Introduction to Event Generators

Frank Krauss

Institute for Particle Physics Phenomenology
Durham University

DESY, 14.-18.3.2011

Modern usage of Monte Carlo event generators

Two ways of using MCs today:

- to generate distributions that look sufficiently close to data to allow for detector calibration etc.
  $\longrightarrow$ there, no real theory input is needed!
- to extrapolate from a background to a signal region
  $\longrightarrow$ there, you better rely on underlying theory!

Goal of the lectures

In these lectures, I aim to convince you that event generators

- are "proper" theory tools, based on clearly defined physical paradigms and ideas; (which you may find in textbooks)
- are only successful in **consistently** describing data in a meaningful way if their theory inputs are **physically sound**;
- can be analysed and divided into aspects where we fully understand every approximation (matrix elements, parton showers, merging thereof), and into others, where we rely on heavy modelling (hadronization, underlying event);
- that the latter must be tuned, whereas the former should not need too much tuning.

Topics of the lectures

1. Lecture 1: *The Monte Carlo Principle*
2. Lecture 2: *Parton level event generation*
3. Lecture 3: *Dressing the Partons*
4. Lecture 4: *Modelling beyond Perturbation Theory & Improving the showers*

Thanks to

- My fellow MC authors, especially S.Gieseke, K.Hamilton, L.Lonnblad, F.Maltoni, M.Mangano, P.Richardson, M.Seymour, T.Sjostrand, B.Webber.
- the other Sherpas: J.Archibald, T.Gleisberg, S.Höche, S.Schumann, F.Siegert, M.Schönherr, and J.Winter.

Menu of lecture 1

- Prelude: Selecting from a distribution
- Standard textbook numerical integration (quadratures)
- Monte Carlo integration
- A basic simulation example

Prelude: Selecting from a distribution

The problem

- A typical Monte Carlo/simulation problem:
  Distribution of "usual" random numbers #:
  "flat" in [0, 1].
- But: Want random numbers $x \in [x_{\min}, x_{\max}]$,
  distributed according to (probability) density $f(x)$.

The exact solution

- The first method applies if both the integral of the density $f(x)$ and its inverse are known (i.e. practically never).
- To see how it works realise that the

  diff. probability $\mathcal{P}(x \in [x', x' + \mathrm{d}x']) = f(x')\mathrm{d}x'$.
- Therefore: x given by

$$\int\limits_{x_{\rm min}}^{x} \mathrm{d}x' f(x') = \# \int\limits_{x_{\rm min}}^{x_{\rm max}} \mathrm{d}x' f(x').$$

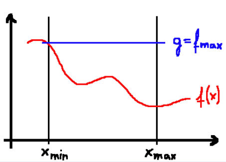
- Since everything known:

$$x = F^{-1}\left[F(x_{\rm min}) + \#\left(F(x_{\rm max}) - F(x_{\rm min})\right)\right].$$

The work-around solution: "Hit-or-miss"

(Solution, if exact case does not work.)

- Builds on "over-estimator" $g(x)$ (G and G^{-1} known):
$$g(x) > f(x) \ \ \forall x \in [x_{\min},\, x_{\max}].$$
- Select an x according to g (with exact algorithm);
- Accept with probability $f(x)/g(x)$ (with another random number);
- Obvious fall-back choice for $g(x)$:
$$g(x) = \mathrm{Max}_{[x_{\min},\, x_{\max}]}\{f(x)\}.$$



Quadratures: standard numerical integration

Reminder: Basic techniques

- Typical problem: Need to evaluate an integral, cannot do it in closed form.
- Example: nonlinear pendulum.
  Can calculate period T from E.o.M. $\ddot{\theta} = -g/l \sin\theta$:

$$T = \sqrt{\frac{8l}{g}} \int\limits_0^{\theta_{\max}} \frac{\mathrm{d}\theta}{\sqrt{\cos\theta - \cos\theta_{\max}}}$$

  Elliptic integral, no closed solution known
  $\Longrightarrow$ entering (again) the realm of numerical solutions.

Numerical integration: Newton-Cotes method

- Nomenclature now: Want to evaluate $I_f^{(a,b)} = \int\limits_a^b \mathrm{d}x f(x)$.
- Basic idea: Divide interval $[a, b]$ in N subintervals of size $\Delta x = (b-a)/N$ and approximate
$$I_f^{(a,b)} = \int\limits_a^b \mathrm{d}x f(x) \approx \sum_{i=0}^{N-1} f(x_i)\Delta x = \sum_{i=0}^{N-1} f(a + i\Delta x)\Delta x,$$
i.e. replace integration by sum over rectangular panels.
- Obvious issue: What is the error? How does it scale parametrically with "step-size" (or, better, number of function calls)? Answer: It is linear in Δx.

