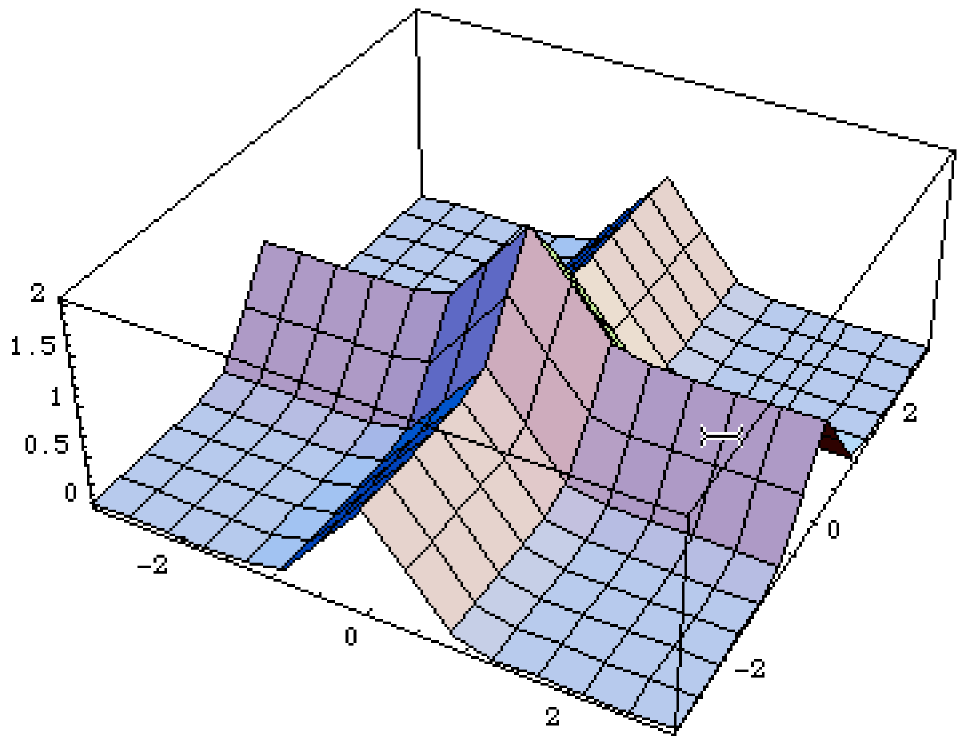
This is not ok...

IMPORTANCE SAMPLING

can be generalized to n dimensions:

$$p(\vec{x}) = p(x) \cdot p(y) \cdot p(z) \ldots$$

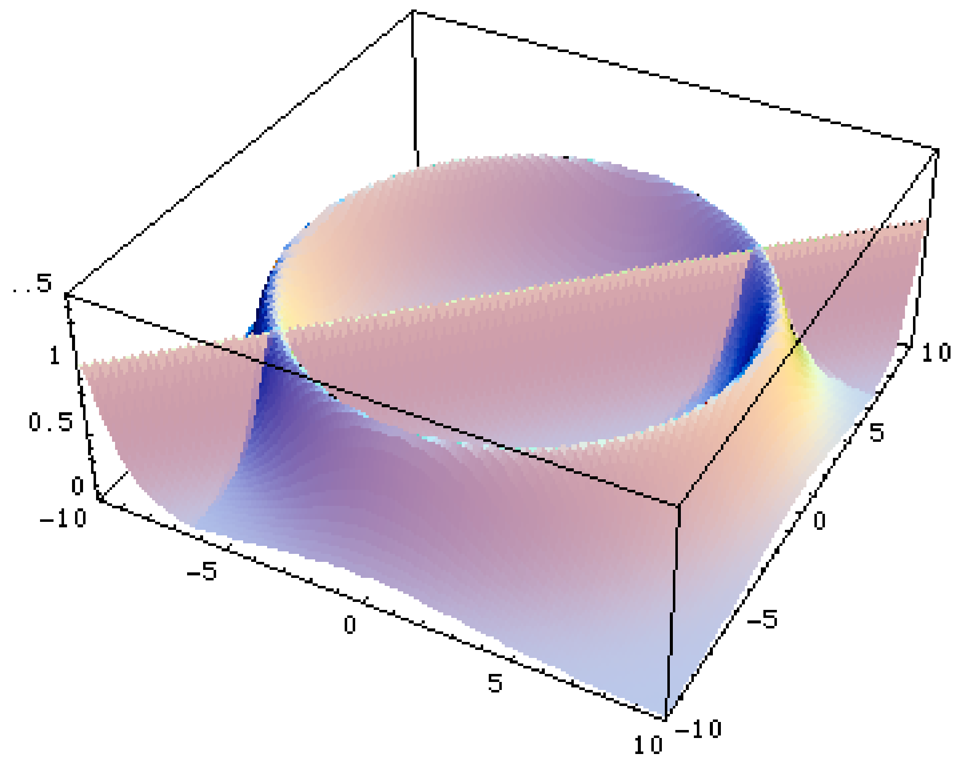
but the peaks of $f(\vec{x})$ need to be "aligned" to the axis!



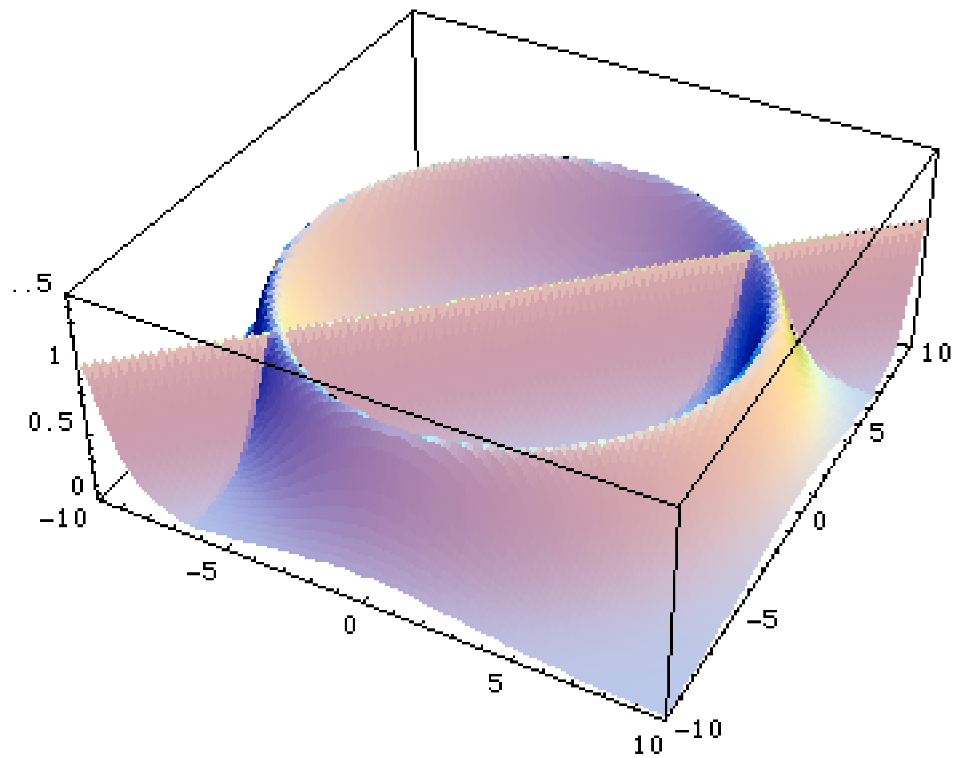
but it is sufficient to make a change of variables!

Multi-channel

Multi-channel

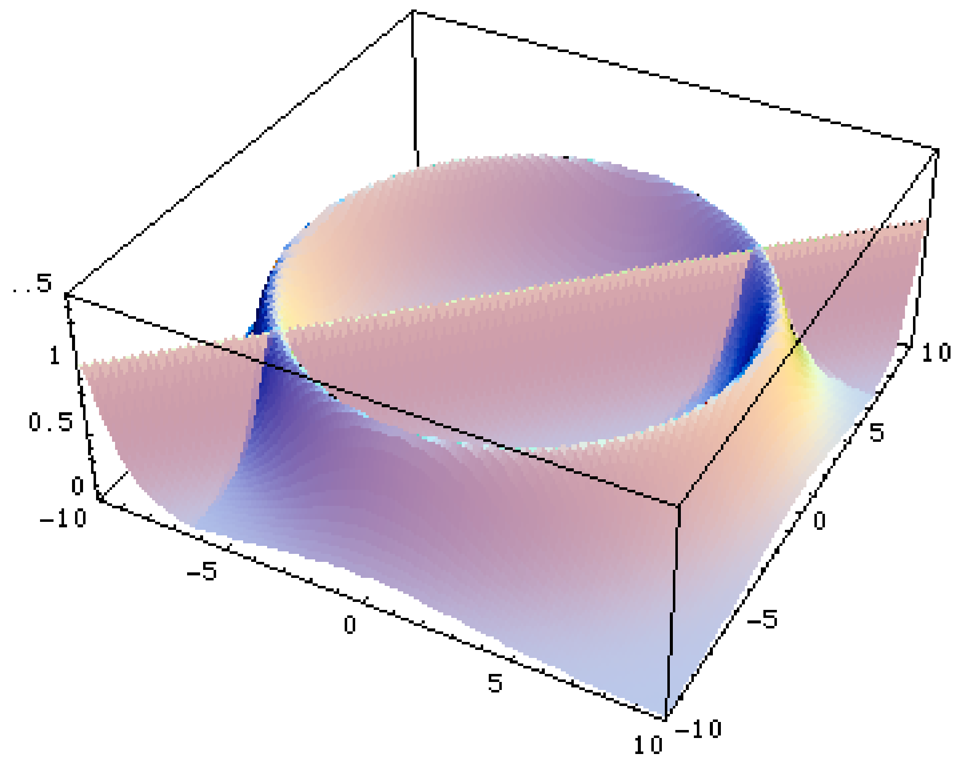


Multi-channel



In this case there is no unique tranformation: Vegas is bound to fail!

Multi-channel



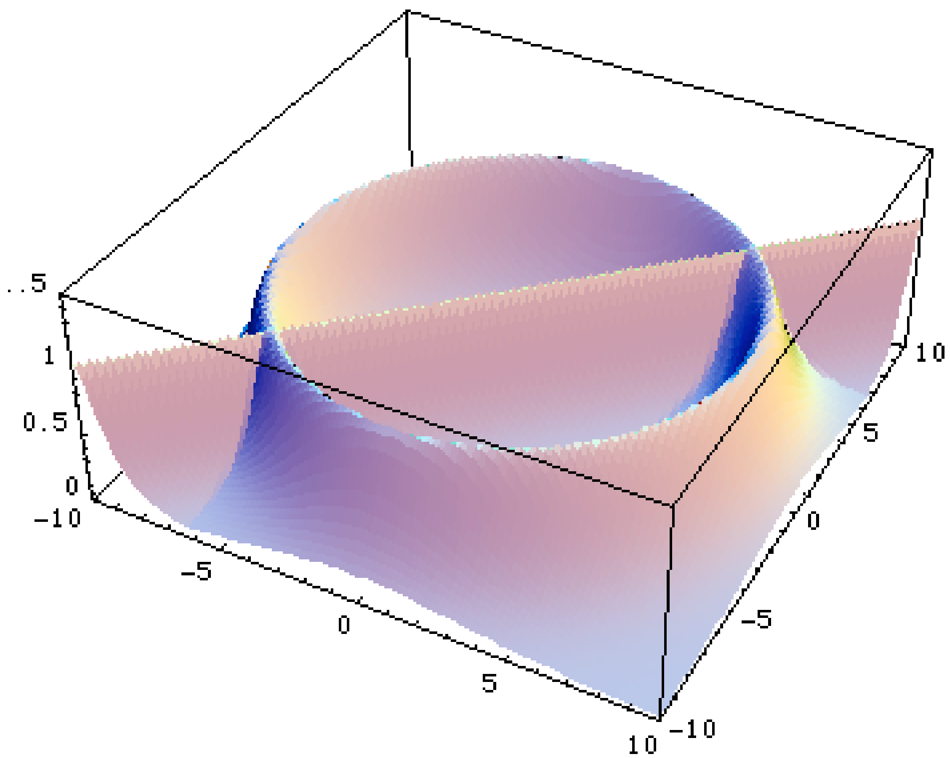
In this case there is no unique tranformation: Vegas is bound to fail!

