Monte Carlo event generation: Introduction

Marek Schönherr

Institute for Particle Physics Phenomenology

Dakar, 06/08/2014
Organisation

1. Monte Carlo event generation: Introduction I & II
   - Organisation of the calculation
   - Fixed-order calculations
   - Monte Carlo integration

2. Monte Carlo event generation: Introduction III & IV
   - Parton showers
   - Multiple parton interactions
   - Hadronisation, hadron decays, QED corrections
Organisation

1 Tutorial I
   - Structure of a Monte Carlo event

2 Tutorial II
   - Effect & impact of the individual event stages
   - Understand the consequences of the employed approximations
Literature

- F. James
  *Monte Carlo theory and practice*

- R. K. Ellis, W. J. Stirling, B. R. Webber
  *QCD and Collider Physics*
  Cambridge University Press, 2003

- T. Sjöstrand, S. Mrenna, P. Z. Skands
  *PYTHIA 6.4 Physics and Manual*
  JHEP 05 (2006) 026

- A. Buckley et al.
  *General-Purpose Event Generators for LHC Physics*
  Phys. Rept. 504 (2011) 145

- R. D. Field
  *Applications of Perturbative QCD*
  Addison-Wesley, 1995
Monte Carlo event generation: Introduction I

1. Event generators
   A hadron collider event
   Divide et impera

2. Fixed-order calculation
   Leading order matrix elements
   Next-to-leading order matrix elements

3. Monte Carlo integration
   Text book methods
   Monte Carlo integration

4. Summary
Monte Carlo event generation: Introduction I

1. Event generators
   A hadron collider event
   Divide et impera

2. Fixed-order calculation
   Leading order matrix elements
   Next-to-leading order matrix elements

3. Monte Carlo integration
   Text book methods
   Monte Carlo integration

4. Summary
Aim

Try to falsify theoretical models by comparison with data:

- define observables (operative instruction to assign a number to a given configuration)
  → needs to be realisable experimentally
  - cross section of process $pp \rightarrow X$
    → defined in terms of identified particles, acceptance cuts, isolation criteria, etc.
  - jet quantities ($p_\perp$, $\eta$, $\Delta\phi$, ...)
    → defined in terms of a jet algorithm which takes energy deposits, tracks, energy flows, etc.

- calculate observables in given model, e.g. Standard Model, BSM, etc.

- measure observables in data

⇒ compare
## Task: Compute the probability for this

**Image:** CMS Experiment at the LHC, CERN

Data recorded: 2013-run-25 00:34:23.96783 GMT (08:34:20 CEST)
Run / Event: 167476 / 076091767

---

### Summary

- **Event generators**
- **Fixed-order calculation**
- **Monte Carlo integration**

---

*Marek Schönherr*
Task: Compute the probability for this
Task: Compute the probability for this
Task: Compute the probability for this

A typical Drell-Yan event:

\[ pp \rightarrow \ell^+ \ell^- + O(100) \gamma, e^\pm, \mu^\pm, \pi^\pm, K^\pm, K_L, p^\pm, n, (\nu) \]

A typical 4\(\ell\) event:

\[ pp \rightarrow 4\ell + O(150) \gamma, e^\pm, \mu^\pm, \pi^\pm, K^\pm, K_L, p^\pm, n, (\nu) \]

A typical hadronic \(t\bar{t}\) event:

\[ pp \rightarrow O(700) \gamma, e^\pm, \mu^\pm, \pi^\pm, K^\pm, K_L, p^\pm, n, (\nu) \]

A typical semi-leptonic \(t\bar{t}h\) event:

\[ pp \rightarrow \ell + O(1200) \gamma, e^\pm, \mu^\pm, \pi^\pm, K^\pm, K_L, p^\pm, n, (\nu) \]

We want to make predictions that can be compared to experimental data, thus we have to describe the final state at the stage where it enters the detector.
Divide et impera

Event structure
Factorise into event stages according to characteristic scales, use relevant approximation in each regime

