Monte Carlo Methods in High Energy Physics IV

Peter Uwer
Memory consumption Vegas

Full decomposition of the integration volume $[0,1]^d$ into $n$ sub-intervals for each variable would mean:

$$n^d \times 8\text{Byte} = \frac{10^8 \times 8}{1024^3} \text{GByte} = 74\text{GByte}$$

one double per cell \hspace{1cm} n=10, d=10

→ this is not done for obvious reasons

(would correspond to multi-dimensional array: $w[n][n][n] \ldots [n]$)

Due to factorization assumption:

$$p(x_1, x_2, \ldots, x_d) \rightarrow p_1(x_1)p_2(x_2) \ldots p_n(x_n)$$

variables are binned independent from each other: $w[n][d]$

$$n \times d \times 8\text{Byte} = 800\text{Byte}$$

n=10, d=10
Memory consumption vegas

Code from the stone age…

3x4KByte (d=10), would even fit into first level cache of modern CPU’s
Contents

1 Introduction
   1.1 Monte Carlo methods ................................................
      1.1.1 Simulation of LHC physics ..............................
      1.1.2 The Ising modell ..........................................
      1.1.3 Buffon’s needle ............................................
   1.2 Probability and statistics ........................................
      1.2.1 Basic facts ............................................... 
      1.2.2 Specific probability distribution functions ...........
      1.2.3 The central limit theorem ..............................

2 Generation of random numbers
   2.1 Generation of uniform distributions ..........................
      2.1.1 How to calculate random numbers ....................
      2.1.2 Testing random numbers ..............................
   2.2 Generation of non-uniform distributions ....................
      2.2.1 General algorithms ............................... 
      2.2.2 Specific distributions ..............................

3 Monte Carlo integration
   3.1 Introduction ................................................
   3.2 Variance Reduction ........................................
   3.3 A concrete example: Vegas by Peter Lepage ..............
   3.4 A note on convergence of Monte Carlo methods — and how to compare results ........................................

4 Phase integration
   4.1 Flat phase space with RAMBO ..............................
   4.2 Sequential splitting à la Byckling and Kajantie .......... 
   4.3 Multi-channel methods ....................................
   4.4 From phase-space integration to a full Monte Carlo  .......
The problem:

Central value
Uncertainty
Numerical integration
Shift towards $\mu = 2m, m/2$

Cross sections

Top-quark pair production with an additional jet at the Tevatron

<table>
<thead>
<tr>
<th>$p_T^{cut}$ [GeV]</th>
<th>Cross section [pb]</th>
<th>Charge asymmetry [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LO</td>
<td>NLO</td>
</tr>
<tr>
<td>20</td>
<td>$1.583(2)^{+0.96}_{-0.55}$</td>
<td>$1.791(1)^{+0.16}_{-0.31}$</td>
</tr>
<tr>
<td>30</td>
<td>$0.984(1)^{+0.60}_{-0.34}$</td>
<td>$1.1194(8)^{+0.11}_{-0.20}$</td>
</tr>
<tr>
<td>40</td>
<td>$0.6632(8)^{+0.41}_{-0.23}$</td>
<td>$0.7504(5)^{+0.072}_{-0.14}$</td>
</tr>
<tr>
<td>50</td>
<td>$0.4670(6)^{+0.29}_{-0.17}$</td>
<td>$0.5244(4)^{+0.049}_{-0.096}$</td>
</tr>
</tbody>
</table>

- minimum $p_T$ of additional jet
- central value
- uncertainty num. intgration
- shift towards $\mu = 2m, m/2$

$\Rightarrow$ total cross sections including cuts and observables

$\Rightarrow$ total cross sections are difficult to measure not necessarily the best to test theory
The problem

**Distributions**

\[ \left( \frac{d\sigma}{dp_{T,\text{jet}}} \right) \left( \frac{\text{fb}}{\text{GeV}} \right) \]

\( p_T^{\min} = 20 \text{ GeV} \)

\( \sqrt{s} = 1.96 \text{ TeV} \)

\( \text{pp} \rightarrow \bar{t}t + \text{jet} + X \)

Tevatron

\( p_T^{\min} = 50 \text{ GeV} \)

\( \sqrt{s} = 14 \text{ TeV} \)

LHC

\( K = \text{NLO}/\text{LO} \)

\( \Rightarrow \text{more sensitive probe of theory} \)
The problem

We want to calculate

\[ \sigma_{\text{cut}} = \frac{1}{2s} \int \prod_{i=1}^{n} \frac{d^3 p_i}{(2\pi)^3 2E_i} \delta(k_1 - k_2 - \sum_{i=1}^{n} p_i) |T_{fi}|^2 \theta_{\text{cut}}(p_1, \ldots, p_n) \]

number of independent variables \(3n - 4\)

without cuts simple for \(n=2,3\)