Solution: use different transformations= channels

$$p(x) = \sum_{i=1}^{n} \alpha_i p_i(x) \qquad \text{with} \qquad \sum_{i=1}^{n} \alpha_i = 1$$

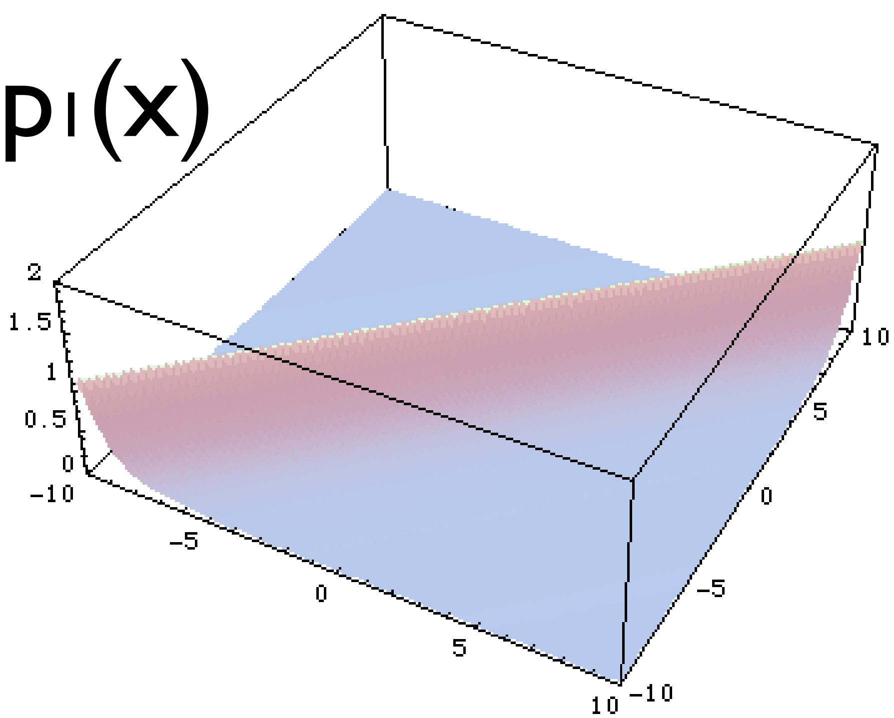
with each $p_i(x)$ taking care of one "peak" at the time

MULTI-CHANNEL

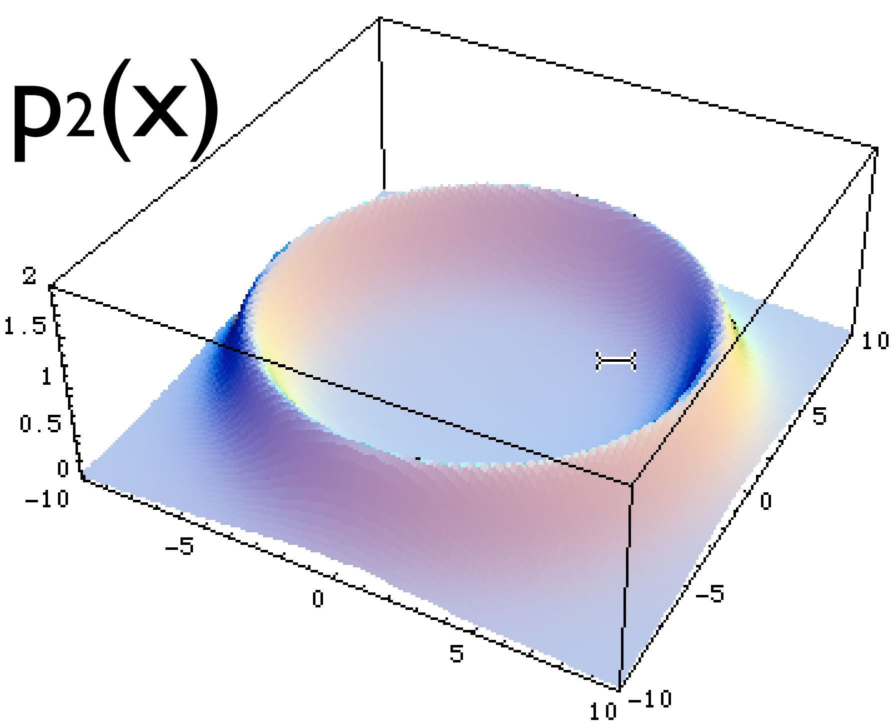


In this case there is no unique tranformation: Vegas is bound to fail!

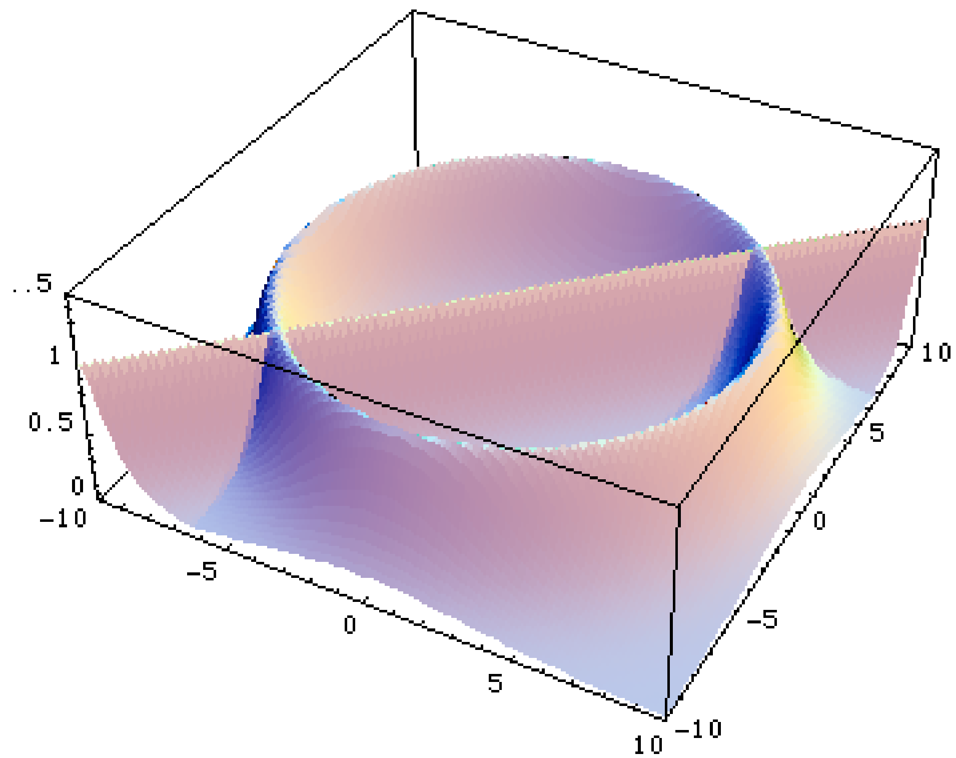
$p_1(x)$



$p_2(x)$



Multi-channel



In this case there is no unique tranformation: Vegas is bound to fail!

But if you know where the peaks are (=in which variables) we can use different transformations= channels:

$$p(x) = \sum_{i=1}^{n} \alpha_i p_i(x) \qquad \textbf{with} \qquad \sum_{i=1}^{n} \alpha_i = 1$$

$$I = \int f(x)dx = \sum_{i=1}^{n} \alpha_i \int \frac{f(x)}{p(x)} p_i(x) dx$$

Multi-channel

- Advantages
  - The integral does not depend on the α_i but the variance does and can be minimised by a careful choice
- Drawbacks
  - Need to calculate all g_i values for each point
  - Each phase space channel must be invertible
  - N coupled equations for α_i so it might only work for small number of channels

Multi-channel

- Advantages
  - The integral does not depend on the α_i but the variance does and can be minimised by a careful choice
- Drawbacks
  - Need to calculate all g_i values for each point
  - Each phase space channel must be invertible
  - N coupled equations for α_i so it might only work for small number of channels

Very popular method!

Multi-channel based on single diagrams

Consider the integration of an amplitude |M|^2 at treel level which lots of diagrams contribute to. If there were a basis of functions,

$$f = \sum_{i=1}^{n} f_i \quad \text{with} \quad f_i \geq 0\,, \quad \forall \;\; i\,,$$

such that:

1. we know how to integrate each one of them,
2. they describe all possible peaks,

then the problem would be solved:

$$I = \int d\vec{\Phi} f(\vec{\Phi}) = \sum_{i=1}^{n} \int d\vec{\Phi}\, g_i(\vec{\Phi})\, \frac{f_i(\vec{\Phi})}{g_i(\vec{\Phi})} = \sum_{i=1}^{n} I_i\,,$$

Multi-channel based on single diagrams

Consider the integration of an amplitude |M|^2 at treel level which lots of diagrams contribute to. If there were a basis of functions,

$$f = \sum_{i=1}^{n} f_i \quad \text{with} \quad f_i \geq 0\,, \quad \forall\ i\,,$$

such that:

1. we know how to integrate each one of them,
2. they describe all possible peaks,

then the problem would be solved:

$$I = \int d\vec{\Phi} f(\vec{\Phi}) = \sum_{i=1}^{n} \int d\vec{\Phi}\, g_i(\vec{\Phi}) \frac{f_i(\vec{\Phi})}{g_i(\vec{\Phi})} = \sum_{i=1}^{n} I_i\,,$$

Does such a basis exist?