- Hard scattering
- Parton evolution
- Multiple interactions
- Hadronisation
- Hadron decays
- QED corrections
Event generator for LHC physics

**HERWIG**
- originated in coherent shower studies → angular order PS
- first Mc@NLO and POWHEG developments
- simple in-house ME generator & spin-correlated decay chains
- original cluster fragmentation

**PYTHIA**
- originated in hadronisation studies → Lund string
- leading in development of multiple interaction models
- pragmatic attitude to ME generation → external tools
- extensive PS development and earliest MEPS matching

**SHERPA**
- started with PS generator APACIC++ & ME generator AMEGIC++
- current MPI & hadronisation models pragmatic add-ons
- leading in development of automated MEPS merging
- automated framework for NLO calculations and Mc@NLO
Monte Carlo event generation: Introduction I

1 Event generators
   A hadron collider event
   Divide et impera

2 Fixed-order calculation
   Leading order matrix elements
   Next-to-leading order matrix elements

3 Monte Carlo integration
   Text book methods
   Monte Carlo integration

4 Summary
Fixed-order calculations

\[ \langle O \rangle = \int d\Phi \int_0^1 dx_a \int_0^1 dx_b \ f_a(x_a, \mu_F^2) f_b(x_b, \mu_F^2) \ |M(ab \rightarrow X; \mu_R^2, \mu_F^2)|^2 O(\Phi) \]

**First task:**
- calculate matrix element
  - 1st approximation: leading order calculation
  - 2nd approximation: next-to-leading order calculation

**Second task:**
- do integral → 3rd part of this lecture
Availablity of parton-level calculations

- exact $n$-parton perturbative calculation often best, but typically hard to do for “many” external legs
- tree-level case is highly automated
- one-loop case getting there
Leading order matrix elements

LO calculation Born term: \[ B = \sum_{\text{colour}} |A|^2 \sum_{\text{spin}} \tilde{M}_{\text{tree}}^2 \]

\[ \langle O \rangle^{\text{LO}} = \int d\Phi_B B(\Phi_B) O(\Phi_B) \]

Textbook:
- calculate amplitudes using Feynman diagrams
- use completeness relations to square them
- sum/average over external states (helicity and colour)
⇒ computational effort grows quadratically with number of diagrams
Leading order matrix elements

LO calculation Born term: \( B = \sum_{\text{colour}} |A|^2 \sum_{\text{spin}} \left| \tilde{M}_{\text{tree}} \right|^2 \)

\[ \langle O \rangle^{LO} = \int d\Phi_B \ B(\Phi_B) \ O(\Phi_B) \]

Textbook:
- calculate amplitudes using Feynman diagrams
- use completeness relations to square them
- sum/average over external states (helicity and colour)

\( \Rightarrow \) computational effort grows quadratically with number of diagrams
Leading order matrix elements – helicity amplitudes

LO calculation Born term: $B = \int \sum_{\text{colour}} |\mathcal{A}|^2 \sum_{\text{spin}} \tilde{M}_{\text{tree}} \cdot \tilde{M}_{\text{tree}}^*$

**Improvements:**

- **amplitudes are complex numbers**
  - first compute amplitudes using Feynman diagram, then add, square
  - computational effort grows linearly with number of diagrams

- **matrix multiplication is expensive**
  - write everything as spinor products, “gauge” vector fields
    - e.g. $\bar{u}(p_1, h_1)u(p_2, h_2) \in \mathbb{C}$,
    - $\bar{u}(p_1, h_1)\gamma^\mu u(p_2, h_2) \in \mathbb{C}^{3+1}$,
    - $\bar{u}(p_1, h_1)\gamma^\mu u(p_2, h_2) \bar{u}(p_3, h_3)\gamma_\mu u(p_3, h_3) \in \mathbb{C}$, ...