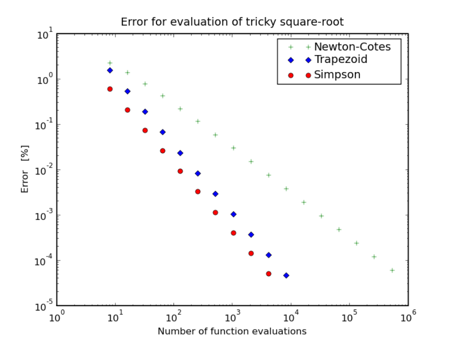
Improving on the error: Trapezoid, Simpson and all that

- A careful error estimate suggests that by replacing rectangles with trapezoids the error can be reduced to quadratic in Δx.
- This boils down to including a term $[f(b) - f(a)]/2$:
$$I_f^{(a,b)} \approx \sum_{i=1}^{N-1} f(x_i)\Delta x + \frac{\Delta x}{2}\left[f(a) + f(b)\right]$$
- Repeating the error-reducing exercise replaces the trapezoids by parabola: Simpson rule. In so doing, the error decreases to $(\Delta x)^4$.

Numerical integration: Results

- Consider test function $f(x) = \sqrt{4-x^2}$ in $[0,2]$.

$(I_f^{(0,2)} = \int\limits_0^2 \mathrm{d}x \sqrt{4-x^2} = \pi)$.



Convergence of numerical integration: Summary

- First observation: Numerical integrations only yield estimators of the integral, with an estimated accuracy given by the error.

(Proviso: the function is sufficiently well behaved.)

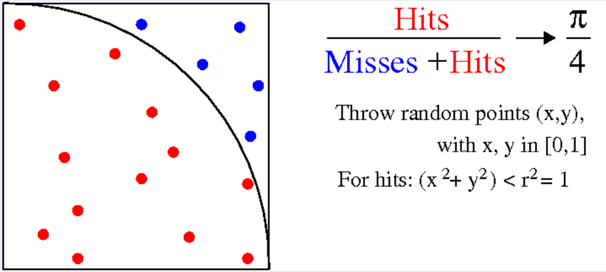
- Scaling behaviour of the error translates into scaling behaviour for the number of function calls necessary to achieve a certain precision.
- In one dimension/per dimension, therefore, the convergence scales like
  - Trapezium rule: $\simeq 1/N^2$
  - Simpson's rule $\simeq 1/N^4$

  with the number N of function calls.

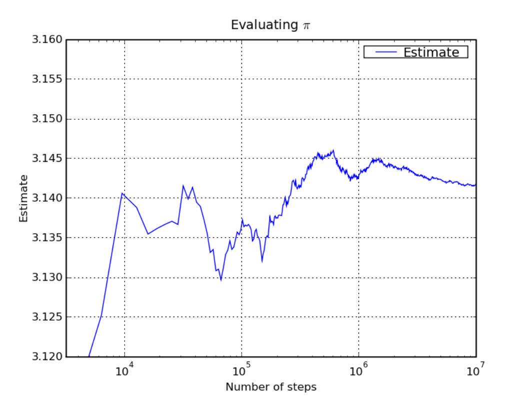
Monte Carlo integration

The underlying idea: Determination of π

- Use random number generator!



Determination of π



Error estimate in Monte Carlo integration

- MC integration: Estimate integral by N probes

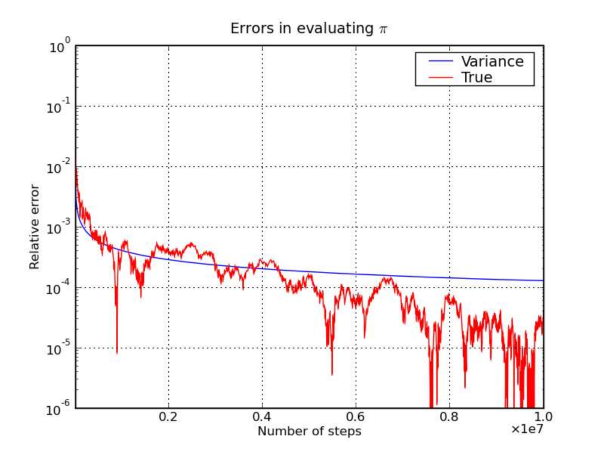
$$
\begin{aligned}
I_f^{(a,b)} &= \int\limits_a^b \mathrm{d}x f(x) \\
\longrightarrow \quad \langle I_f^{(a,b)} \rangle &= \frac{b-a}{N} \sum_{i=1}^{N} f(x_i) = \langle f \rangle_{a,b},
\end{aligned}
$$

  where x_i homogeneously distributed in $[a,\, b]$
- Basic idea for error estimate: statistical sample
  $\Longrightarrow$ use standard deviation as error estimate

$$
\langle E_f^{(a,b)}(N) \rangle = \sigma = \left[\frac{\langle f^2 \rangle_{a,b} - \langle f \rangle_{a,b}^2}{N} \right]^{1/2}.
$$

