Monte Carlo Event Generators at the LHC

Stefan Höche
Centre for Particle Physics and Phenomenology
Université Catholique de Louvain

LHC-D Workshop 2006
QCD and Electroweak Physics
Karlsruhe, 9.3.2006
Outline

- Inner working of MC Event Generators
  (The perturbative part)
  - Hard matrix elements
  - Parton showers
  - Combination of ME & PS
- Selection of MC tools
  (Apologies in advance for all unlisted programs ... )
How do Event Generators work?

- Impose factorisation of perturbative/nonperturbative parts of an event allows decomposition into different stages. E.g. hadronic collision.

- Perturbative part
  Hard process
  (full matrix element
   fixed order in (running) coupling
   generates initial particle kinematics)
How do Event Generators work?

- Impose factorisation of perturbative/nonperturbative parts of an event allows decomposition into different stages, e.g. hadronic collision.

- Perturbative part
  - Hard process
  - Initial state parton shower
    (all orders (N)LL resummation mostly markovian approach)
How do Event Generators work?

- Impose factorisation of perturbative/nonperturbative parts of an event allows decomposition into different stages
  - e.g. hadronic collision

- Perturbative part
  - Hard process
  - Initial state parton shower
  - Final state parton shower
    - (all orders (N)LL resummation)
How do Event Generators work?

- Impose factorisation of perturbative/nonperturbative parts of an event allows decomposition into different stages
  e.g. hadronic collision

- **Perturbative part**
  - Hard process
  - Initial state parton shower
  - Final state parton shower

- **Nonperturbative part**
  - Parton Distributions (PDF)
How do Event Generators work?

- Impose factorisation of perturbative/nonperturbative parts of an event allows decomposition into different stages.

  e.g. hadronic collision

- **Perturbative part**
  - Hard process
  - Initial state parton shower
  - Final state parton shower

- **Nonperturbative part**
  - Parton Distributions (PDF)
  - Cluster/String formation
    (modelling of nonperturbative dynamics of parton system)
How do Event Generators work?

- Impose factorisation of perturbative/nonperturbative parts of an event allows decomposition into different stages
  - e.g. hadronic collision

- **Perturbative part**
  - Hard process
  - Initial state parton shower
  - Final state parton shower

- **Nonperturbative part**
  - Parton Distributions (PDF)
  - Cluster/String formation
  - Cluster decays
  - Hadron Decays
Simulating the Hard Process

General task: generate events (unweighted or weighted) according to the differential cross section

$$d\sigma = \frac{1}{F} d\Phi |M|^2$$

Two steps: calculate the hard matrix element $|M|^2$
sample the phase space $\Phi$

Problems: calculation of hard ME rather complex for large number of final state particles (factorial growth of Feynman diagrams)
Example: $W+5$jets: about 7000 diagrams

high-dimensional phase space ($3N-4$) with probably sharply peaked integrand (e.g. QCD multi-parton matrix elements) and cuts on kinematic variables
Calculating the Hard ME

Methods to evaluate the hard ME:

- Pre-calculated matrix elements
  - In general very fast evaluation
  - Limited set of processes, lacks generality
- Calculate total amplitude from Feynman rules
  (state of the art: helicity formalism (hep-ph/9403244))
  - Can handle (in principle) arbitrary processes
  - Limitations for large multiplicities due to factorial growth of diagrams
  - No Feynman rules needed at all
  (total amplitude directly from Lagrangian)
  - Very fast evaluation, large multiplicities
  - So far formulated for tree level only
General idea of MC integration: $I = \int_{\Omega} dx f(x) = \Omega \langle f \rangle \{ x_i | i = 1 \ldots n \}$

Problem: to be efficient, must sample such that each single weight is close to the average

But: main peak structure of integrand known (in principle) and given by Feynman diagrams

Employ multi-channel method (e.g. hep-ph/9405257)

$I = \int_{\Omega} dx g(x) w_g(x), \quad V = \int_{\Omega} dx g(x) w_g^2(x), \quad w_g(x) = \frac{f(x)}{g(x)}$

where $g(x) = \sum \alpha_i g_i(x)$ with $\sum \alpha_i = 1$

Single channels $g_i$ are constructed acc. to diagrams

$D_{iso}(23, 45) \otimes P_0(23) \otimes P_0(45) \otimes D_{iso}(2, 3) \otimes D_{iso}(4, 5)$