For \(n=3\) with cuts i.e. Durham-Jetalgorithm, already non-trivial

\[ \Theta_{\text{Durham}}(p_1, \ldots, p_n) = \prod_{i<j} \Theta \left( \frac{\max(E_i^2, E_j^2)}{s} (1 - \cos \theta_{ij}) - y_{\text{cut}} \right) \]
The problem

In addition we also want to calculate distributions

\[ \frac{d\sigma}{dO} = \frac{1}{2s} \int \prod_{i=1}^{n} \frac{d^3p_i}{(2\pi)^3 2E_i} \delta(k_1 - k_2 - \sum_{i=1}^{n} p_i) |T_{fi}|^2 \theta_{cut}(p_1, \ldots, p_n) \delta(O - O(p_1, \ldots, p_n)) \]

Hopeless to solve this integral analytically apart from rather simple cases

→ use Monte Carlo integration

→ we need to identify the integration variables

additional argument for MC: can easily deal with non-continous or otherwise strange integrands
Hadron colliders

If hadronic collisions are studied we have in addition two integrations over the momentum fractions of the incoming partons:

\[
\sigma_{\text{cut}}^{\text{Had.}} = \int dy_1 \int dy_2 F(y_1, \mu_f) F(y_2, \mu_f) \frac{1}{2s_{\text{had}}y_1y_2} \int \prod_{i=1}^{n} \frac{d^3p_i}{(2\pi)^3 2E_i} \delta(y_1k_1 - y_2k_2 - \sum_{i=1}^{n} p_i) |T_{fi}(y_1k_1, y_2k_2, p_1 \ldots)|^2 \theta_{\text{cut}}(p_1, \ldots, p_n)
\]

→ just two additional integration variables, MC integration does not care!
Integration variables: simple cases

2-particle final state:

\( \phi, \theta \)

Azimuthal and polar angle of one particle, everything else is fixed by momentum conservation

In most cases the matrix elements do not depend on \( \phi \), \( \rightarrow \) integrate out, only one integration variable

\( \rightarrow \) integration boundaries straight forward

3-particle final state

\( \phi, \theta, x_1, x_2 \)

Two angles to describe the orientation of the event plane, \( x_i = \frac{2E_i}{\sqrt{s}} \) energies of two outgoing particles

\( \rightarrow \) integration boundaries not so straight forward anymore
Integration boundaries

for the massive case complicated boundaries
Integration variables: simple cases

4 parton final space:

\[ \phi, \theta, t_{12}, t_{13}, t_{14}, t_{23}, t_{34} \]

\[ t_{ij} = 2 p_i \cdot p_j \] → need phase space boundaries, and jacobian

→ rather involved phase space boundaries,

→ search for general approach based on MC methods
Phase space integration

If we want to use a simple Monte Carlo integrator we need:

\[ [0, 1]^{3n-4} \rightarrow (p_1, p_2, \ldots, p_n) \]

satisfying “on-shell”-condition and momentum conservation:

\[ p_i^2 = m_i^2, \quad k_1 + k_2 = \sum_i p_i \]

In addition we need the jacobian/weight of the transformation:

\[ \prod_i d^3 p_i = \frac{\partial(p_1, \ldots, p_n)}{\partial(x_1, \ldots, x_{3n-4})} dx_1 \ldots dx_{3n-4} \]
Democratic approach to phase space

RAMBO = RA(NDOM) M(OMENTA) B(EAUTIFULLY) O(RGANIZED)  

[Ellis (SD), Kleiss, Stirling]

Scetch of the derivation:

Consider:  
\[ R_n = \int \prod_{i=1}^{n} d^4 q_i \delta(q_i^2) f(q_i^0) \Theta(q_i^0) \quad \text{with} \quad f(x) = \exp(-x) \]

Replace \( q_i \) by (use delta-functions!):

\[ p_i^0 = x(\gamma q_i^0 + b q_i), \quad p_i = x(q_i + b q_i^0 + a(b q_i)b) \]

\[ b = -Q/M, \quad x = \sqrt{s}/M, \quad \gamma = Q^0/M = \sqrt{1 + b^2}, \]

\[ a = 1/(1 + \gamma), \quad Q^\mu = \sum_{i=1}^{n} q_i^\mu, M = \sqrt{Q^2} \]

\[ \rightarrow \text{Integration over } b \text{ and } x \text{ can done} \]
Democratic approach to phase space