- Independent of the number of integration dimensions!
  $\Longrightarrow$ Method of choice for high-dimensional integrals.

Determination of π: Errors



Improve convergence: Importance sampling

- Want to minimise number of function calls.

  (They are potentially CPU-expensive.)

  $\Longrightarrow$ Need to improve convergence of MC integration.

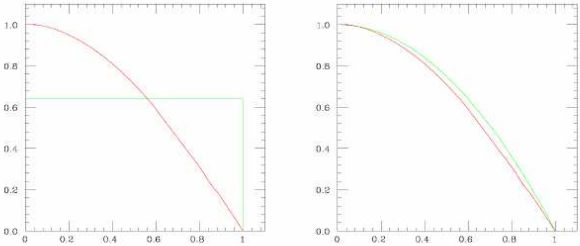
- First basic idea: Samples in regions, where f largest

  ($\Longrightarrow$ corresponds to a Jacobian transformation of integral.)

- Algorithm:
  - Assume a function $g(x)$ similar to $f(x)$.
  - Obviously $f(x)/g(x)$ is smooth $\Longrightarrow \langle E(f/g)\rangle$ is small.
  - Must sample according to $\mathrm{d}x\, g(x)$ rather than $\mathrm{d}x$: $g(x)$ plays role of probability distribution; we know already how to deal with this!
- Works, if $f(x)$ is well-known. Hard to generalise.

Importance sampling: Example results

- Consider $f(x) = \cos\frac{\pi x}{2}$ and $g(x) = 1 - x^2$:



$$
\begin{aligned}
I &= \int_0^1 dx\, \cos\tfrac{\pi}{2}x \\
&= 0.637 \pm 0.308/\sqrt{N}
\end{aligned}
$$

$$
\begin{aligned}
I &= \int_0^1 dx(1-x^2)\,\frac{\cos\frac{\pi}{2}x}{1-x^2} \\
&= \int d\rho\,\frac{\cos\frac{\pi}{2}x}{1-x^2}[x(\rho)] \\
&= 0.637 \pm 0.032/\sqrt{N}
\end{aligned}
$$

Improve convergence: Stratified sampling

- Want to minimise number of function calls.

  (They are potentially CPU-expensive.)

  $\Longrightarrow$ Need to improve convergence of MC integration.

- Basic idea here: Decompose integral in M sub-integrals

$$\langle I(f)\rangle = \sum_{j=1}^{M}\langle I_j(f)\rangle, \quad \langle E(f)\rangle^2 = \sum_{j=1}^{M}\langle E_j(f)\rangle^2$$

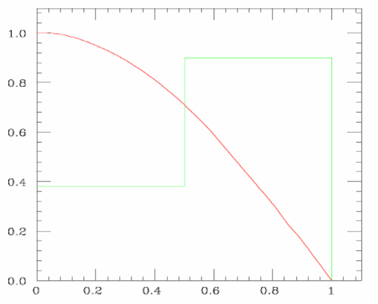
- Then: Overall variance smallest, if "equally distributed".

  ($\Longrightarrow$ Sample, where the fluctuations are.)

- Algorithm:
  - Divide interval in bins (variable bin-size or weight);
  - adjust such that variance identical in all bins.

Stratified sampling: Example results

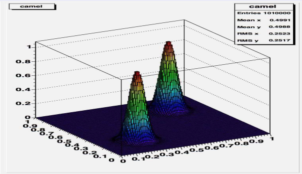
- Consider $f(x) = \cos\frac{\pi x}{2}$ and $g(x) = 1 - x^2$:



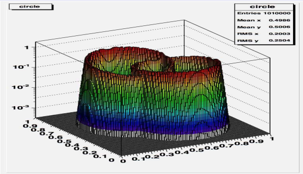
$$\langle I \rangle = 0.637 \pm 0.147/\sqrt{N}$$

Example for stratified sampling: VEGAS

- Good for Vegas:
  Singularity "parallel" to integration axes



- Bad for Vegas:
  Singularity forms ridge along integration axes



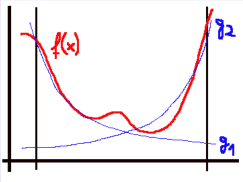
Improve convergence: Multichannel sampling

- Want to minimise number of function calls.

