CTEQ - Introduction to Monte Carlo

Lecture 1
Overview and Monte Carlo Methods

Philip Ilten

Who am I?

1. faculty at the University of Cincinnati
2. experimental physicist on LHCb
   - searches for dark sectors (including CODEX-b)
   - precision electroweak physics
   - jet substructure measurements
3. heavily involved with PYTHIA 8 ("codemaster")
   - sophisticated τ decays with spin correlations
   - quarkonia production
   - non-perturbative models (coalescence, hadronic rescattering)

Tutorials

- tutorials via Docker containers,
  https://gitlab.com/cteq-tutorials/2022
- containers can be large (order of GB), please download early

2022 CTEQ School Schedule										
6 July 2022	07 Jul 2022	08 Jul 2022	09 Jul 2022	10 Jul 2022	11 Jul 2022	12 Jul 2022	13 Jul 2022	14 Jul 2022	15 Jul 2022	16 Jul 2022
Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Arrive	Day 1	Day 2	Day 3	Day 4	Free Day	Day 6	Day 7	Day 8	Day 9	Depart
8:00 - 8:45	Breakfast	Breakfast	Breakfast	Breakfast	No Breakfast or dinner	Breakfast	Breakfast	Breakfast	Breakfast	NO MEALS
9:00 - 10:00	Intro 1 Nadolsky	Intro 3 Nadolsky	Jets 1 Sterman	Jets 2 Sterman		PDF 1 Lin	PDF 2 Yuan	EW & Higgs 1 Mellado	EFT 2 Dawson	
10:00 - 10:30	Coffee	Coffee	Coffee	Coffee		Coffee	Coffee	Coffee	Coffee	
10:30 - 11:30	MC Intro 1 Ilten	MC Intro 3 Ilten	Higgs 1 Mistlberger	Higgs 2 Mistlberger		Vec Boson 1 Boughezal	Vec Boson 2 Boughezal	QCD & Top 1 Bruscino	QCD & Top 2 Bruscino	
11:45 - 13:00	Lunch	Lunch	Lunch	Lunch	Lunch	Lunch	Lunch	Lunch	Lunch	
13:00 - 14:00	Intro 2 Nadolsky	Tut. Lec 1 Gellersen	Intro 4 Nadolsky	Tut. Lec 2 Gellersen		Machine Learning 1 Isaacson	Machine Learning 2 Isaacson	Neutrino 1 Hobbs	Neutrino 2 Hobbs	
14:00-14:30	Coffee	Coffee	Coffee	Coffee		Coffee	Coffee	Coffee	Coffee	
14:30-15:30	MC intro 2 Ilten	MC Tutorial Gellersen	DIS 2 Reimer	MC Tutorial Gellersen		ML Tutorial Isaacson	ML Tutorial Isaacson	EFT 1 Dawson	EW & Higgs 2 Mellado	
15:30+	DIS 1 Reimer									
18:00 - 19:00	Dinner	Dinner	Dinner	Dinner		Dinner	Dinner	Dinner	Dinner	
19:00 - 21:00	Recitation	Recitation	Recitation	Recitation		Recitation	Recitation	Recitation	Recitation	
21:00 - 22:00	NightCap	NightCap	NightCap	NightCap		NightCap	NightCap	NightCap	NightCap	

Resources

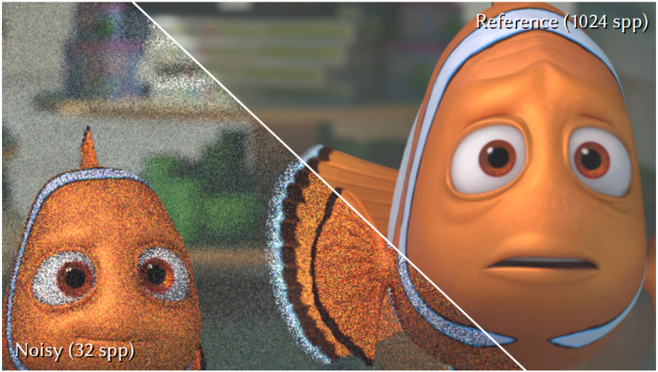
- these lectures are heavily based on those by Torbjörn Sjöstrand http://home.thep.lu.se/~torbjorn/welcomeaux/talks.html
- great lectures from previous CTEQ summer schools https://www.physics.smu.edu/scalise/cteq/#Summer
- QCD and Collider Physics by Ellis, Stirling, and Webber
- Pythia 6 Physics and Manual by Sjöstrand, Mrenna, and Skands
- General-purpose event generators for LHC physics by Buckley, *et al.*
- Introduction to parton-shower event generators by Höche

Overview

- **lecture 1 - introduction and Monte Carlo techniques**
- lecture 2 - matrix elements and parton showers
- lecture 3 - multi-parton interactions, hadronization, and non-perturbative effects

What is Monte Carlo?

using random sampling to calculate numerical results for problems that may or may not be deterministic



Why Monte Carlo Event Generators?