Channels can be improved further using adaptive algorithms like VEGAS (CLNS-80/447 (1980))
Calculating the Hard ME

Comparison of ME Generators in context of MC4LHC
(http://indico.cern.ch/categoryDisplay.py?categId=152)

<table>
<thead>
<tr>
<th>Number of jets</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPGEN</td>
<td>3904(6)</td>
<td>1013(2)</td>
<td>364(2)</td>
<td>136(1)</td>
<td>53.6(6)</td>
<td>21.6(2)</td>
<td>8.7(1)</td>
</tr>
<tr>
<td>AMEGIC++</td>
<td>3908(3)</td>
<td>1011(2)</td>
<td>362.3(9)</td>
<td>137.5(5)</td>
<td>54(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompHEP</td>
<td>3947.4(3)</td>
<td>1022.4(5)</td>
<td>364.4(4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GR@PPA</td>
<td>3905(5)</td>
<td>1013(1)</td>
<td>361.0(7)</td>
<td>133.8(3)</td>
<td>53.8(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JetI</td>
<td>3786(81)</td>
<td>1021(8)</td>
<td>361(4)</td>
<td>157(1)</td>
<td>46(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MadEvent</td>
<td>3902(5)</td>
<td>1012(2)</td>
<td>361(1)</td>
<td>135.5(3)</td>
<td>53.6(2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of jets</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPGEN</td>
<td>5423(9)</td>
<td>1291(13)</td>
<td>465(2)</td>
<td>182.8(8)</td>
<td>75.7(8)</td>
<td>32.5(2)</td>
<td>13.9(2)</td>
</tr>
<tr>
<td>AMEGIC++</td>
<td>5432(5)</td>
<td>1277(2)</td>
<td>466(2)</td>
<td>184(1)</td>
<td>77.3(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompHEP</td>
<td>5485.8(6)</td>
<td>1287.5(7)</td>
<td>467.3(8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GR@PPA</td>
<td>5434(7)</td>
<td>1273(2)</td>
<td>467.7(9)</td>
<td>181.8(5)</td>
<td>76.6(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JetI</td>
<td>5349(143)</td>
<td>1275(12)</td>
<td>487(3)</td>
<td>212(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MadEvent</td>
<td>5433(8)</td>
<td>1277(2)</td>
<td>464(1)</td>
<td>182(1)</td>
<td>75.9(3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Parton Shower Concept

- Multi-parton amplitudes exhibit soft & collinear singularities.
- If \( b \) and \( c \) adjacent in color and collinear, they can be thought to originate from a common mother parton \( a \) with \( p_b = z p_a \) and \( p_c = (1 - z) p_a \).
- \( n+1 \) parton amplitude factorizes into \( n \) parton amplitude and invariant 1\(\rightarrow\)2 splitting acc. to
  \[
d\sigma_{n+1} = d\sigma_n \otimes \sum_{a \in q, g} \frac{dt}{t} \frac{dz}{z} \frac{\alpha_s(t, z)}{2\pi} P_{a \rightarrow bc}(z)
\]
- No interference of daughter partons with remaining partons.
  - allows probabilistic interpretation \( \Rightarrow \) Markov chain
- But: Must respect angular ordering (color coherence)!
- Ambiguities: \( z(1 - z) t = k_t^2 \); \( t/z(1 - z) \approx E^2 \theta^2 \) yield formally equivalent evolution
  - \( z \) \( \Rightarrow \) light-cone momentum or energy fraction
  - Starting scale not fixed (process dependent)
Parton Shower Issues

- Test of different PS prescriptions in PYTHIA
  (by P. Skands, T. Plehn, D. Rainwater
  presented at TeV4LHV, CERN, 29.4.2005)
Matrix Element vs. Parton Shower

Matrix Elements

- Exact to fixed order in running coupling $\alpha$
- Include all quantum interferences
- Calculable only for low FS multiplicity ($n \leq 6-8$)

Parton Showers

- Resum (next-to) leading logarithms to all orders
- Interference effects e.g. through angular ordering

$\sigma_{n+1} = \sigma_n \otimes \sum_{b \in q, g} \frac{dt}{t} \frac{d\alpha_s(t, z)}{2\pi} P_{a\to b}(z)$