The remaining integral in $p$ gives the ordinary phase space measure:

$$R = c \times \int \delta(P - \sum_i p_i) \prod_i d^4p_i \delta(p_i^2) \Theta(p_i^0)$$

Algorithm:

1. Generate the $q_i$

2. Calculate the $p_i$ from the $q_i$
   
   $q_i^0$ is distributed according to $x \exp(-x)$
   
   $q_i^0 = -\ln(u_1u_2), c_i = 2u_3 - 1, \phi = 2\pi u_4$
   
   can also use this to map $[0, 1]^{4n} \rightarrow (p_1, \ldots, p_n)$

works with minor modification also for massive momenta
C Modified version of Rambo, instead of creating the
C random numbers in Rambo, they are passed through an
C additional variable RN(4,100).
C
SUBROUTINE phpoint(N,ET,XM,RN,P,WT)
C
RAMBO
C
RA(NDOM) M(ONENTA) B(EAUTIFULLY) O(RGANIZED)
C
A DEMOCRATIC MULTI-PARTICLE PHASE SPACE GENERATOR
C AUTHORS: S.D. ELLIS, R. KLEISS, W.J. STIRLING
C THIS IS VERSION 1.0 - WRITTEN BY R. KLEISS
C
N = NUMBER OF PARTICLES (>1, IN THIS VERSION <101)
C ET = TOTAL CENTRE-OF-MASS ENERGY
C XM = PARTICLE MASSES ( DIM=100 )
C P = PARTICLE MOMENTA ( DIM=(4,100) )
C WT = WEIGHT OF THE EVENT
C
** IMPLICIT REAL*8(A-H,O-Z)
*** implicit none

double precision xm,rn,p,q,z,r,b,p2,v2,m2,v2,m2

double precision acc,F0,x,bq,wt,wtmax,accu,g0,g2,wt2,wt3,wtm
double precision twopi,C,Rmas,G,A,F,S,p2log,et
double precision iwarn,ibegin,iter,imax,k,i,nm,n

dimension xm(100),rn(4,100),p(4,100),q(4,100),z(100),r(4),
   b(3),p2(100),xm2(100),e(100),v(100),iwarn(5)
data acc/1.0-14/,'itmax/6/','ibegin/0/','iwarn/5*0/

C I added the following line otherwise the variables are not static, for ibegin the statement is not required (P.U.)
SAVE IBEGIN,TWOPIC,P2LOG,Z

C INITIALIZATION STEP: FACTORIALS FOR THE PHASE SPACE WEIGHT
C IF(IBEGIN.NE.0) GOTO 103
IBEGIN=1
TWOPIC=8.*D1AN(1.0D0)
Comments:

- Events have uniform weight in phase space
- Useful for testing purposes
- For real integration not that useful:
  - more integration variables than actually needed
  - Due to complicated mapping vegas unable to optimize
- Useful in constructing multi channel generators
Sequential splitting

Phase space can be factorized:

\[
R_n(M_n^2) = \frac{1}{2M_n} \int_{\mu_{n-1}}^{M_n - m_n} dM_{n-1} d\Omega_{n-1} \frac{1}{2} p_n \cdots \int_{\mu_2}^{M_3 - m_3} dM_2 d\Omega_2 \frac{1}{2} P_3 \int d\Omega_1 \frac{1}{2} P_2
\]

\[
M_n^2 = k_n^2, k_i = p_1 + \ldots + p_i, \mu_i = m_1 + \ldots + m_i
\]

\[
P_i = \sqrt{\lambda(M_i^2, M_{i-1}^2, m_i^2)} \quad \frac{1}{2M_i}
\]
Sequential splitting

The momenta are generated in the respective rest frames.

Generate angles in the individual restframes, generate the masses $M_i$

The momenta are generated in the respective rest frames

Apply boosts to all the momenta to transform them into the same (overall) rest frame (iterative procedure)

method gives mapping $[0, 1]^{3n-4} \rightarrow (p_1, \ldots, p_n)$

$$w \sim \frac{1}{2M_n} \prod_i \frac{1}{2} p_i$$

[Byckling, Kajantie]
Sequential splitting

Comments:

- Some freedom in ordering
- Can also be used for direct integration
- Seems to work better than Rambo when combined with Vegas
- Can be adopted to generate soft/collinear configurations
- Possible to combine different orderings

Note: There is no “one size fits all” general solution to phase integration

RAMBO and sequential splitting should be taken as a starting point, very useful to get a “first” program
In typical phase space integrals there are usually more problematic variables than integration variables

→ not possible to be good in all problematic variables!