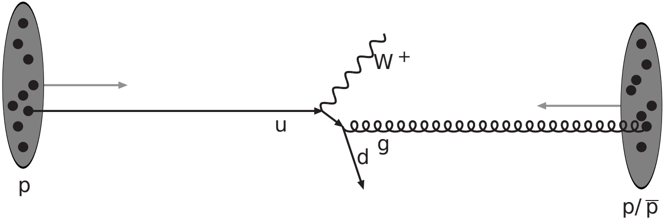
1. connect perturbative and non-perturbative regimes
2. provide complete events with final state particles
3. robustly perform high-dimension integrals

Generator Overview

- beams - proton parton density functions

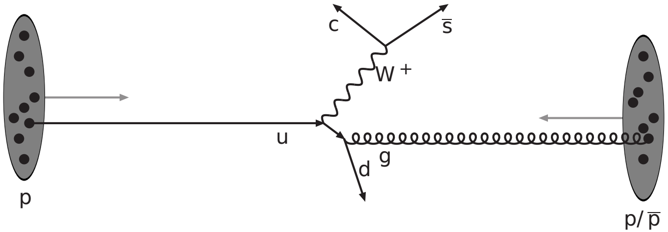
Generator Overview

- hard process - calculate with matrix element(s)



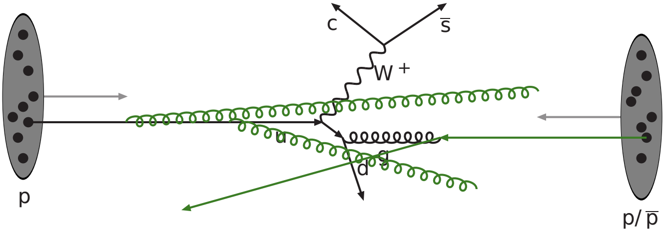
Generator Overview

- resonance decays - also calculate with matrix element(s)



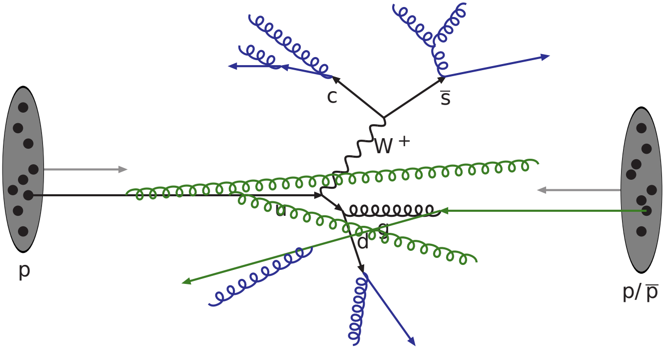
Generator Overview

- initial state radiation - spacelike parton shower



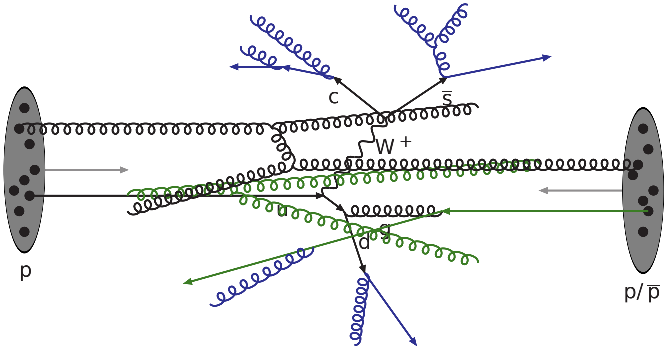
Generator Overview

- final state radiation - timelike parton shower



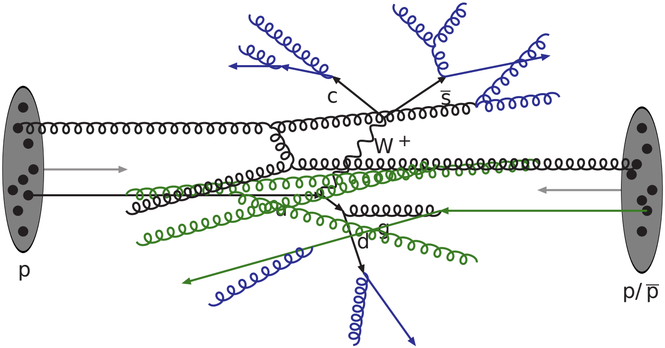
Generator Overview

- multi-parton interactions - calculate with matrix element(s)