Multi-channel based on single diagrams

Consider the integration of an amplitude |M|^2 at treel level which lots of diagrams contribute to. If there were a basis of functions,

$$f = \sum_{i=1}^{n} f_i \quad \text{with} \quad f_i \geq 0\,, \quad \forall\ i\,,$$

such that:

1. we know how to integrate each one of them,
2. they describe all possible peaks,

then the problem would be solved:

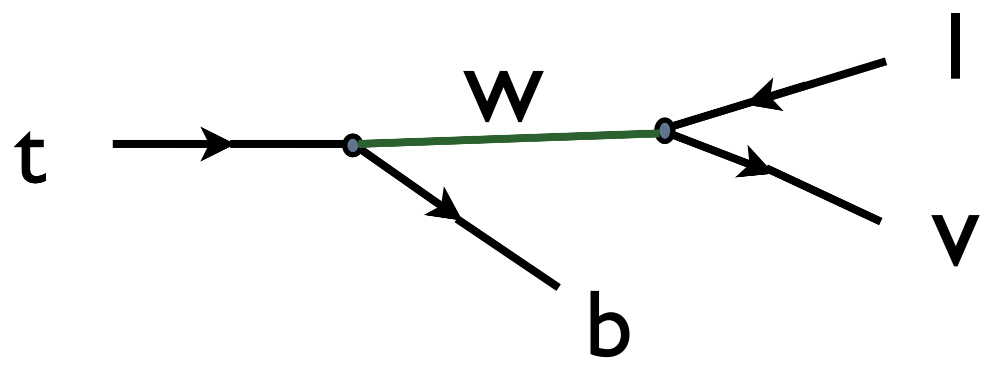
$$I = \int d\vec{\Phi} f(\vec{\Phi}) = \sum_{i=1}^{n} \int d\vec{\Phi}\, g_i(\vec{\Phi}) \frac{f_i(\vec{\Phi})}{g_i(\vec{\Phi})} = \sum_{i=1}^{n} I_i\,,$$

Does such a basis exist? YES! $f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} |A_{\rm tot}|^2$

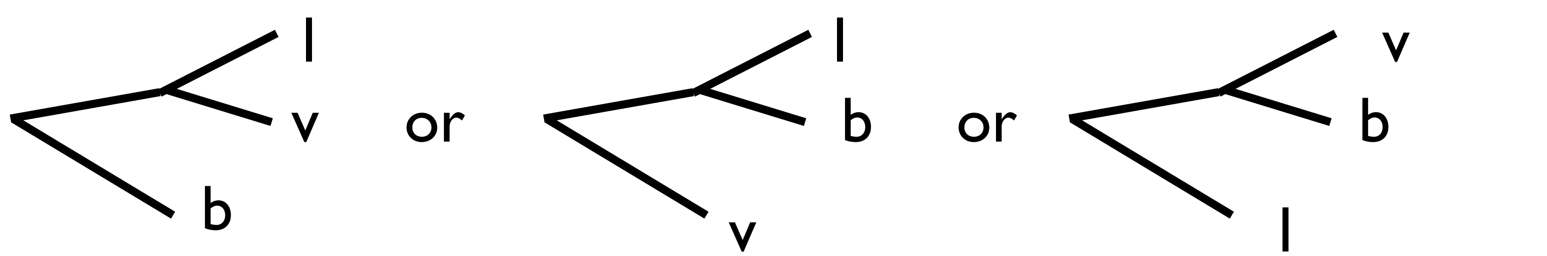
Multi-channel : MadGraph

- Key Idea
  - Any single diagram is "easy" to integrate
  - Divide integration into pieces, based on diagrams
- Get N independent integrals
  - Errors add in quadrature so no extra cost
  - No need to calculate "weight" function from other channels.
  - Can optimize # of points for each one independently
  - Parallel in nature
- What about interference?
  - Never creates "new" peaks, so we're OK!

Exercise: top decay

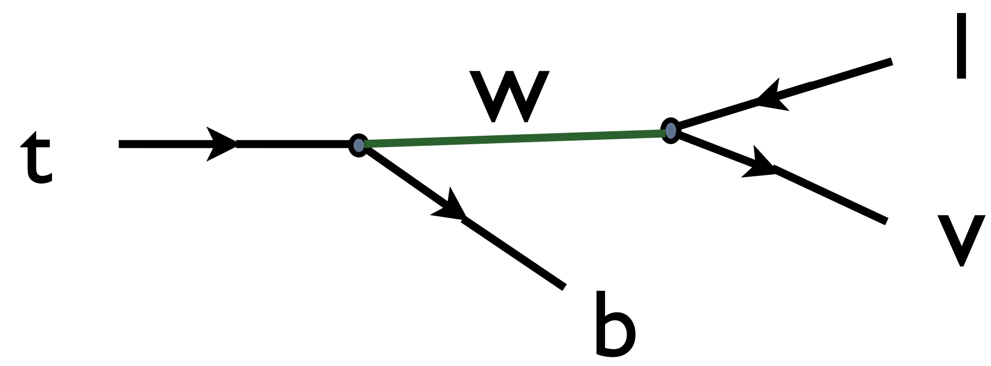


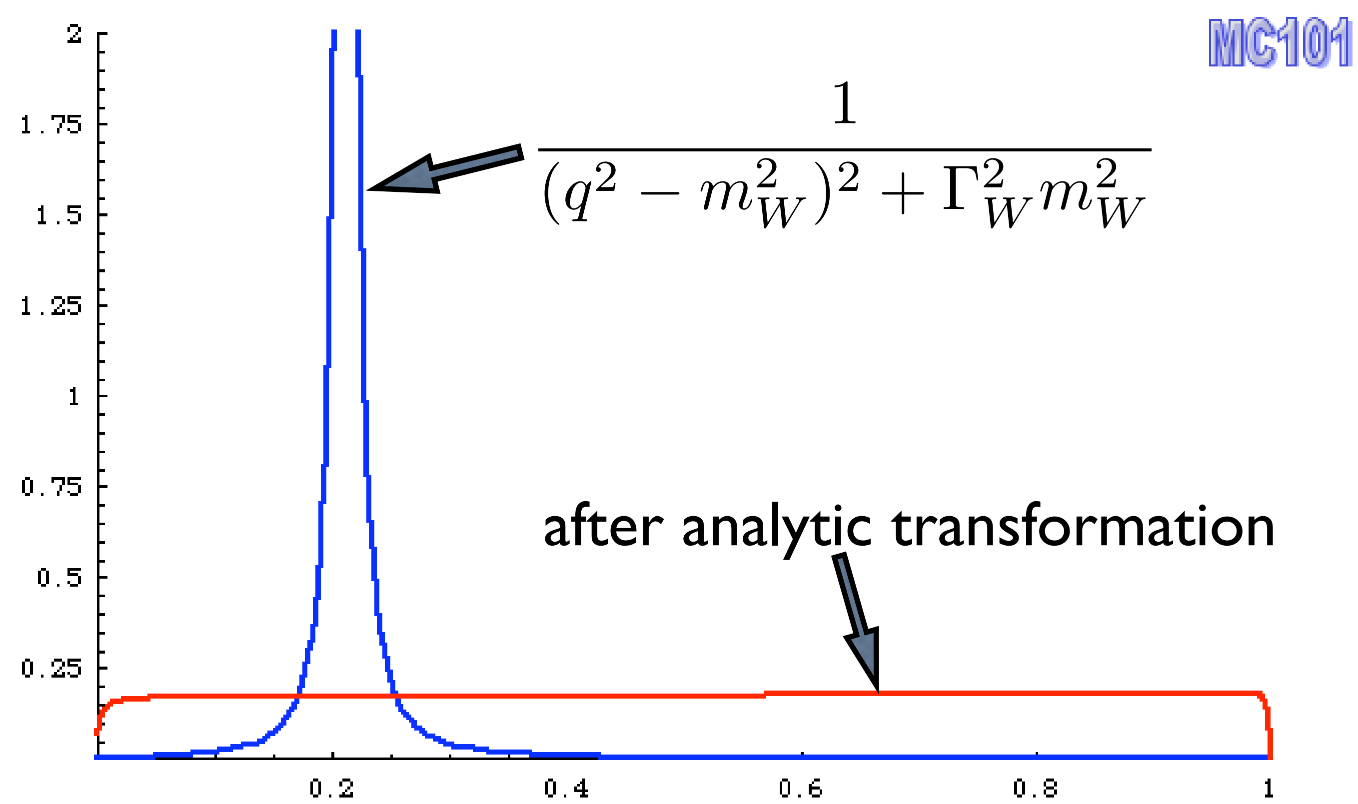
- Easy but non-trivial
- Breit-Wigner peak $\frac{1}{(q^2 - m_W^2)^2 + \Gamma_W^2 m_W^2}$ to be "flattened :
- Choose the right "channel" for the phase space