  (They are potentially CPU-expensive.)

  $\Longrightarrow$ Need to improve convergence of MC integration.

- Basic idea: Best of both worlds: Hybrid between importance and stratified sampling.



- Have "bins" – weight α_i – of "eigenfunctions" – $g_i(x)$:
  $\Longrightarrow \quad g(\vec{x}) = \sum_{i=1}^{N} \alpha_i g_i(\vec{x})$.
- In particle physics, this is the method of choice for parton level event generation!

Basic simulation paradigm

An example from thermodynamics

- Consider two-dimensional Ising model:

$$\mathcal{H} = -J\sum_{\langle ij\rangle} s_i s_j$$

(Spins fixed on 2-D lattice with nearest neighbour interactions.)

- Traditional model to understand (spontaneous) magnetisation & phase transitions.
- To evaluate an observable $\mathcal{O}$, sum over all micro states $\phi_{\{i\}}$, given by the individual spins. (Similar to path integral in QFT.)

$$\langle\mathcal{O}\rangle = \int\mathcal{D}\phi_{\{i\}}\,\mathrm{Tr}\left\{\mathcal{O}(\phi_{\{i\}})\,\exp\left[-\frac{\mathcal{H}(\phi_{\{i\}})}{k_B T}\right]\right\}$$

- Typical problem in such calculations (integrations!): Phase space too large $\Longrightarrow$ need to sample.

Metropolis-Algorithm

- Metropolis algorithm simulates the canonical ensemble, summing/integrating over micro-states with MC method.
- Necessary ingredient: Interactions among spins in probabilistic language (will come back to us.)
- Algorithm will look like: Go over the spins, check whether they flip (compare $\mathcal{P}_{\rm flip}$ with random number), repeat to equilibrate.
- To calculate $\mathcal{P}_{\rm flip}$: Use energy of the two micro-states (before and after flip) and Boltzmann factors.
- While running, evaluate observables directly and take thermal average (average over many steps).

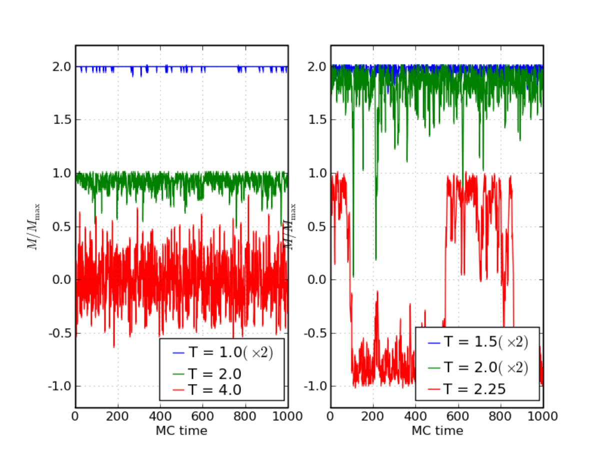
Why Metropolis is correct: Detailed balance

- Consider one spin flip, connecting micro-states 1 and 2.
- Rate of transitions given by the transition probabilities $\mathcal{W}$
- If $E_1 > E_2$ then $\mathcal{W}_{1\to 2} = 1$ and $\mathcal{W}_{2\to 1} = \exp\left(-\frac{E_1-E_2}{k_B T}\right)$
- In thermal equilibrium, both transitions equally often:
  $$\mathcal{P}_2\mathcal{W}_{2\to 1} = \mathcal{P}_1\mathcal{W}_{1\to 2}$$
  This takes into account that the respective states are occupied according to their Boltzmann factors.

  $(\mathcal{P}_i \sim \exp(-E_i/k_B T))$
- In principle, all systems in thermal equilibrium can be studied with Metropolis - just need to write transition probabilities in accordance with detailed balance, as above $\Longrightarrow$ general simulation strategy in thermodynamics.

Some example results

- Fix temperature, use a 10×10 lattice



Summary of lecture 1

- Discussed some basic numerical techniques.
- Introduced Monte Carlo integration as the method of choice for high-dimensional integration space (like phase space in multi-particle production).
- Introduced some standard improvement strategies to the convergence of Monte Carlo integration.
- Discussed connections between simulations and Monte Carlo integration with the example of the Ising model.