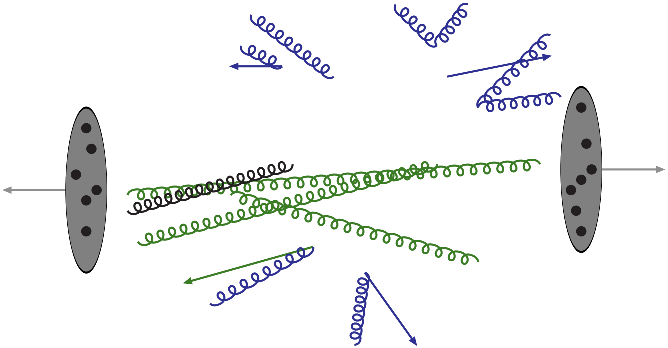
Generator Overview

- MPI radiation - additional ISR and FSR on each MPI



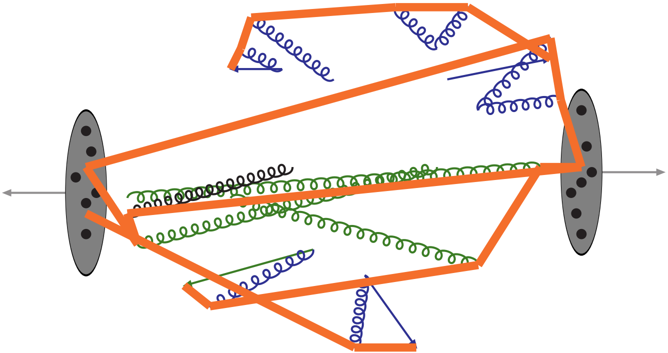
Generator Overview

- final partons and beam remnants



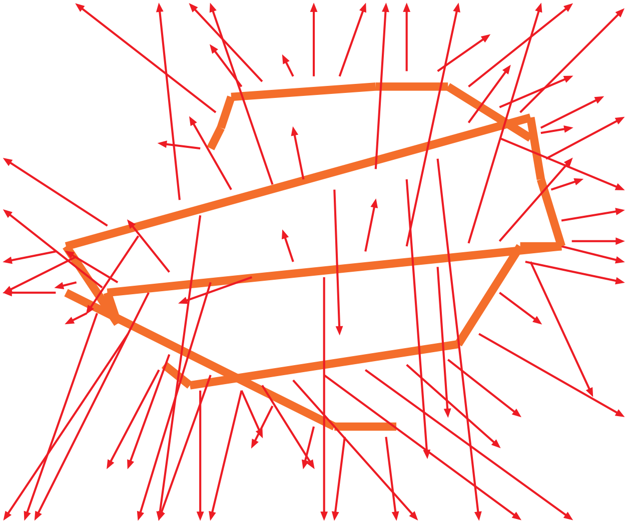
Generator Overview

- color connections - use a phenomenological model



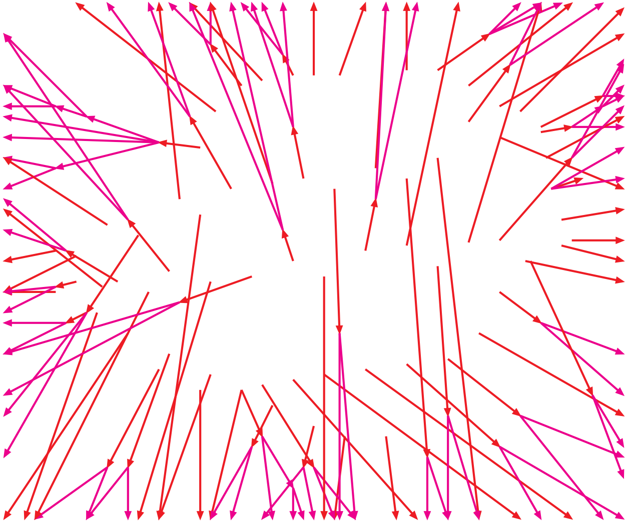
Generator Overview

- hadronize - use a phenomenological model

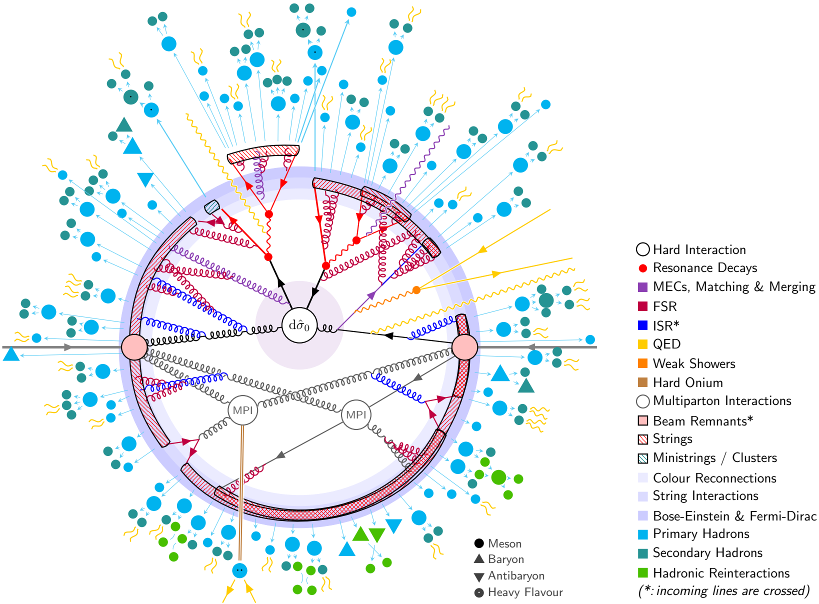


Generator Overview

- decays - calculate with matrix elements when possible



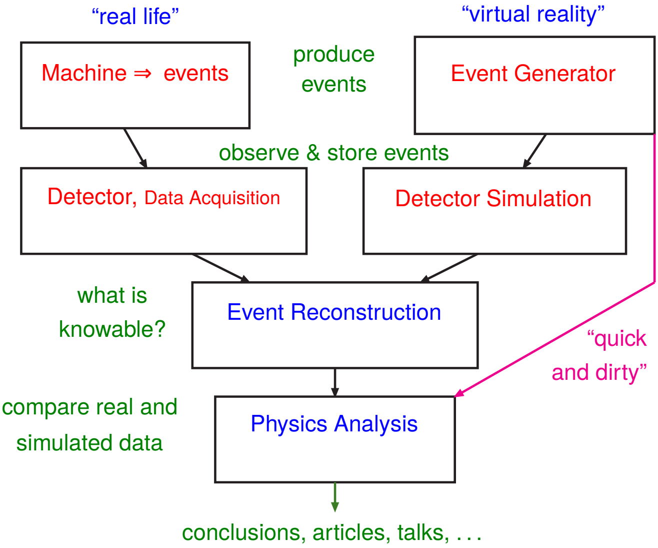