EXERCISE: TOP DECAY

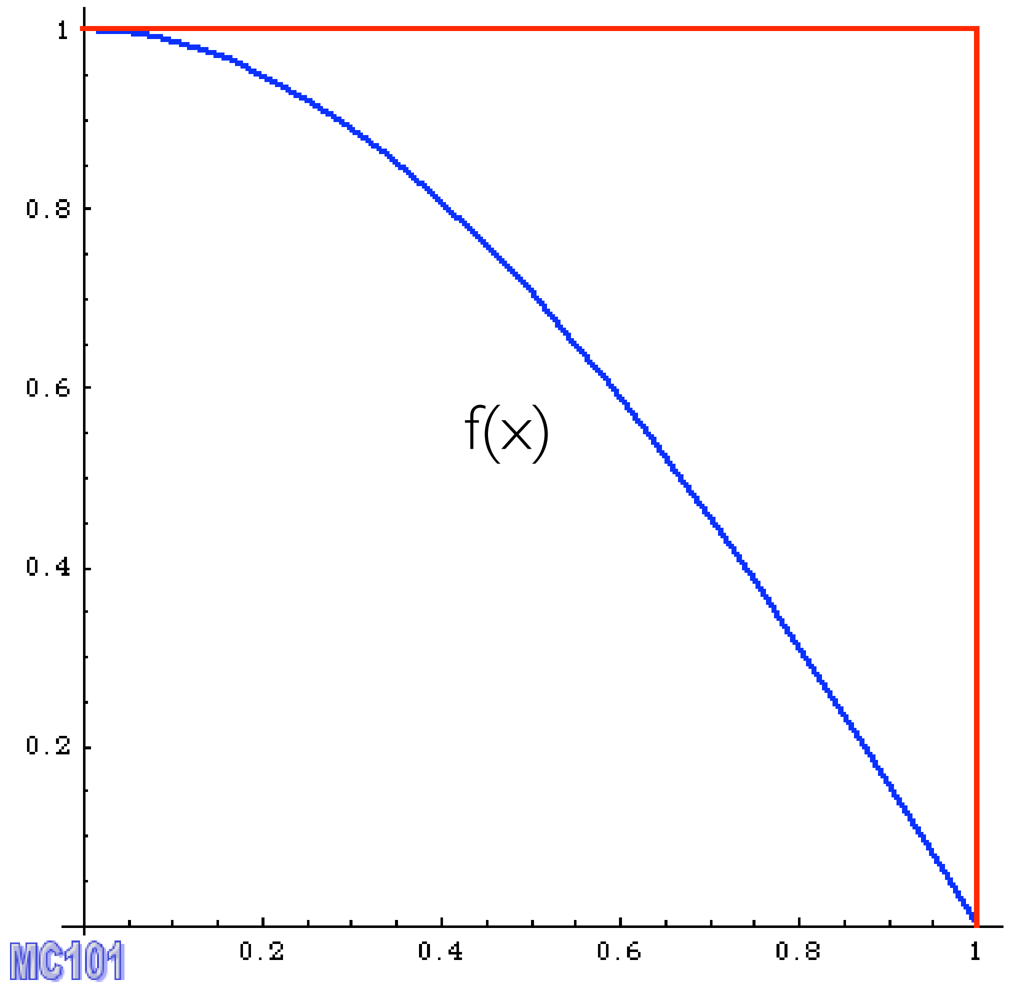
MC101





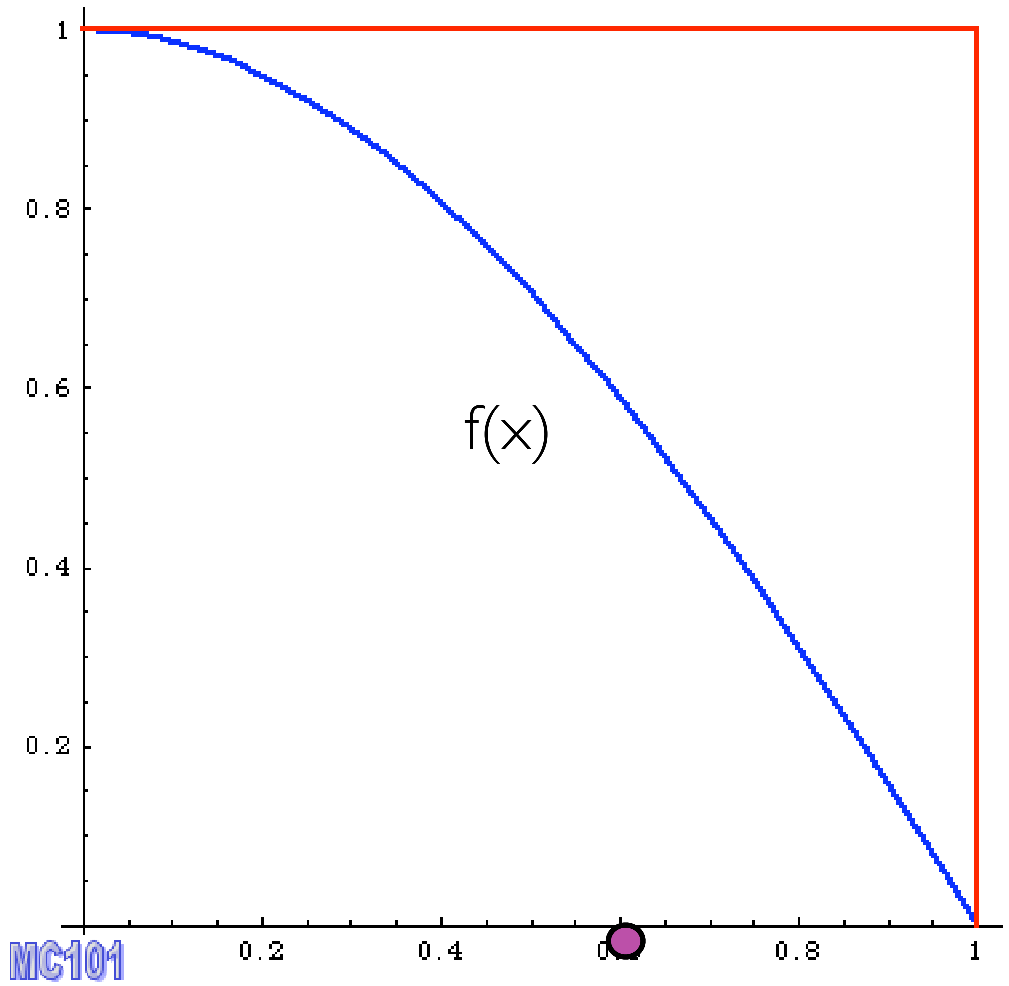
$$\frac{1}{(q^2 - m_W^2)^2 + \Gamma_W^2 m_W^2}$$

Event Generation



Alternative way

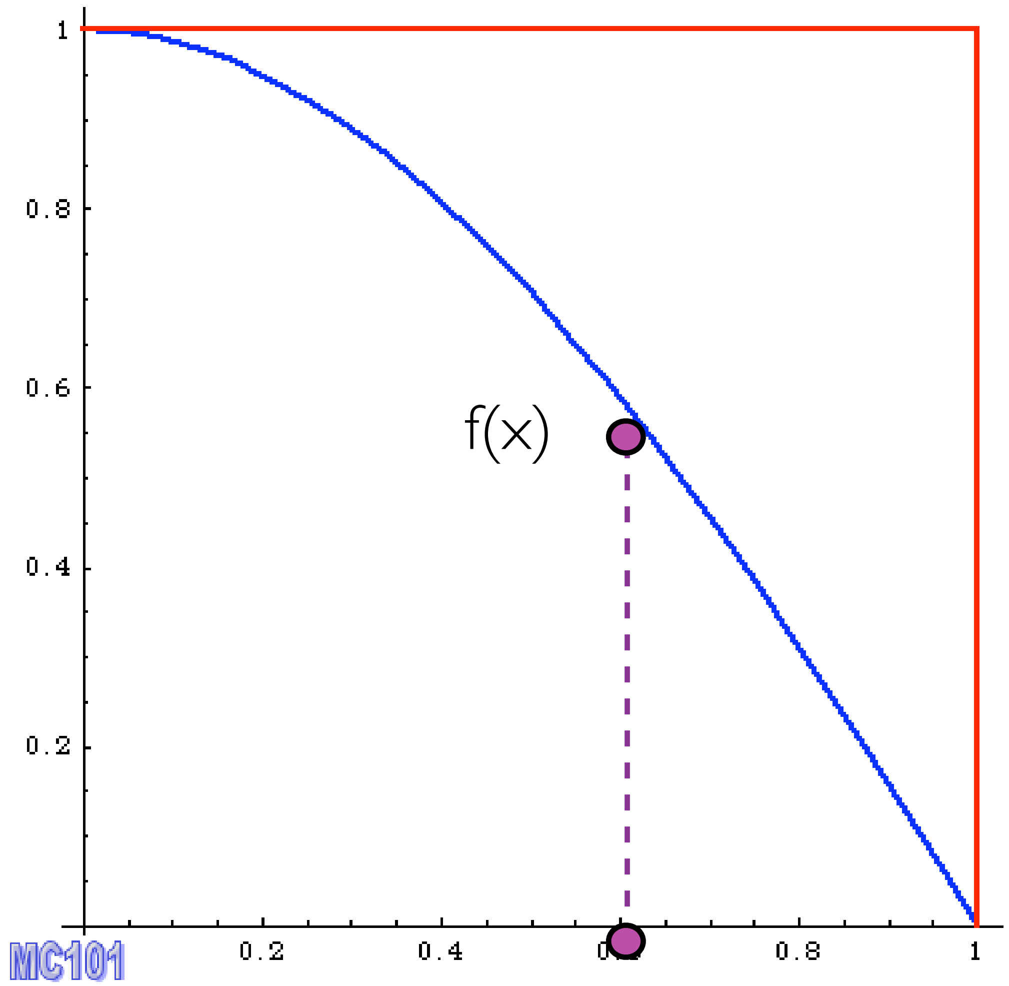
Event Generation



Alternative way

1. pick x

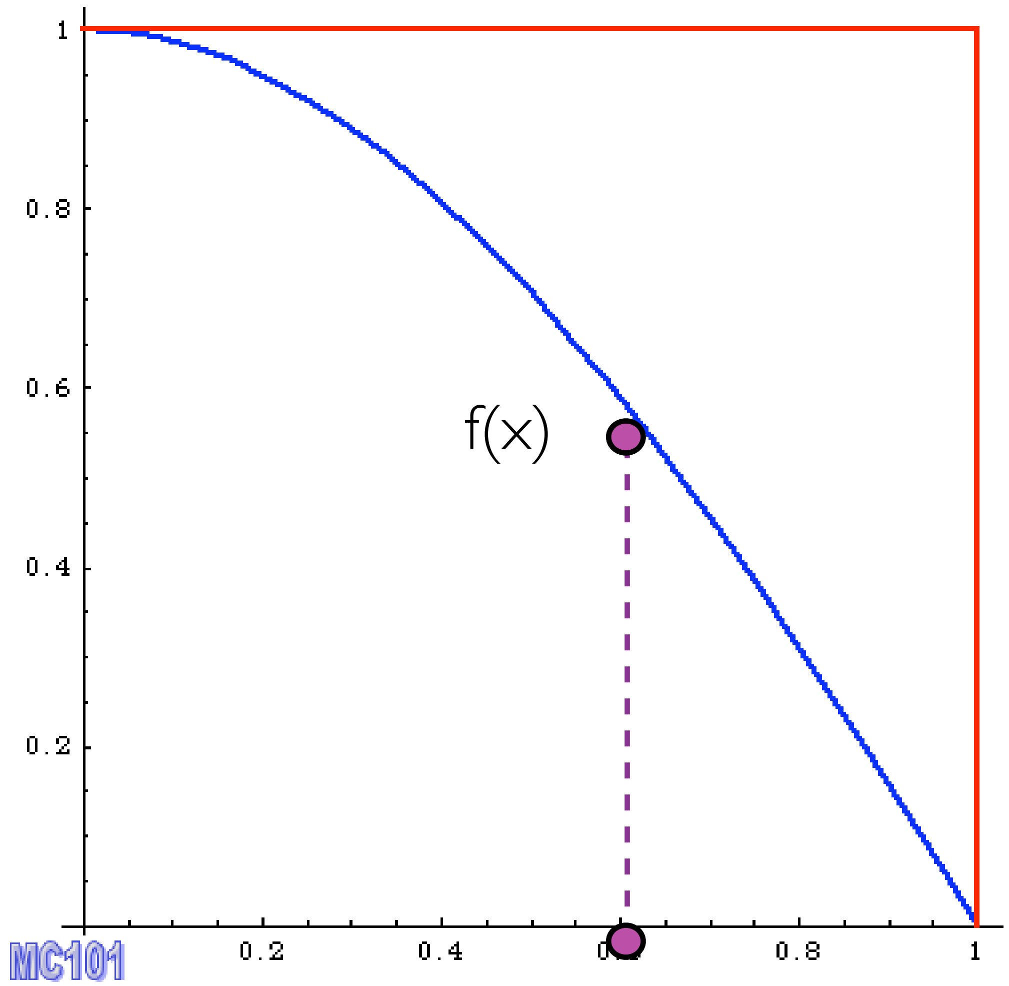
Event Generation



Alternative way

1. pick x
2. calculate f(x)