→ multi-channel methods

Define different mappings optimized for specific configurations

Sample/integrate using a weighted sum over the individual mappings

$$\sum_i p_i f_i(\vec{x}, p_1, \ldots, p_n)$$

→ sampling by composition
Multi-channel

Taken to the extreme:

Generate one mapping for each Feynman integral

Combine all channels as it was done for the probability distributions (sampling by composition)

Individual channels can be constructed using sequential splitting, RAMBO
In the case of QCD tree-amplitudes where the pole structure is pretty well understood there exist dedicated algorithms

→ Sarge, an algorithm for generating QCD-antennas

[Hamre, Kleiss, Draggiotis]

Note:

In next-to-leading order calculations the situation is different:

We integrate \[ |T| - \sum_i \text{Dipoles}_i \] [see Kouhei Hasegawa’s talk]

→ behaviour of the combination very different compared to un-subtracted matrix elements

→ no general technique
Distributions

Monte Carlo integrator provides weight and configuration

Possible to calculate (discrete) distributions = histograms at the same time

\[ p_{ij} = \sqrt{(p_i + p_j)} \]

fill histogram with \( d\sigma \times w \) according to the value of \( O \)

\[ \Rightarrow \quad \frac{d\sigma}{dO} \]

Can also be understood as integrating a vector

modern MC integration packages are usually prepared for that, see i.e. Cuba by Thomas Hahn
Steps towards a full Monte Carlo

Goal: Want to have full simulation as close as possible to nature

i.e. want to have hadronic events which are distributed as in nature

→ we need so called un-weighted events in difference from weighted ones

in ideal simulation no difference between real and simulated events

→ optimal to test the experimental analysis

If affordable (CPU time!):

Create as many MC events as you expect to observe
From weighted to un-weighted events

For un-weighted events distribution should be according to underlying theory, i.e. matrix elements, parton distribution, ...

Events generated in MC integration are weighted:

\[ w \sim \frac{d(p_1, \ldots, p_n)}{d(x_1, \ldots, x_{3n-4})} \left| T(y_1 k_1, y_2 k_2, p_1, \ldots, p_n) \right|^2 F(y_1, \mu_f) F(y_2, \mu_f) \]

If the maximum weight \( w_m \) is known we can “un-weight” events:

1. For each event generate uniform random number \( r \) between 0 and \( w_m \)
2. If \( w(p1, \ldots) < r \) reject the event otherwise keep the event
3. Give any surviving event the weight 1

\[ \rightarrow \text{hit and miss algorithm} \]

(as far as efficiency is concerned only useful if processing takes much longer then generating)
Event generators

Very important tool in today's experimental analysis

→ everybody should have a rough idea what goes in there

what we might see at the LHC

how we understand it

Higgs event

[Image of event generators diagram]
Event generators – a closer look

[Gieseke]
Event generators – a closer look

Parton shower

[Gieseke]
Event generators – a closer look

Hadronisation

[Gieseke]
Event generators – a closer look

underlying event

[Gieseke]
Hadronic cross sections

\[ d\sigma = d\sigma_{\text{hard}} dP(\text{partons} \rightarrow \text{hadrons}) \]

\[ \int dP(\text{partons} \rightarrow \text{hadrons}) = 1 , \]

\[ dP(\text{partons} \rightarrow \text{hadrons}) = dP(\text{resonance decays}) \]

\[ \times dP(\text{parton shower}) \quad [\Gamma > Q_0] \]

\[ \times dP(\text{hadronisation}) \quad [\text{TeV} \rightarrow Q_0] \]

\[ \times dP(\text{hadronic decays}) \quad [\sim Q_0] \]

\[ \times dP(\text{hadronic decays}) \quad [O(\text{MeV})] \]

Complex simulation \(\rightarrow\) Herwig, Pythia


1 Introduction
1.1 Monte Carlo methods
   1.1.1 Simulation of LHC physics
   1.1.2 The Ising model
   1.1.3 Buffon's needle
1.2 Probability and statistics
   1.2.1 Basic facts
   1.2.2 Specific probability distribution functions
   1.2.3 The central limit theorem

2 Generation of random numbers
2.1 Generation of uniform distributions
   2.1.1 How to calculate random numbers
   2.1.2 Testing random numbers
2.2 Generation of non-uniform distributions
   2.2.1 General algorithms
   2.2.2 Specific distributions

3 Monte Carlo integration
3.1 Introduction
3.2 Variance Reduction
3.3 A concrete example: Vegas by Peter Lepage
3.4 A note on convergence of Monte Carlo methods — and how to compare results

4 Phase integration
4.1 Flat phase space with RAMBO
4.2 Sequential splitting à la Byckling and Kajantie
4.3 Multi-channel methods
4.4 From phase-space integration to a full Monte Carlo
The End