Another Overview



Generator Specialization

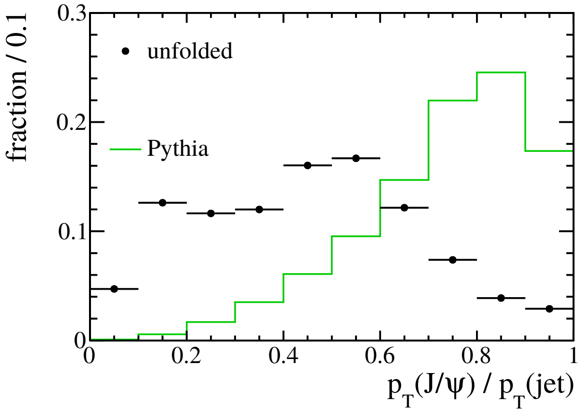
	General-Purpose	Specialized
Hard Processes		a lot
Resonance Decays	**HERWIG**	HDECAY, . . .
Parton Showers	**PYTHIA**	Ariadne/LDC, VINCIA, . . .
Underlying Event	**SHERPA**	PHOJET/DPMJET
Hadronization	......	MLHad
Ordinary Decays		TAUOLA, EvtGen

Not Just a Generator



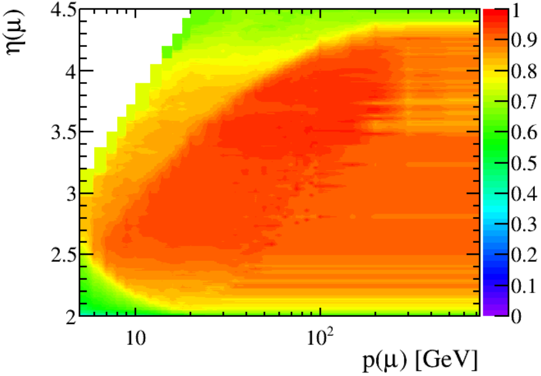
Detector Example

- select events with $J/\psi \to \mu\mu$ produced in jets (arXiv:1701.05116)
- measure $z \equiv p_\mathrm{T}(J/\psi)/p_\mathrm{T}(\mathrm{jet})$