Desirable to combine both approaches to have
- Good description of hard/wide-angle emissions (ME)
- Correct intrajet evolution (PS)

Must prevent double counting e.g. through CKKW
Combining ME & PS à la CKKW

- Define jet resolution parameter $Q_{\text{cut}}$ (Q-jet measure)
  - divide phase space into regions of jet production (n-jet ME) & jet evolution (PS)
- Select jet multiplicity and kinematics according to $\sigma$ 'above' $Q_{\text{cut}}$
- $K_T$ cluster backwards (construct PS tree) and identify core process
- Reweight ME to get exclusive samples at resolution scale $Q_{\text{cut}}$
- Start PS at scale $\mu_{\text{hard}}$, reject all emissions above $Q_{\text{cut}}$

Yields correct jet rates, e.g. 2-jet rate in 2-jet event at scale $q$

$$ R_2(q^2) = \left( \Delta(Q_{\text{cut}}, \mu_{\text{hard}}) \frac{\Delta(q, \mu_{\text{hard}})}{\Delta(Q_{\text{cut}}, \mu_{\text{hard}})} \right)^2 $$
There exist several variants of the above algorithm ...

- **The MLM prescription** *(Nucl. Phys. B632 343 (2002))*
  - Employ cone algorithm to define jets after PS
  - Match N+M reconstructed jets to N ME partons
  - Different samples can be added w/o need for analytic Sudakovs

- **The Lönnblad prescription** *(JHEP 0205 046 (2002))*
  - Employs dipole cascade instead of PS
  - Sudakov weights are calculated using the cascade itself

**Similar systematics for all algorithms**

- Residual dependence on the phase space separation cut
- Variations with the number of hard ME partons
- Dependencies on the internal jet algorithm
Comparison of CKKW as implemented in Sherpa with PYTHIA and MC@NLO in W+jets at LHC (hep-ph/0503280)

- Sherpa uses $Q_{\text{cut}}=20\text{ GeV}$ and $N_{\text{max jet}}=1$
- MC@NLO in default mode (NLO)
- PYTHIA with PS starting scale (14TeV)$^2$

Rates in CKKW are still LO!
Combining ME & PS

- $\Delta \phi$ of two hardest jets in Z+jets at LHC from Sherpa (hep-ph/0503280)
Detailed comparison of merging approaches started

$E_T$ spectra of jets in $pp \rightarrow e^+\bar{\nu}_e + X$
Detailed comparison of merging approaches started

$E_T$ spectra of jets in $pp \rightarrow e^+ \bar{\nu}_e + X$

ALPGEN $\mu_R$ rescaled by 0.5 (hint on scale uncertainty)
Now the MCs ...
General Purpose MCs: PYTHIA

T. Sjöstrand, L. Lönnblad, S. Mrenna, P. Skands, ... hep-ph/0308153

- THE standard event generator (FORTRAN version)
- Includes large collection of pre-calculated hard matrix elements (2$\rightarrow$2 and 2$\rightarrow$3 processes)
- PS: virtuality ordered w/ angular veto and $z \rightarrow$ energy fraction
- $k_T$ ordered with $z \rightarrow$ light cone momentum fraction (since version 6.3)
- Lund string fragmentation (and others)
- No automatic ME Generator, no ME-PS merging

Currently being rewritten in C++
(PYTHIA 8, proposed to be useable by mid 2007)
Robust and fast general purpose MC (handles ‘everything’)

Stefan Höche
General Purpose MCs: HERWIG


- **General purpose MC** (FORTRAN version)
- Large collection of **pre-calculated hard matrix elements** (2→2 and 2→3 processes)
- **Angular ordered PS** w/ full spin correlations
- **Cluster fragmentation model**
- Models for hard (JIMMY hep-ph/9601371) and soft UE
- **No automatic ME Generator**, no ME-PS merging

Currently being rewritten in C++ (HERWIG++, improved PS) (recently tested in pp→Z (S. Gieseke et al. hep-ph/0602069))

New hadron decay package w/ web-interface by P. Richardson

**General purpose MC** like PYTHIA but different models
General Purpose MCs: Sherpa


- **Automatic ME Generator AMEGIC++** (JHEP 0202 (2002) 044)