  - $\Rightarrow$ helicity amplitudes, explicit spin information
Leading order matrix elements – helicity amplitudes

LO calculation Born term: \[ B = \sum_{\text{colour}} |A|^2 \sum_{\text{spin}} \tilde{M}_{\text{tree}} \cdot \tilde{M}_{\text{tree}}^* \]

Improvements:

- amplitudes are complex numbers
  - first compute amplitudes using Feynman diagram, then add, square
  \[ \Rightarrow \text{computational effort grows linearly with number of diagrams} \]
- matrix multiplication is expensive
  - write everything as spinor products, “gauge” vector fields
    e.g. \[ \bar{u}(p_1, h_1)u(p_2, h_2) \in \mathbb{C}, \]
    \[ \bar{u}(p_1, h_1)\gamma^\mu u(p_2, h_2) \in \mathbb{C}^{3+1}, \]
    \[ \bar{u}(p_1, h_1)\gamma^\mu u(p_2, h_2) \bar{u}(p_3, h_3)\gamma_\mu u(p_3, h_3) \in \mathbb{C}, \ldots \]
  \[ \Rightarrow \text{helicity amplitudes, explicit spin information} \]
Leading order matrix elements – book-keeping

LO calculation Born term: \( B = \sum_{\text{colour}} |A|^2 \sum_{\text{spin}} \tilde{M}_{\text{tree}} \cdot \tilde{M}_{\text{tree}}^* \)

Improvements:

- recurring subgraphs
  many Feynman diagrams share same subgraphs

⇒ book-keep subamplitudes and reuse
Leading order matrix elements – recurrence relations

LO calculation Born term: \( B = \sum_{\text{colour}} |A|^2 \sum_{\text{spin}} \tilde{M}_{\text{tree}} \cdot \tilde{M}^*_{\text{tree}} \)

**Improvements:**

- complexity of Feynman diagram grows factorial with number of external legs

\[ J^\mu = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]

\[ = \sum_{i=2}^{n-1} i + 2 \]
Leading order matrix elements – helicity & colour sums

LO calculation Born term: \( B = \sum_{\text{colour, spin}} A\tilde{M}_{\text{tree}} \cdot \tilde{M}^*_\text{tree} A^* \)

Improvements:

- not all helicity configurations contribute equally
  not applicable when using traces/completeness relations
  \( \Rightarrow \) calculate amplitudes for only one helicity configuration, include helicity dof. in phase space integral
- not all colour configurations contribute equally
  not applicable when colour matrix has been separated from Lorentz-structure of matrix elements, e.g. when using helicity methods
  \( \Rightarrow \) calculate amplitudes for only one colour configuration, include colour dof. in phase space integral
# Leading order matrix elements

Computation time (s) of the $2 \rightarrow n$ gluon amplitudes for $10^4$ phase space points for one random helicity and colour configuration

<table>
<thead>
<tr>
<th>Final State</th>
<th>BG</th>
<th>BCF</th>
<th>CSW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CO</td>
<td>CO</td>
<td>CO</td>
</tr>
<tr>
<td>2g</td>
<td>0.24</td>
<td>0.28</td>
<td>0.33</td>
</tr>
<tr>
<td>3g</td>
<td>0.45</td>
<td>0.42</td>
<td>0.51</td>
</tr>
<tr>
<td>4g</td>
<td>1.20</td>
<td>0.84</td>
<td>1.32</td>
</tr>
<tr>
<td>5g</td>
<td>3.78</td>
<td>2.59</td>
<td>7.26</td>
</tr>
<tr>
<td>6g</td>
<td>14.2</td>
<td>11.9</td>
<td>59.1</td>
</tr>
<tr>
<td>7g</td>
<td>58.5</td>
<td>73.6</td>
<td>646</td>
</tr>
<tr>
<td>8g</td>
<td>276</td>
<td>597</td>
<td>8690</td>
</tr>
<tr>
<td>9g</td>
<td>1450</td>
<td>5900</td>
<td>127000</td>
</tr>
<tr>
<td>10g</td>
<td>7960</td>
<td>64000</td>
<td>489000</td>
</tr>
</tbody>
</table>
Next-to-leading order matrix elements

NLO calculation

Born term: \( B = \int \)

Virtual terms: \( V = 2 \text{Re} \left\{ \right\} \)

Real terms: \( R = \int \)