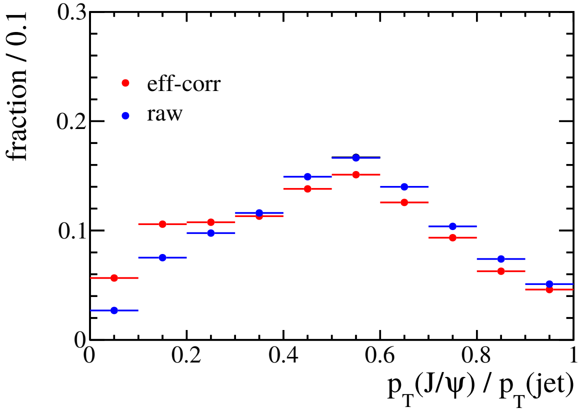


Detector Efficiency

- cannot reconstruct μ with 100% efficiency



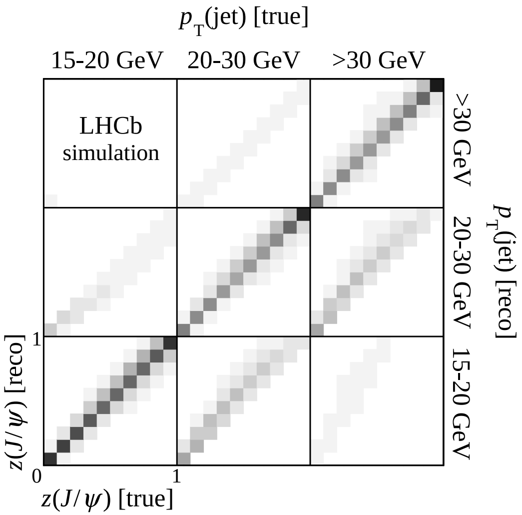
Detector Efficiency



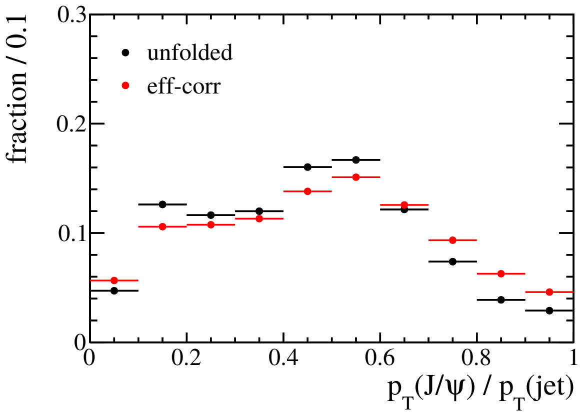
Detector Resolution

- cannot measure p_{T}(jet) perfectly

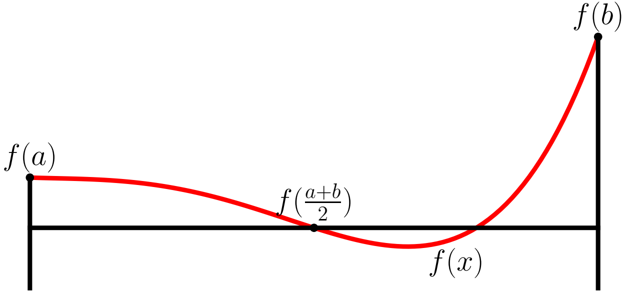
$$\vec{x}_{\mathrm{obs}} = \mathbf{R}\vec{x}_{\mathrm{org}} \Rightarrow \vec{x}_{\mathrm{org}} = \mathbf{R}^{-1}\vec{x}_{\mathrm{obs}}$$



Detector Resolution

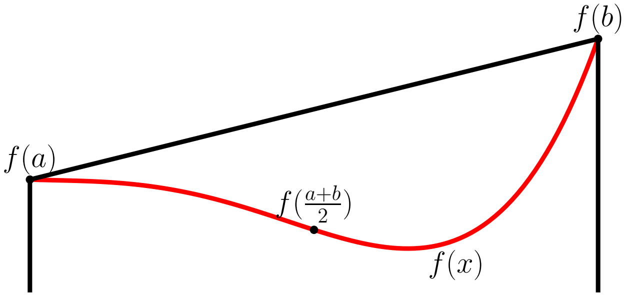


Quadrature Methods - Midpoint



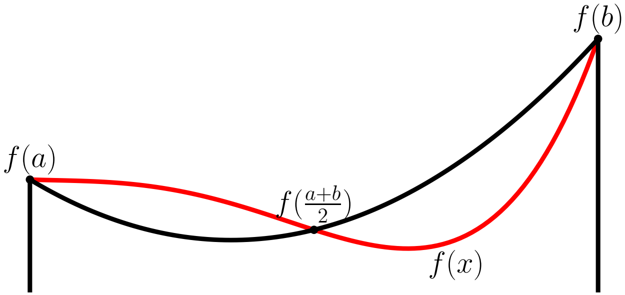
$$\int_a^b \mathrm{d}x\, f(x) \approx (b-a) f\left(\frac{a+b}{2}\right)$$