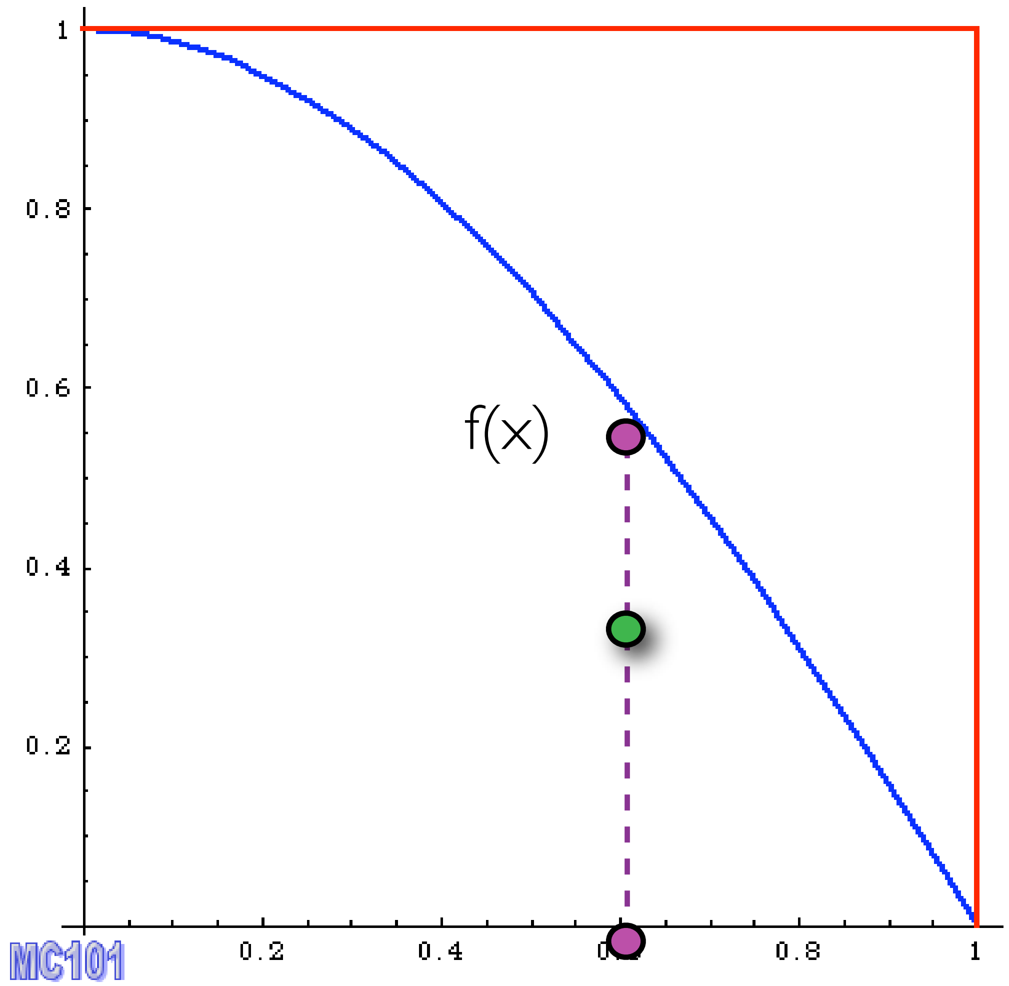
Event Generation



Alternative way

1. pick x
2. calculate f(x)
3. pick 0<y<fmax

Event Generation

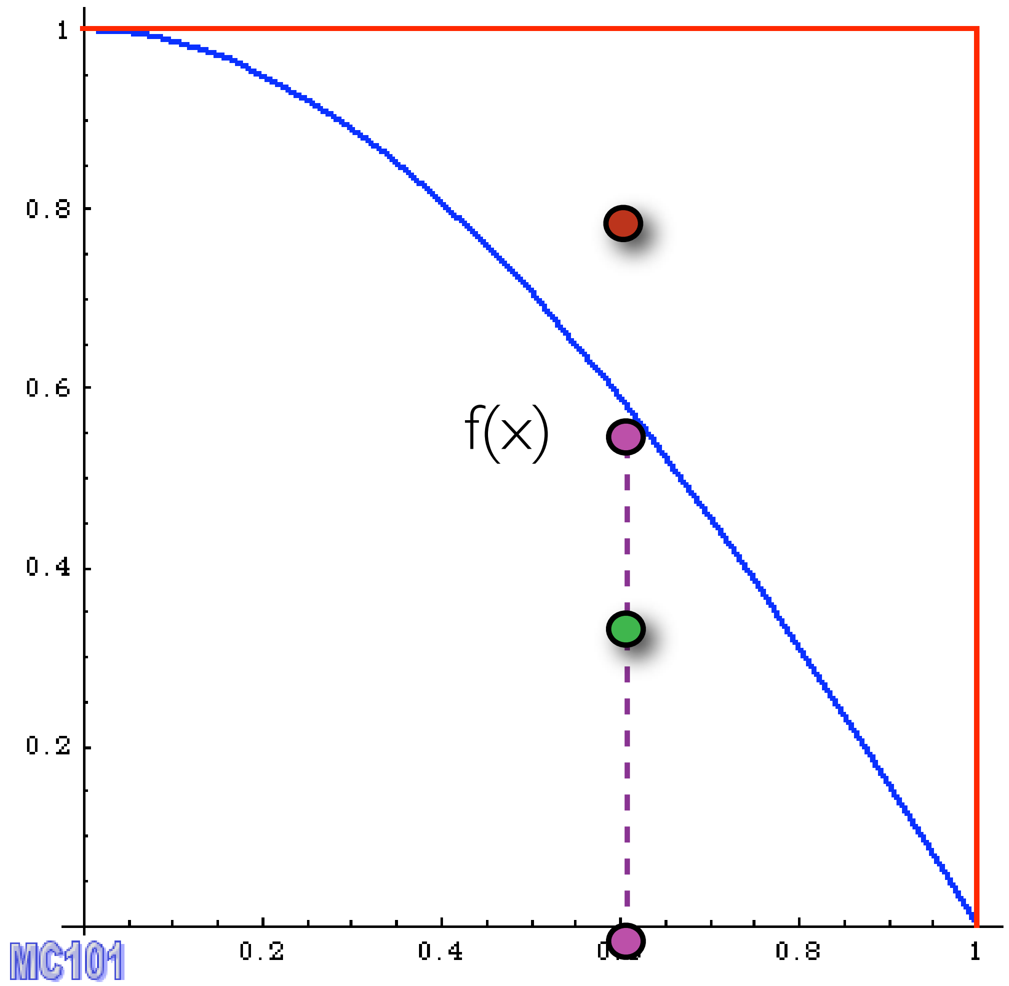


Alternative way

1. pick x
2. calculate f(x)
3. pick 0<y<fmax

4. Compare:
if f(x)>y accept event,

Event Generation

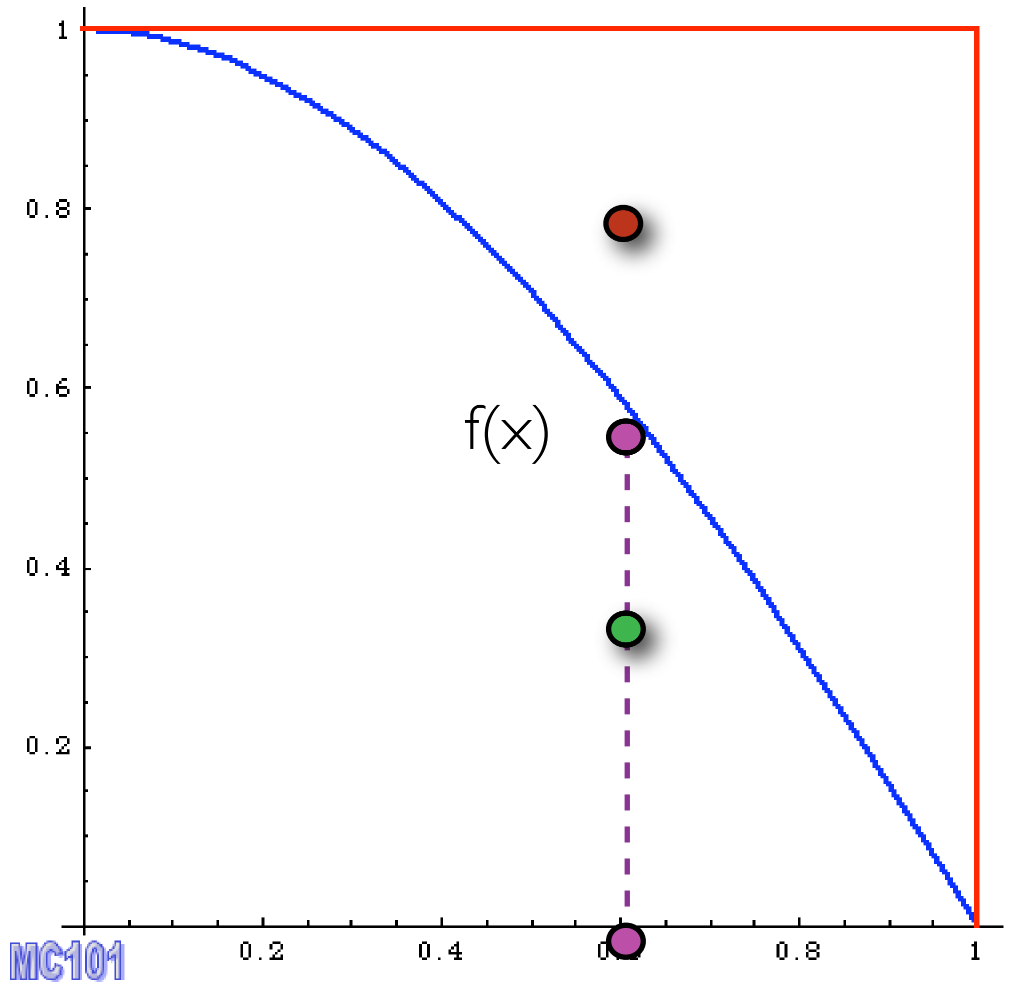


Alternative way

1. pick x
2. calculate f(x)
3. pick 0<y<fmax
4. **Compare:**
   if f(x)>y accept event,

   else reject it.

Event Generation



Alternative way

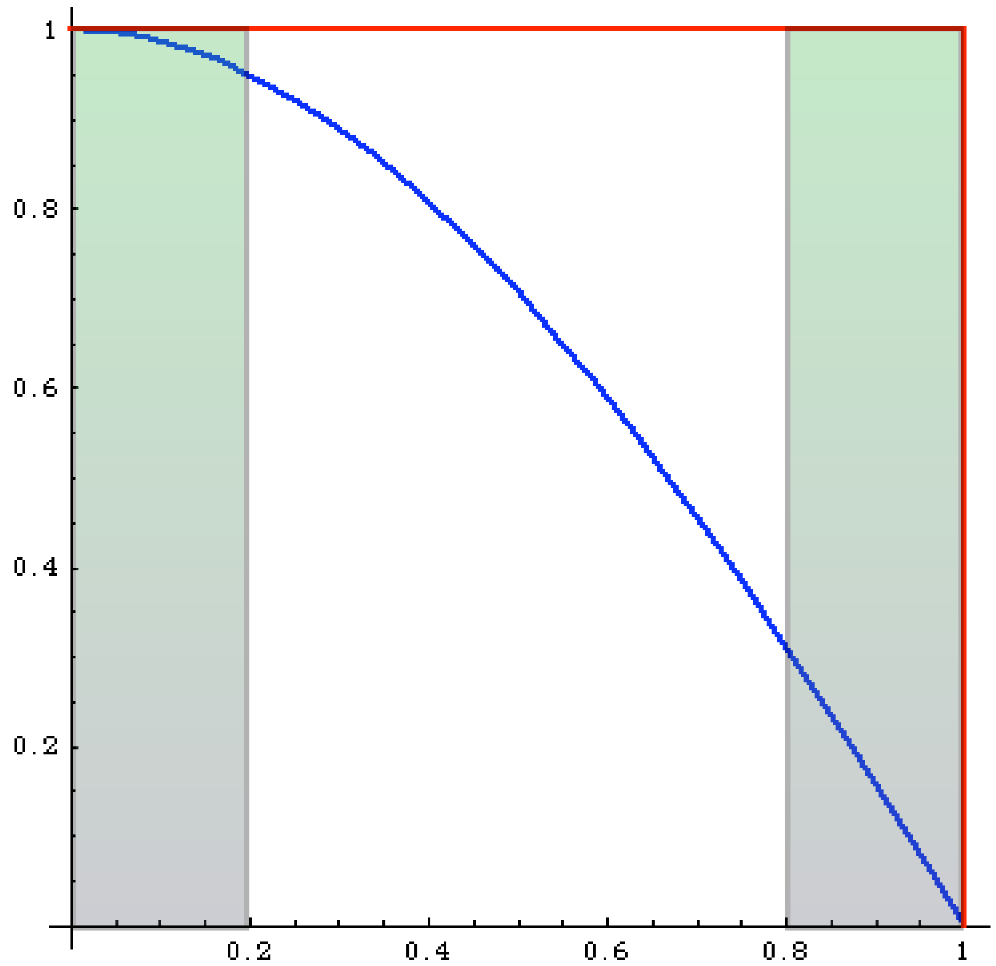
1. pick x
2. calculate f(x)
3. pick 0<y<fmax

4. Compare:
if f(x)>y accept event,

else reject it.