- **virtuality ordered PS** similar to PYTHIA (hep-ph/0503087)

- currently relying on Lund string fragmentation (PYTHIA)
  but: Cluster fragmentation model ready (hep-ph/0311085)

- **Model for hard UE** (based on Sjöstrand/Zijl model)
  new model in preparation

- **Special emphasis on ME-PS merging** (hep-ph/0205283)

- **New: Hadron decay package** (tested in \(\tau\) decays vs. Tauola)

Written in C++ from scratch

MSSM part tested against MadGraph & WHIZARD

Well tested ME Generator, general purpose MC, original CKKW
(S)MadGraph / MadEvent

K. Hagiwara, F. Maltoni, T. Plehn, D. Rainwater, T. Stelzer

- **Automatic ME Generator MadGraph** (hep-ph/9401258)
  based on HELAC (KEK-91-11)
  (SM, MSSM (SMadGraph) (hep-ph/0512260), Higgs EFT)

- **Single diagram enhanced integration / event generation**
  (suitable basis for multi-channel independent weights (hep-ph/0208156))

- **LHA interface** → full events from HERWIG/PYTHIA

- **ME-PS merging à la CKKW / MLM** in preparation

Parallel nature of integration perfect for PC cluster

Web interface: [http://madgraph.hep.uiuc.edu](http://madgraph.hep.uiuc.edu)

MSSM part tested against AMEGIC++ & WHIZARD

**Very fast SM / MSSM ME Generator,**
large multiplicities (6–7 FS particles e.g. in W+jets)
ALPGEN


- Collection of processes calculated in the ALPHA algorithm
- Full events from HERWIG/PYTHIA
- Phase space not "multi-channelled"; uses adaptive algorithm
- ME-PS merging in the MLM prescription

ALPGEN provides full spin & color information to the PS MC
Currently largest multiplicities, original MLM

Processes currently available in the package:

- W Q Qbar + up to 4 jets
- Z/gamma* Q Qbar + up to 4 jets
- W + up to 6 jets
- W + charm + up to 5 jets
- Z + up to 6 jets
- nW+mZ+kH + up to 3 jets
- Q Qbar plus up to 6 jets
- Q Qbar Q' Qbar' plus up to 4 jets
- Q Qbar Higgs plus up to 4 jets
- Inclusive N jets, with N up to 6
- N photons + M jets, with N larger than 0, N+M up to 8 and M up to 6
- **NEW** Higgs + N jets, with N<5
- **NEW** Single top: tq, tb, tW, tbW. No extra jets.
WHIZARD / O’Mega

WHIZARD:
W. Kilian (LC-TOOL-2001-39)

- General integration & event generation package
  for O’Mega, MadGraph, CompHep, ...
  (up to six particle final states)
- Interfaces PYTHIA to generate full events

O’Mega:
T. Ohl, M. Moretti, J. Reuter (hep-ph/0102195)

- Generator-generator based on ALPHA algorithm
  many new physics models (spin 2 particles, gravitinos, AGC)
  very flexible, easy to extend
- MSSM part tested against AMEGIC++ & MadGraph
Collection of processes calculated at NLO