Quadrature Methods - Trapezoid



$$\int_a^b \mathrm{d}x\, f(x) \approx \frac{b-a}{2}\Big(f(a) + f(b)\Big)$$

Quadrature Methods - Simpson's

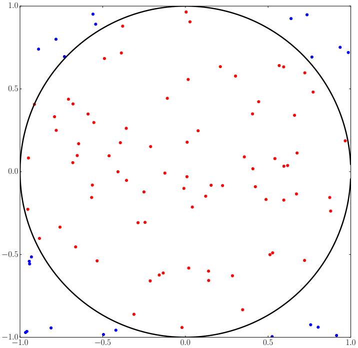


$$\int_a^b \mathrm{d}x\, f(x) \approx \frac{b-a}{6}\Big(f(a) + 4f\left(\tfrac{a+b}{2}\right) + f(b)\Big)$$

Monte Carlo Integration

$$\int_a^b \mathrm{d}x\, f(x) \approx \langle f(x) \rangle (b-a) \Rightarrow \int_V \mathrm{d}\vec{x}\, f(\vec{x}) \approx \langle f(\vec{x}) \rangle V$$

```
# Import the random number generator →
      library.
import random
# Set the random seed for →
      reproducibility.
random.seed(1)
rng = random.uniform
# Sample n points and find the sum.
s, n = 0, 1000000
for i in range(n):
    # Uniformly pick an x and y and →
          check point.
    x, y = rng(−1, 1), rng(−1, 1)
    s += 1. if x**2 + y**2 < 1 else 0.
# Print the integral (average times →
      integration volume).
print((s/n)*(2*2))

# Everything in one line!
print(sum([(rng(−1, 1)**2 + rng(−1, 1)→
      **2) < 1 for i in range(n)])→
      /(0.25*n))
```



Speeding Things Up

- adaptive quadrature - subdivide space until necessary accuracy is reached
- stratified sampling - same as above, but for MC integration

$$\int_V \mathrm{d}\vec{x}\, f(\vec{x}) = \sum \int_{V_i} \mathrm{d}\vec{x} \approx \sum_i \langle f(\vec{x}) \rangle V_i$$

- importance sampling - sample a non-uniform distribution to minimize variance

$$\int_V \mathrm{d}\vec{x}\, f(\vec{x}) \approx \langle f(\vec{x}) \rangle V = \frac{V}{N} \sum_i \frac{f(\vec{x}_i)}{p(\vec{x}_i)}$$

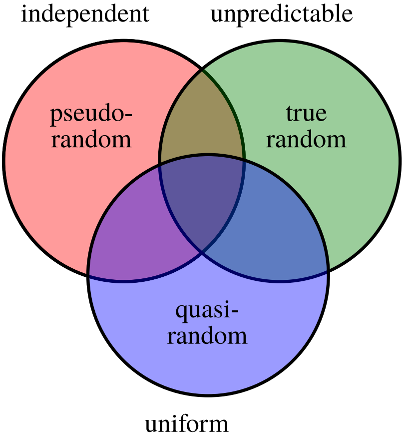
Comparison

- assume baseline algorithms (no adaptive, stratified, *etc.*)
- integral with d dimensions, and sampling n times

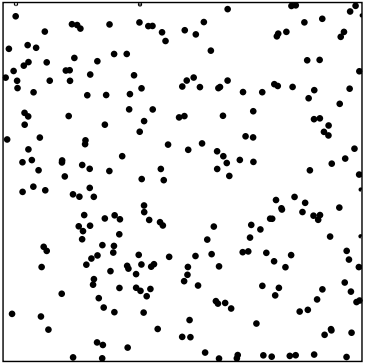
method	convergence
trapezoid	$1/n^{2/d}$
Simpson's	$1/n^{4/d}$
Monte Carlo	$1/\sqrt{n}$

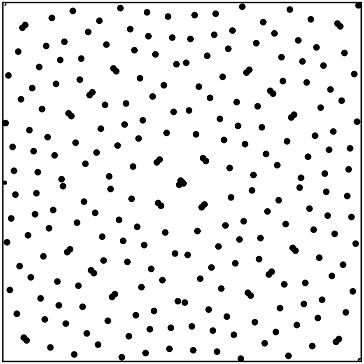
Randomness

- *true random* - what we see in nature
- *pseudorandom* - approximates *true random* but deterministic
- *quasirandom* - like *pseudorandom* but more uniform



Pseudo vs Quasi





Metrics

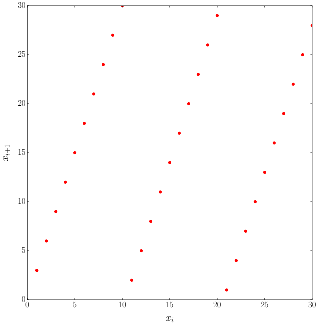
1. sequence is bounded
2. cannot easily determine pattern
3. moments approach expectation ($\mu = 1/2$, $\sigma = 1/12$, $\dots$)
4. divided in bins, each bin is Poissonian

- reproducible sequence
- fast to calculate
- long periodicity
- theoretically validated
- works ...