- UV divergences in \( V \) removed by renormalization procedure
- \( V \) and \( R \) both still infrared divergent
- IR divergences cancel between \( V \) and \( R \) (KLN theorem)
  \( \rightarrow \) finite result for IR safe observables
Next-to-leading order matrix elements

NLO calculation

\[ \langle O \rangle^{\text{NLO}} = \int d\Phi_B \left[ B(\Phi_B) + V(\Phi_B) \right] O(\Phi_B) \]
\[ + \int d\Phi_R R(\Phi_R) O(\Phi_R) \]

- IR divergences cancel between $V$ and $R$ (KLN theorem), but live in different phase spaces
  - IR divergences in $V$ arise from integral over loop momentum
  - IR divergences in $R$ arise from integral over soft-collinear external momentum

- **subtraction method**: construct universal integrable terms that reproduce $R$ in the soft-collinear limit
Subtraction method

NLO calculation

\[
\langle O \rangle^{\text{NLO}} = \int d\Phi_B \left[ B(\Phi_B) + V(\Phi_B) + \int d\Phi_1 D(\Phi_B \cdot \Phi_1) \right] O(\Phi_B)
+ \int d\Phi_R \left[ R(\Phi_R) O(\Phi_R) - D(\Phi_B \cdot \Phi_1) O(\Phi_B) \right]
\]

- **subtraction method**: construct universal integrable terms \( D \) that reproduce \( R \) in the soft-collinear limit
- holds for infrared-safe observables, i.e. \( O(\Phi_R) \rightarrow O(\Phi_B) \) in IR limit
- need to add \( \int d\Phi_1 D \) back
  \( \rightarrow \) cancels divergences in \( V \) (KLN)
  \( \Rightarrow \) integrands of both phase space integrals separately finite
Subtraction method

NLO calculation

\[ \langle O \rangle^{\text{NLO}} = \int d\Phi_B \left[ B(\Phi_B) + V(\Phi_B) + \int d\Phi_1 D(\Phi_B \cdot \Phi_1) \right] O(\Phi_B) \]
\[ + \int d\Phi_R \left[ R(\Phi_R) O(\Phi_R) - D(\Phi_B \cdot \Phi_1) O(\Phi_B) \right] \]

- **subtraction method**: construct universal integrable terms \( D \) that reproduce \( R \) in the soft-collinear limit
- holds for infrared-safe observables, i.e. \( O(\Phi_R) \rightarrow O(\Phi_B) \) in IR limit
- need to add \( \int d\Phi_1 D \) back
  \( \rightarrow \) cancels divergences in \( V \) (KLN)
  \( \Rightarrow \) integrands of both phase space integrals separately finite
Subtraction method

NLO calculation

\[ \langle O \rangle^{\text{NLO}} = \int d\Phi_B \left[ B(\Phi_B) + V(\Phi_B) + \int d\Phi_1 D(\Phi_B \cdot \Phi_1) \right] O(\Phi_B) \]
\[ + \int d\Phi_R \left[ R(\Phi_R) O(\Phi_R) - D(\Phi_B \cdot \Phi_1) O(\Phi_B) \right] \]

- **subtraction method**: construct universal integrable terms \( D \) that reproduce \( R \) in the soft-collinear limit
- holds for infrared-safe observables, i.e. \( O(\Phi_R) \rightarrow O(\Phi_B) \) in IR limit
- need to add \( \int d\Phi_1 D \) back
  \( \rightarrow \) cancels divergences in \( V \) (KLN)

\( \Rightarrow \) integrands of both phase space integrals separately finite
Subtraction method

NLO calculation

\[ \langle O \rangle^{\text{NLO}} = \int \! d\Phi_B \left[ B(\Phi_B) + V(\Phi_B) + \int \! d\Phi_1 D(\Phi_B \cdot \Phi_1) \right] O(\Phi_B) \]

\[ + \int \! d\Phi_R \left[ R(\Phi_R) \ O(\Phi_R) - D(\Phi_B \cdot \Phi_1) \ O(\Phi_B) \right] \]