$$I = \frac{\text{accepted}}{\text{total tries}} = \text{efficiency}$$

Event Generation

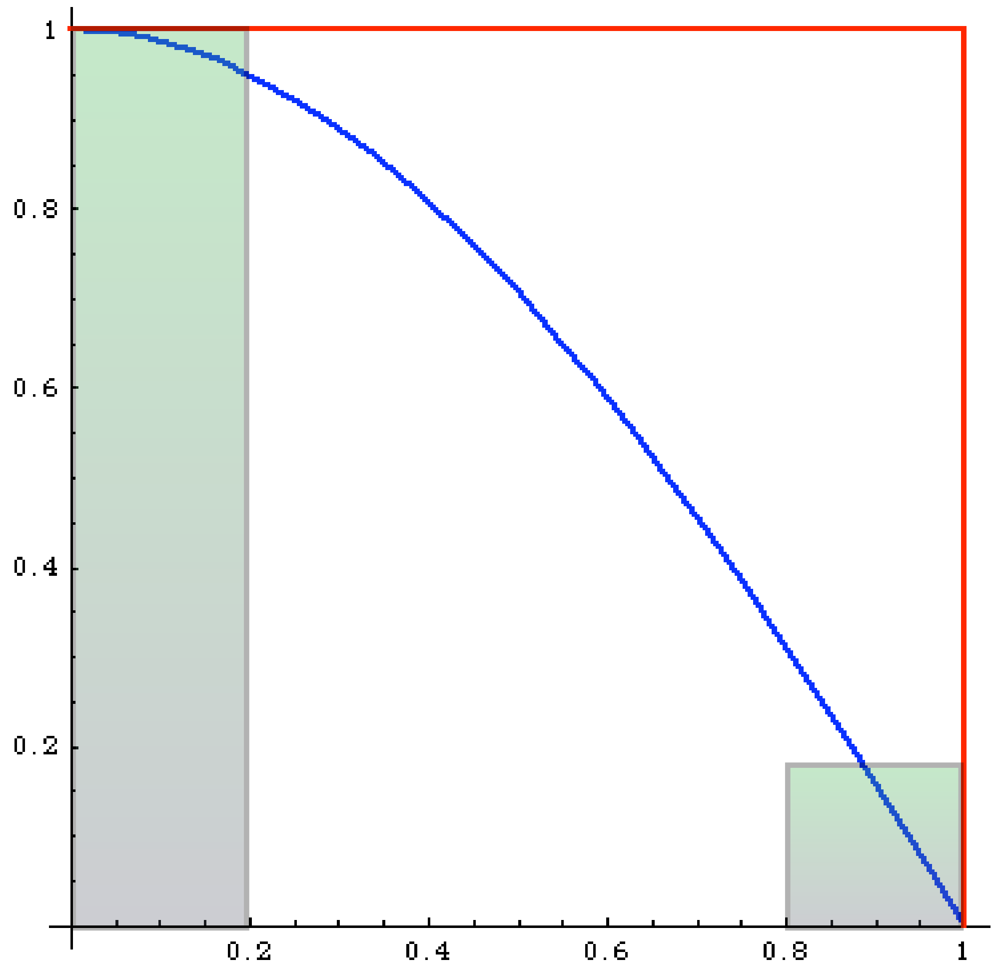


What's the difference?

before:

same # of events in areas of phase space with very different probabilities: events must have different weights

Event Generation



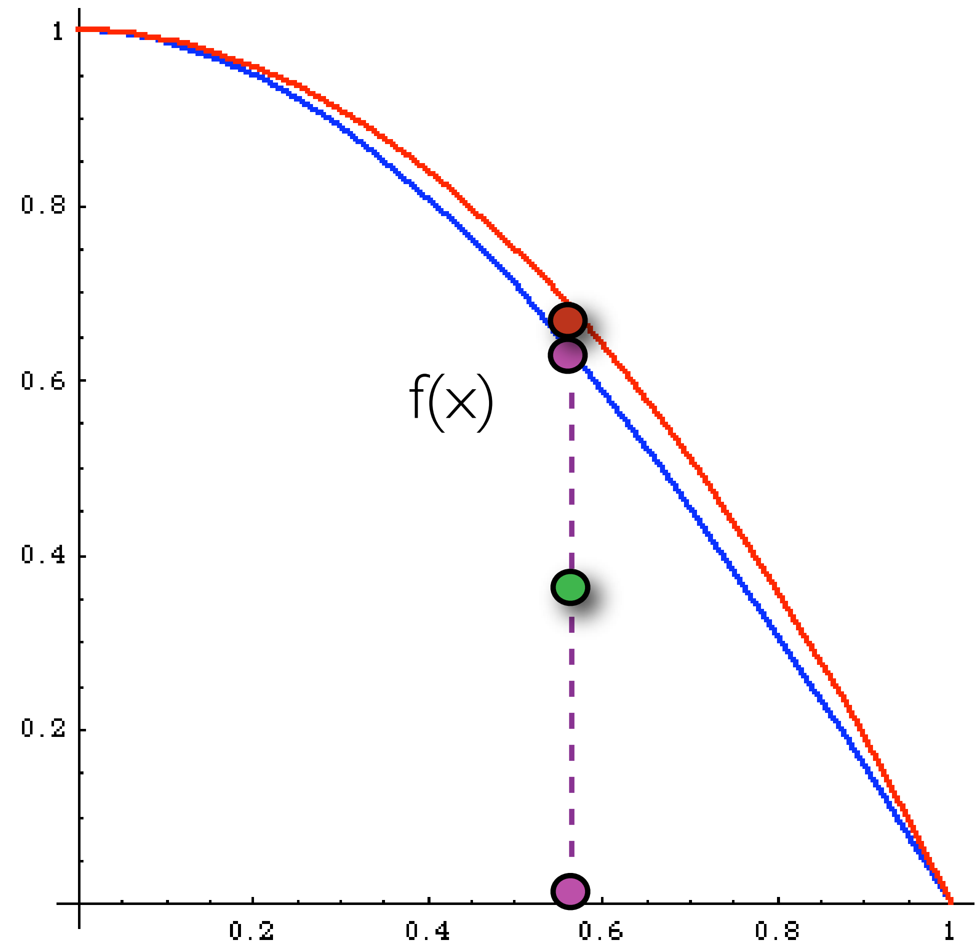
What's the difference?

before:

events is proportional to the probability of areas of phase space:
events have all the same weight ("unweighted")

Events distributed as in Nature

Event Generation



Improved

1. pick x distributed as p(x)
2. calculate f(x) and p(x)
3. pick 0<y<1
4. Compare:
   if f(x)>y p(x) accept event,

else reject it.

much better efficiency!!!