Spectral Test

$$x_i = (ax_0 + b) \mod m$$

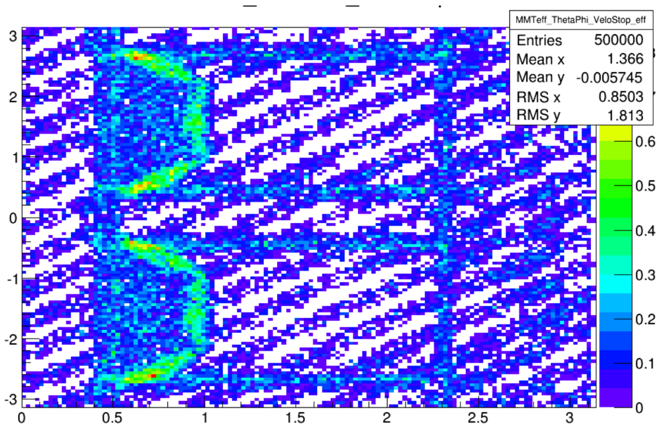
```python
# Define the random number function.
def rng(x0, a, b, m):
    return (a*x0 + b)%m

# Define the parameters used in minstd_rnd for C++11.
x0, a, b, m = 1, 48271, 0, 2**31 - 1

# Generate some random numbers.
for i in range(100):
    x0 = rng(x0, a, b, m)
    print(x0/float(m))
```



- linear congruential RNGs fail the spectral test
- known as the Marsaglia effect

Spectral Test in Practice

- used RANLUX in LHCb/Moedal simulation code, issues emerged
- commonly use Mersenne twister (use Mersenne primes, bit shift, and bit mask)



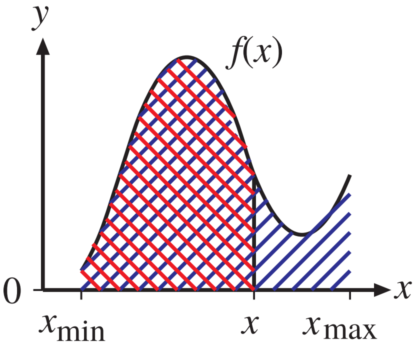
Sampling a Distribution

- we usually don't just want an integral
- we want to sample points from the distribution and an integral
- an n-dimensional distribution typically requires $n+1$ random numbers per point

- $f(x)$ - function to sample
- $F(x) = \int \mathrm{d}x\, f(x)$ - primitive, integral of function to sample
- $F^{-1}(x)$ - inverse of primitive

Analytic Sampling

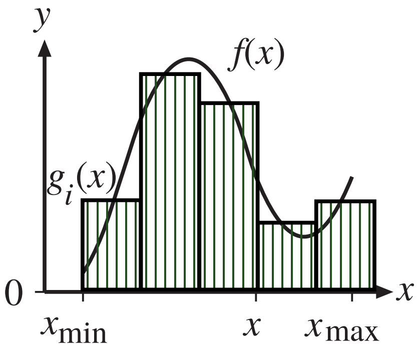
- sample from $f(x)$ with uniform sampling of bounded x

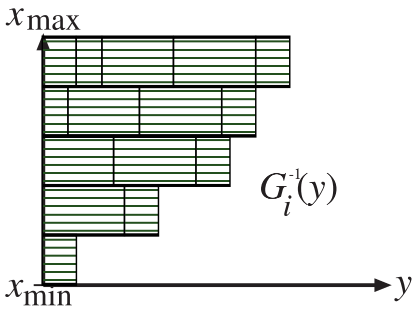


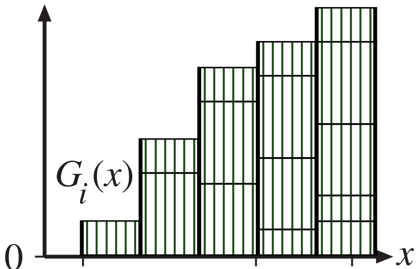
$$\begin{aligned}
\int_{x_{\min}}^{x} \mathrm{d}x'\, f(x') &= R \int_{x_{\min}}^{x_{\max}} \mathrm{d}x'\, f(x') \\
F(x) - F(x_{\min}) &= R(F(x_{\max}) - F(x_{\min})) \\
&= RA \\
x &= F^{-1}(F(x_{\min}) + RA)
\end{aligned}$$

- $f(x) = 2x,\ 0 < x < 1 \Rightarrow x = \sqrt{R}$
- $f(x) = e^{-x},\ 0 < x \Rightarrow x = -\ln R$