PS and hadronization from HERWIG 6.5

NLO-MC merging according to the MC@NLO algorithm

NLO cross sections and shapes

Can be used to calculate any observable at hadron level

<table>
<thead>
<tr>
<th>IPROC</th>
<th>IV</th>
<th>IL1</th>
<th>IL2</th>
<th>Spin</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1350-IL</td>
<td>✓</td>
<td>H1H2 → (Z/γ∗ →)l_\text{IL}l_\text{IL} + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1360-IL</td>
<td>✓</td>
<td>H1H2 → (Z →)l_\text{IL}l_\text{IL} + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1370-IL</td>
<td>✓</td>
<td>H1H2 → (γ∗ →)l_\text{IL}l_\text{IL} + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1460-IL</td>
<td>✓</td>
<td>H1H2 → (W:+ →)l_\text{IL}\nu_\text{IL} + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1470-IL</td>
<td>✓</td>
<td>H1H2 → (W:- →)l_\text{IL}\nu_\text{IL} + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1396</td>
<td>×</td>
<td>H1H2 → γ∗(→ \sum f_il_j) + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1397</td>
<td>×</td>
<td>H1H2 → Z:⁰ + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1497</td>
<td>×</td>
<td>H1H2 → W:+ + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1498</td>
<td>×</td>
<td>H1H2 → W:- + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1600-ID</td>
<td></td>
<td>H1H2 → H:⁰ + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1705</td>
<td></td>
<td>H1H2 → b:b + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1706</td>
<td>×</td>
<td>H1H2 → t:t + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2000-1C</td>
<td>×</td>
<td>H1H2 → t:l + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2001-1C</td>
<td>×</td>
<td>H1H2 → t + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2004-1C</td>
<td>×</td>
<td>H1H2 → t + X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2600-ID</td>
<td>1</td>
<td>7</td>
<td>×</td>
<td>H1H2 → H:⁰W:+ + X</td>
<td></td>
</tr>
<tr>
<td>-2600-ID</td>
<td>1</td>
<td>i</td>
<td>✓</td>
<td>H1H2 → H:⁰(W:+ →)l:_i\nu:_i + X</td>
<td></td>
</tr>
<tr>
<td>-2600-ID</td>
<td>-1</td>
<td>7</td>
<td>×</td>
<td>H1H2 → H:⁰W:- + X</td>
<td></td>
</tr>
<tr>
<td>-2600-ID</td>
<td>-1</td>
<td>i</td>
<td>✓</td>
<td>H1H2 → H:⁰(W:- →)l:_i\nu:_i + X</td>
<td></td>
</tr>
<tr>
<td>-2700-ID</td>
<td>0</td>
<td>7</td>
<td>×</td>
<td>H1H2 → H:⁰Z + X</td>
<td></td>
</tr>
<tr>
<td>-2700-ID</td>
<td>0</td>
<td>i</td>
<td>✓</td>
<td>H1H2 → H:⁰(Z →)l:_i + X</td>
<td></td>
</tr>
<tr>
<td>-2850</td>
<td>7</td>
<td>7</td>
<td>×</td>
<td>H1H2 → W:+W:+ + X</td>
<td></td>
</tr>
<tr>
<td>-2850</td>
<td>i</td>
<td>j</td>
<td>✓</td>
<td>H1H2 → (W:+ →)l:_i\nu:(W:+ →)l:_j\nu:_j + X</td>
<td></td>
</tr>
<tr>
<td>-2860</td>
<td>7</td>
<td>7</td>
<td>×</td>
<td>H1H2 → Z:⁰Z:⁰ + X</td>
<td></td>
</tr>
<tr>
<td>-2870</td>
<td>7</td>
<td>7</td>
<td>×</td>
<td>H1H2 → W:+Z:⁰ + X</td>
<td></td>
</tr>
<tr>
<td>-2880</td>
<td>7</td>
<td>7</td>
<td>×</td>
<td>H1H2 → W:-Z:⁰ + X</td>
<td></td>
</tr>
</tbody>
</table>
Other Tools

ME Generators:  
- **GRACE / Gr@ppa**
  J. Fujimoto et al. (hep-ph/0007053)
  **SM / MSSM processes** (tree level)
  BASES/SPRING for integration/
event generation
- **CompHEP**
  A. Pukhov et al. (hep-ph/9908288)
  **SM / MSSM processes**
  (tree level up to four FS particles)

NLO codes:  
- **MCFM**
  J. Campbell, K. Ellis (hep-ph/0006304)
  various processes, see [http://mcfm.fnal.gov](http://mcfm.fnal.gov)
- **NLOjet++**
  Z. Nagy, D. Soper; [http://www.cpt.dur.ac.uk/~nagyz/nlo++-v1/](http://www.cpt.dur.ac.uk/~nagyz/nlo++-v1/)
  pp: up to 3 jets at NLO / 4 jets at LO
Conclusions

- Many good tree level and NLO tools on the market
- So far only one approach for full NLO simulation
  But: To predict correct shapes ME-PS merging à la CKKW/MLM is often sufficient
  → we get the shapes of real emissions right!
- Need more input from experimental community in order to fix systematics (e.g. CKKW/MLM)

Hot topics: 
- Shower algorithms (still !)
- Tree-level ME’s (new recursion relations for QCD)
- Underlying events
- ...

Stefan Höche