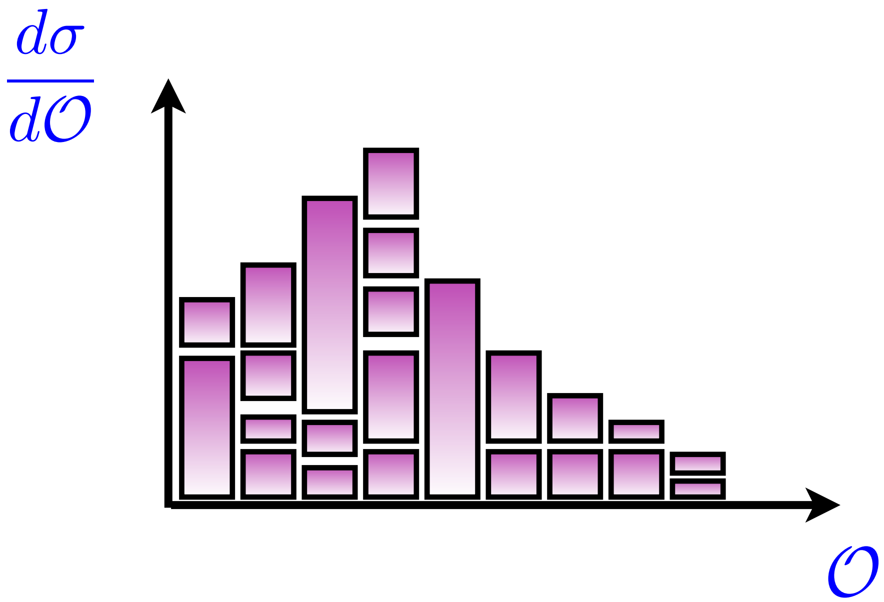
Event generation

Event generation

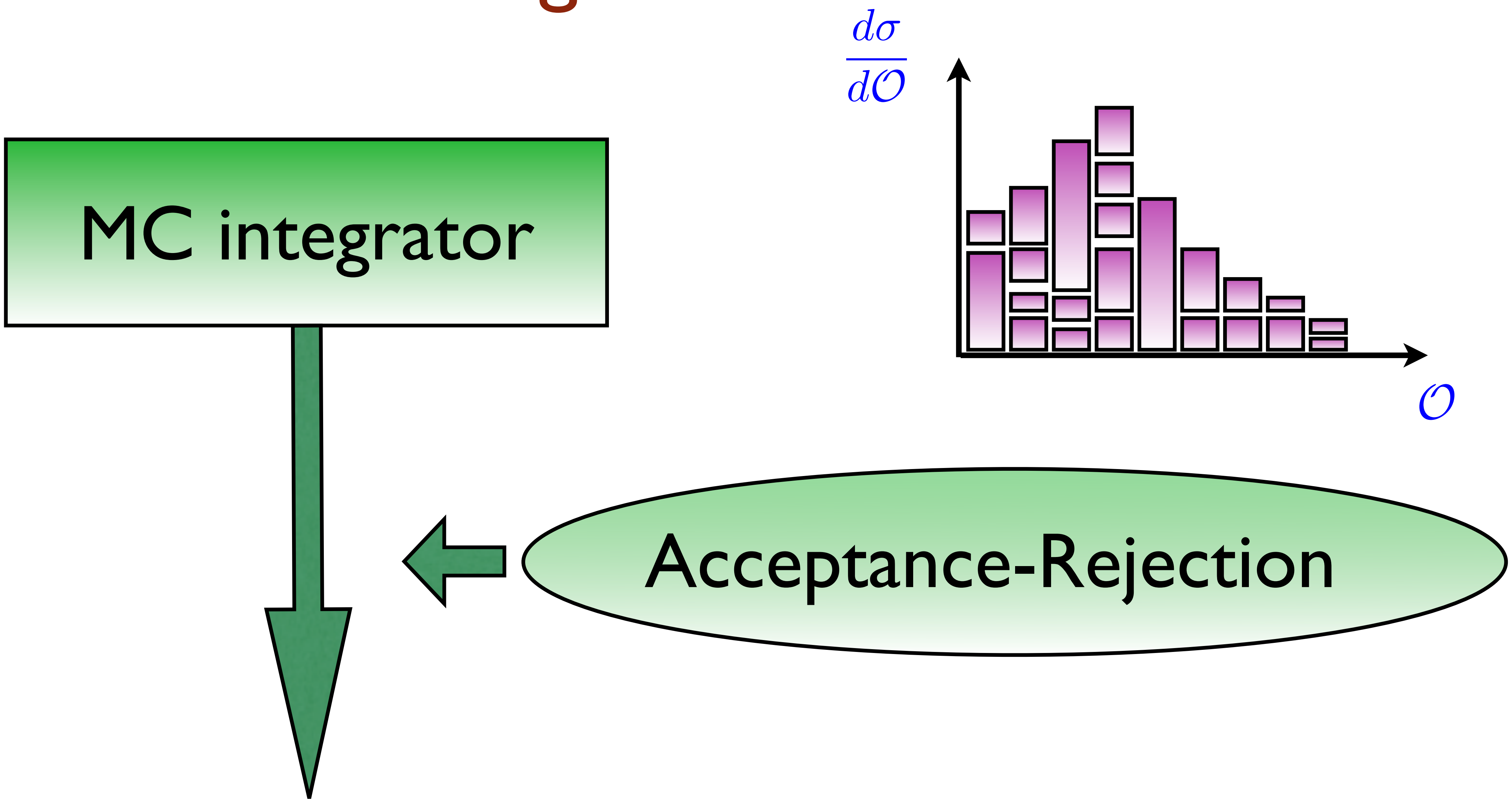
MC integrator

Event generation

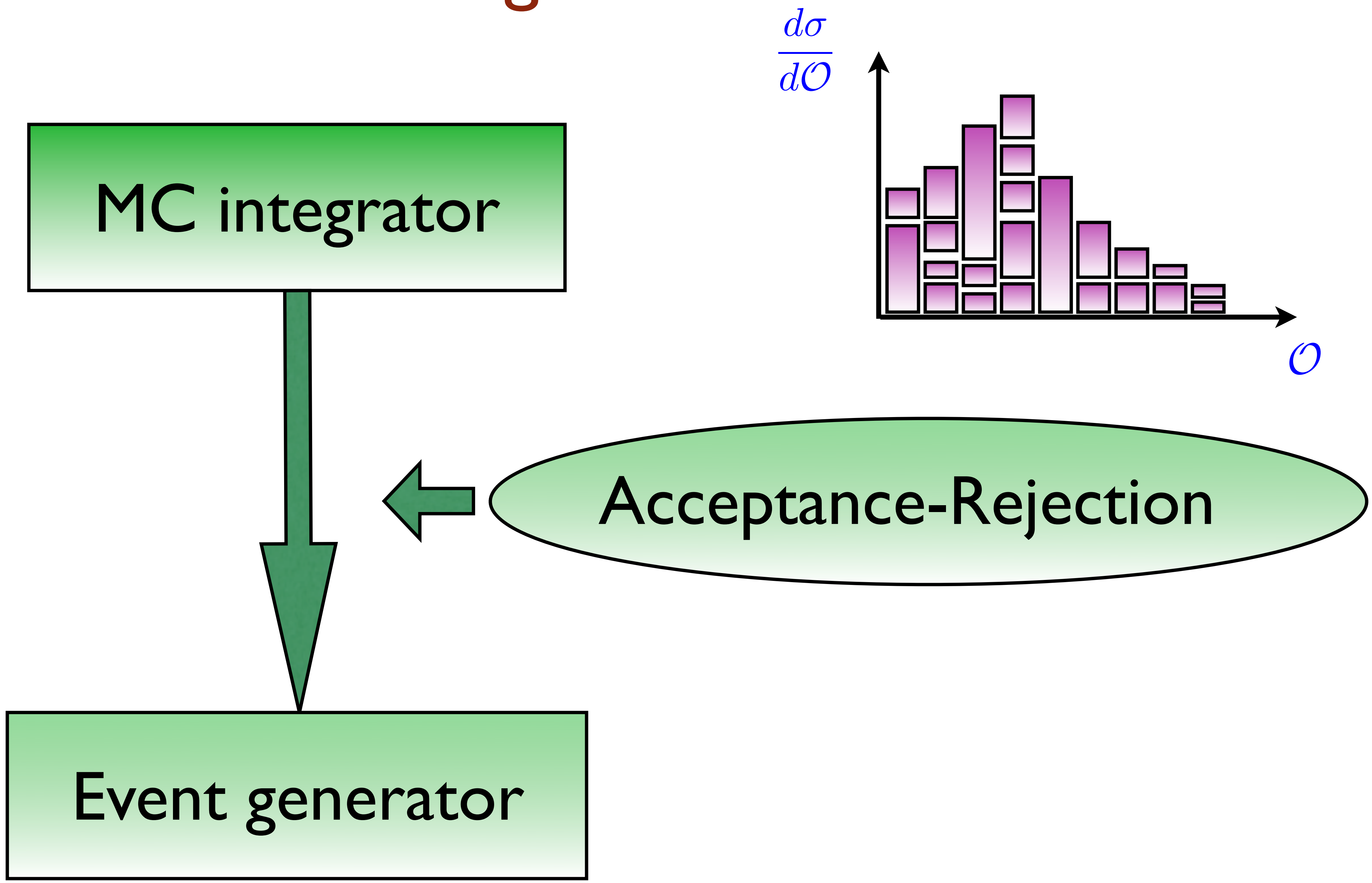
MC integrator



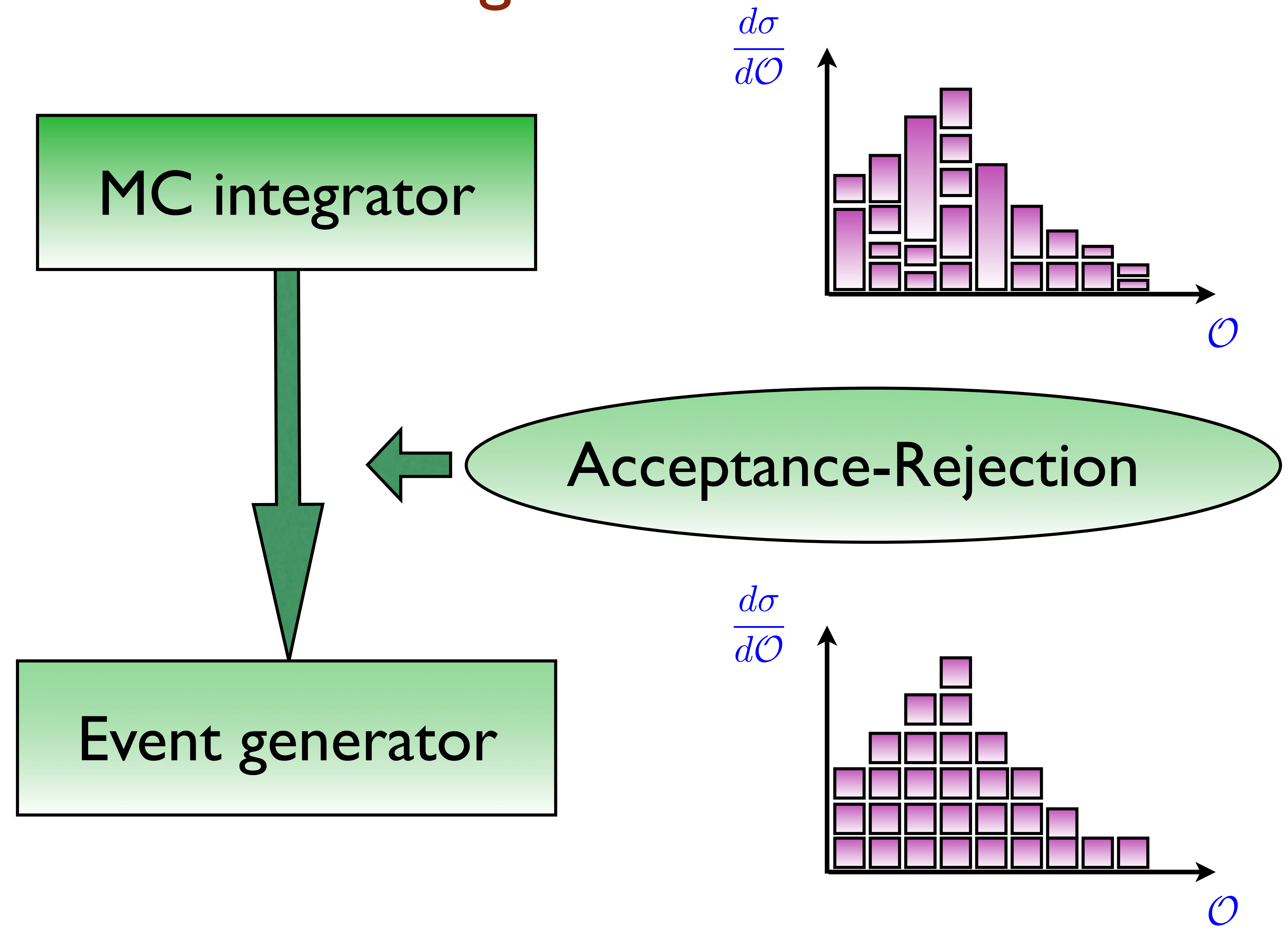
Event generation



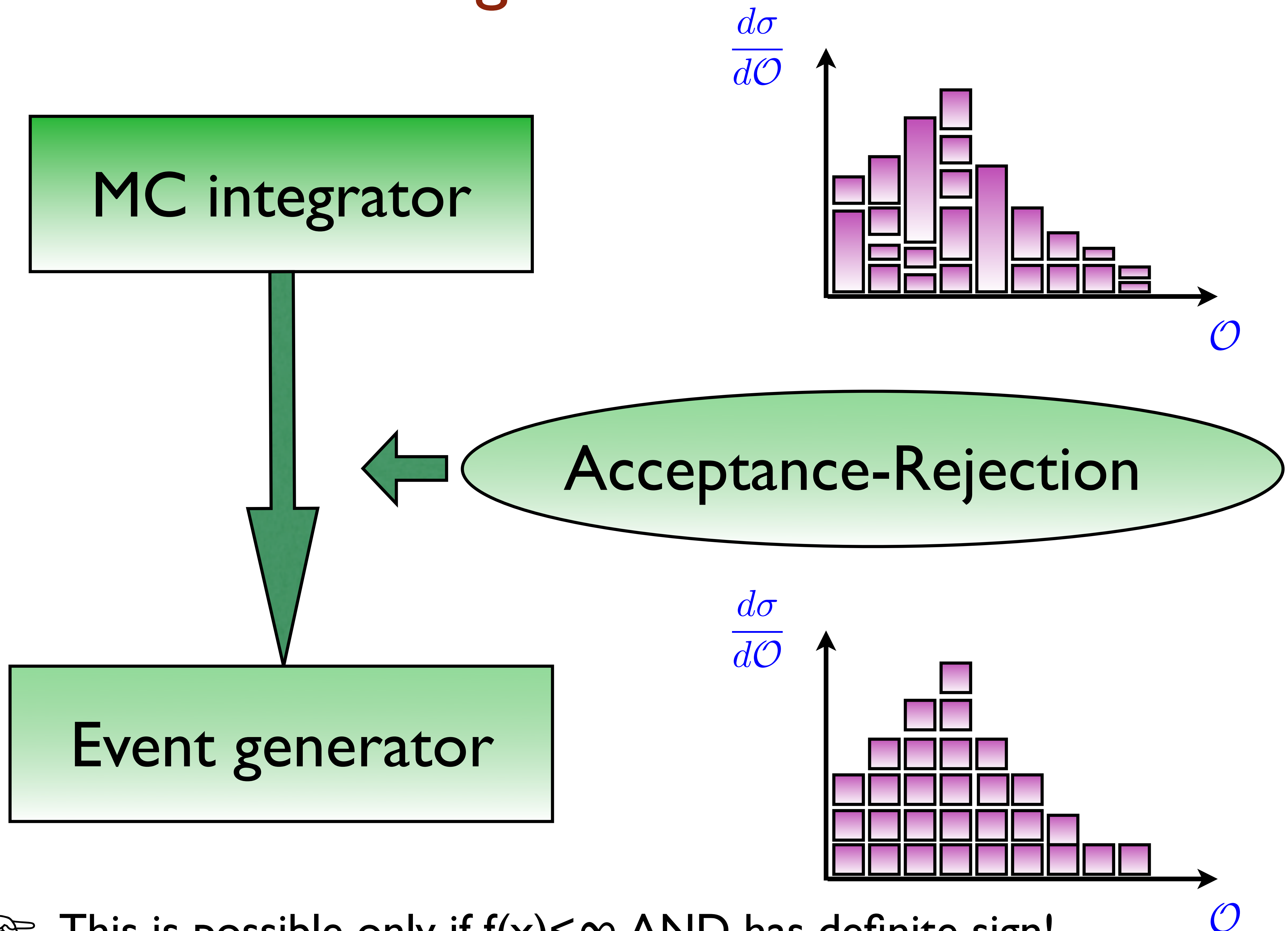
Event generation



Event generation



Event generation



☞ This is possible only if f(x)<∞ AND has definite sign!