Binned Sampling



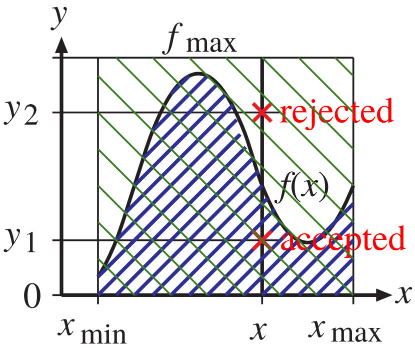




1. sample $0 < R < G_i^{-1}(x_{\max})$
2. find corresponding bin i
3. uniformly sample from bin x-range

Accept or Reject Sampling

- sample from $f(x)$ with uniform sampling of bounded x

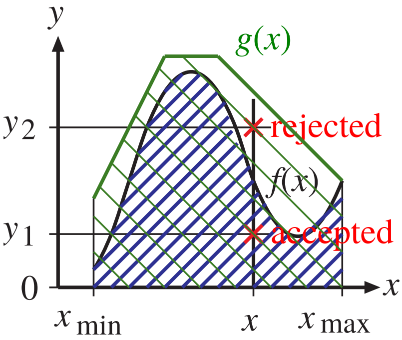


1. $x = x_{\text{min}} + R_1(x_{\text{max}} - x_{\text{min}})$
2. $y = R_2 f_{\text{max}}$
3. if $y > f(x)$ return to 1
   otherwise accept point

$$\int_{x_{\text{min}}}^{x_{\text{max}}} \mathrm{d}x\, f(x) \approx \frac{N_{\text{acc}}}{N_{\text{try}}} f_{\text{max}}(x_{\text{max}} - x_{\text{min}})$$

Importance Sampling

- same as accept or reject, but choose efficient $g(x)$
- $g(x) \geq f(x)$ and be easily sampled

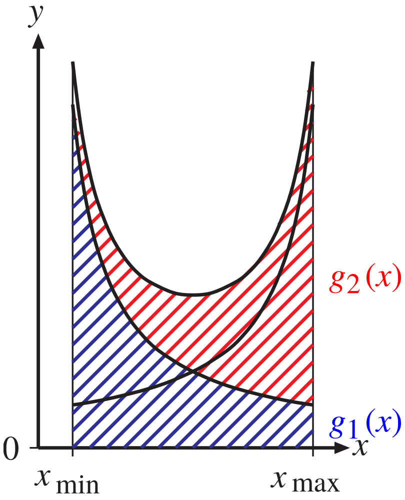


1. x from $g(x)$
2. $y = Rg(x)$
3. if $y > f(x)$ return to 1
   otherwise accept point

$$\int_{x_{\min}}^{x_{\max}} \mathrm{d}x\, f(x) \approx \frac{N_{\mathrm{acc}}}{N_{\mathrm{try}}} \int_{x_{\min}}^{x_{\max}} \mathrm{d}x\, g(x)$$

Multichannel Sampling

- like importance sampling but construct $g(x) = \sum_i g_i(x)$



1. select $g_i(x)$ with relative probability $G_i(x_{\max}) - G_i(x_{\min})$
2. select x from $g_i(x)$
3. $y = Rg(x)$
4. if $y > f(x)$ return to 1 otherwise accept point

Sampling in Time

- considered only sampling in space, no memory
- consider the decay of a particle which is time dependent

- given a particle at time t, define $f(t)$ as probability of decay
- normalize number of particles $N(t)$ with $N(0) = 1$
- $\Rightarrow N(t)$ is probability particle has not decayed by t
- $P(t)$ is probability of decay at time t

$$P(t) = \frac{-\mathrm{d}N(t)}{\mathrm{d}t} = f(t)N(t)$$
$$\Rightarrow N(t) = \exp\left(-\int_0^t \mathrm{d}t'\, f(t')\right) = R$$
$$\Rightarrow t = F^{-1}(F(0) - \ln R)$$

- taking $f(t) = \lambda$ recovers particle decay

The Veto Algorithm

- what if we can't sample $f(t)$ and need importance sampling?

$$P(t) = f(t) \exp\left(-\int_0^t \mathrm{d}t'\, g(t')\right)$$

- the exponentiated factor is wrong!

1. start with $i = 0$ and $t = 0$
2. increment i
3. $t_i = G^{-1}(G(t_{i-1}) - \ln R)$
4. $y = Rg(t_i)$
5. if $y > f(t_i)$ return to 2 otherwise accept point

Winner Takes All

- what if we have have multiple decay channels?

1. set $f(t) = f_1(t) + f_2(t)$
2. sample t using $f(t)$
3. select channel using probabilities $f_1(t)$ and $f_2(t)$

- winner-takes-all method

1. sample t using $f_1(t)$
2. sample t' using $f_2(t')$
3. select channel with the smaller t

Summary

1. connect perturbative and non-perturbative regimes
2. provide complete events with final state particles
3. robustly perform high-dimension integrals

- MC integration and sampling are not the same
- multichannel and veto sampling are commonly used
- use a good random number generator