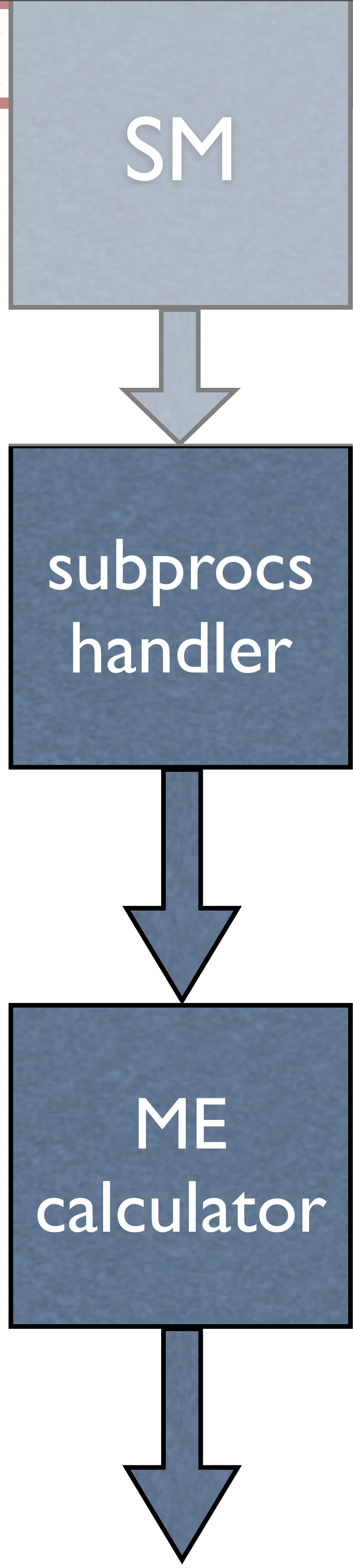
Monte Carlo Event Generator: definition

At the most basic level a Monte Carlo event generator is a program which produces particle physics events with the same probability as they occur in nature (virtual collider).

In practice it performs a large number of (sometimes very difficult) integrals and then unweights to give the four momenta of the particles that interact with the detector (simulation).

Note that, at least among theorists, the definition of a "Monte Carlo program" also includes codes which don't provide a fully exclusive information on the final state but only cross sections or distributions at the parton level, even when no unweighting can be performed. I will refer to these kind of codes as "MC integrators".

General structure

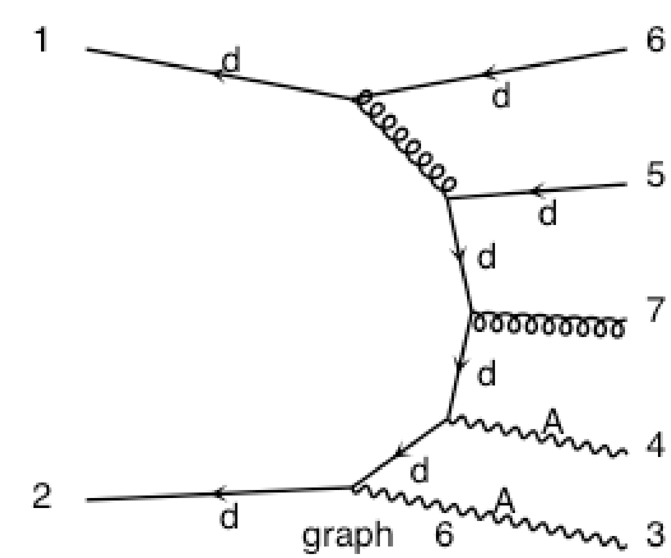


Includes all possible subprocess leading to a given multi-jet final state automatically or manually (done once for all)

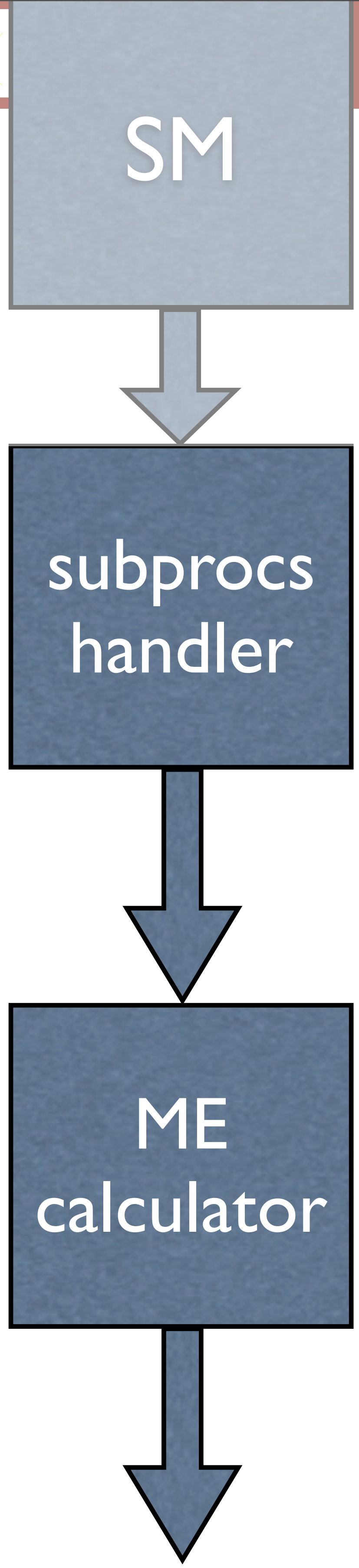
d~ d -> a a u u~ g
d~ d -> a a c c~ g
s~ s -> a a u u~ g
s~ s -> a a c c~ g

"Automatically" generates a code to calculate |M|^2 for arbitrary processes with many partons in the final state.

Most use Feynman diagrams w/ tricks to reduce the factorial growth, others have recursive relations to reduce the complexity to exponential. ☺



General structure

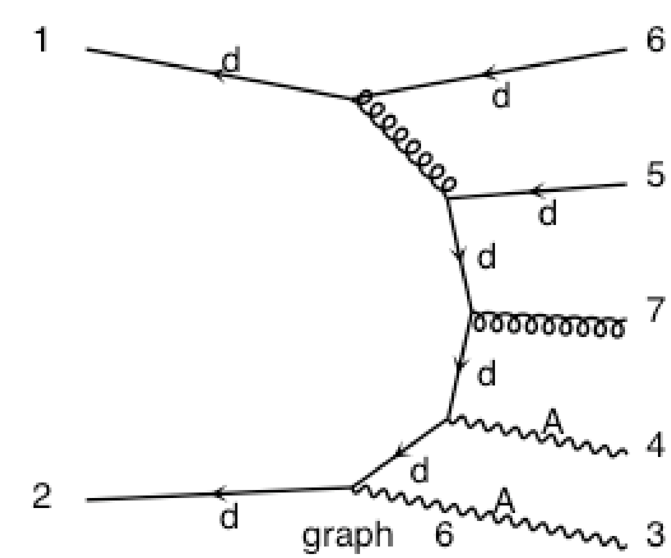


Includes all possible subprocess leading to a given multi-jet final state automatically or manually (done once for all)

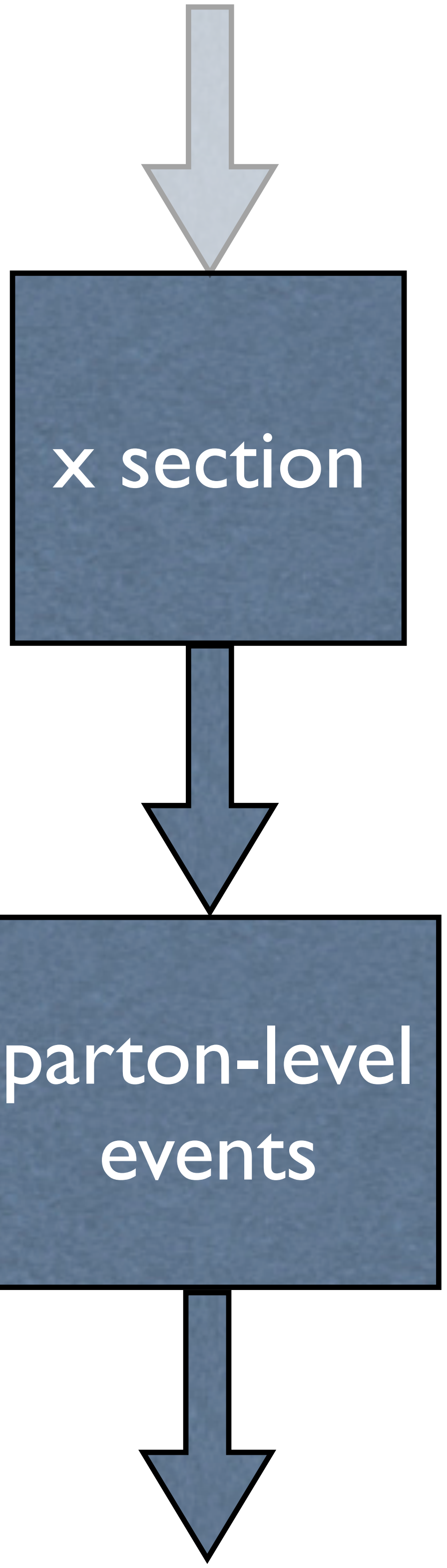
d~ d -> a a u u~ g
d~ d -> a a c c~ g
s~ s -> a a u u~ g
s~ s -> a a c c~ g

<u>"Automatically"</u> generates a code to calculate |M|^2 for arbitrary processes with many partons in the final state.

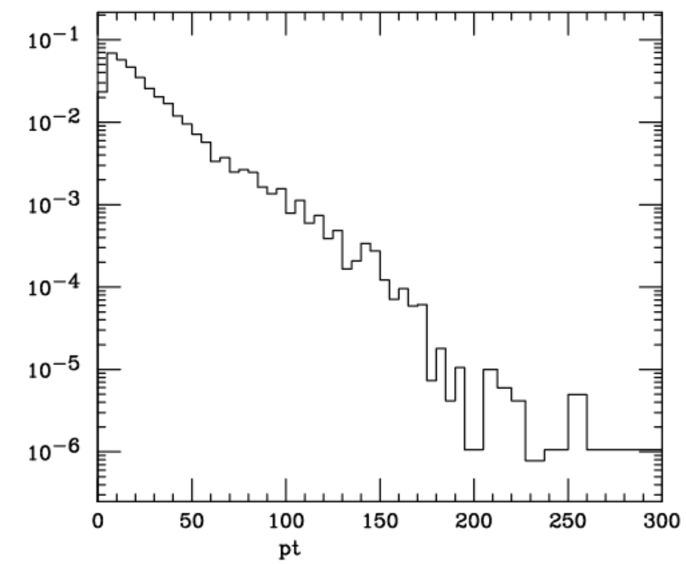
Most use Feynman diagrams w/ tricks to reduce the factorial growth, others have recursive relations to reduce the complexity to exponential. ☺



General structure



Integrate the matrix element over the phase space using a multi-channel technique and using parton-level cuts.



Events are obtained by unweighting. These are at the parton-level. Information on particle id, momenta, spin, color is given in the Les Houches format.