- **subtraction method**: construct universal integrable terms \( D \) that reproduce \( R \) in the soft-collinear limit
- holds for infrared-safe observables, i.e. \( O(\Phi_R) \rightarrow O(\Phi_B) \) in IR limit
- need to add \( \int \! d\Phi_1 D \) back
  \( \rightarrow \) cancels divergences in \( V \) (KLN)
- \( \Rightarrow \) integrands of both phase space integrals separately finite
Subtraction method

Existing subtraction methods

- Catani-Seymour dipole subtraction  
  implemented in AMEGIC++, COMIX, MADDIPOLE, ...  
  Catani, Seymour NPB485(1997)291

- Frixione-Kunszt-Signer subtraction  
  implemented in MADFKS  
  Frixione, Kunszt, Signer NPB467(1996)

- Antenna subtraction  
  Kosower PRD57(1998)5410

- Nagy-Soper subtraction  
  implemented in HELAC  
  Nagy, Soper JHEP09(2007)114
Real and virtual correction

Real correction: $R = \int \ldots \Rightarrow$ tree-level, same technologies as for $B$

Virtual correction: $V = \int \ldots$

- reduce 1-loop matrix element to master integrals

$$\mathcal{M}^{1\text{-loop}} = D + C + B + A + R$$

- compute coefficients either with tensor reduction or unitarity cuts
- problem: numerical stability, may need quad-precision
Commonly used matrix element generators (selection)

- many automated tools with different technologies

<table>
<thead>
<tr>
<th>Models</th>
<th>$2 \rightarrow n$</th>
<th>MEs</th>
<th>Integration</th>
<th>NLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA (ALPGEN)</td>
<td>SM</td>
<td>$n = 8$</td>
<td>recursive</td>
<td>Multichannel</td>
</tr>
<tr>
<td>AMEGIC++ (SHERPA)</td>
<td>SM, BSM</td>
<td>$n = 6$</td>
<td>hel. amps.</td>
<td>Multichannel</td>
</tr>
<tr>
<td>COMIX (SHERPA)</td>
<td>SM</td>
<td>$n = 8$</td>
<td>recursive</td>
<td>Multichannel</td>
</tr>
<tr>
<td>COMPHEP</td>
<td>SM, BSM</td>
<td>$n = 4$</td>
<td>trace</td>
<td>1Channel</td>
</tr>
<tr>
<td>HELAC</td>
<td>SM</td>
<td>$n = 8$</td>
<td>recursive</td>
<td>Multichannel</td>
</tr>
<tr>
<td>MADGRAPH (MADEVENT)</td>
<td>SM, BSM</td>
<td>$n = 6$</td>
<td>hel. amps.</td>
<td>Multichannel</td>
</tr>
<tr>
<td>OMEGA (WHIZARD)</td>
<td>SM, BSM</td>
<td>$n = 8$</td>
<td>recursive</td>
<td>Multichannel</td>
</tr>
</tbody>
</table>

Limiting factors:

- factorial growth of number of colour configurations
  → solution: colour dressing Duhr, Höche, Maltoni JHEP08(2006)062

- efficient phase space integration
  (matrix elements on their own are not very useful if they cannot be used to calculate cross sections)
  → see 3rd part of this lecture for some answers
One-loop matrix element generators

- for a long time only process specific codes:
  - MCFM
  - VBFNLO
  - NLOJET++
  - NJET++

- recently: (semi-)automated codes
  - BLACKHAT
  - ROCKET
  - HELAC
  - MADLOOP
  - OPENLOOPS

- typically interfaced to leading order event generators that take care of leading order matrix elements & phase space integration
Limitations

• fixed-order implies fixed multiplicity
  higher-multiplicities are of higher order and therefore beyond
  approximation

• $pp \to e^+ e^-$ calculation is always a $pp \to e^+ e^- + X$ calculation
  (inclusive in $X$)

• breakdown of convergence of perturbation theory when phase space
  integration generates large logarithms in every order
  $\rightarrow$ need to rearrange pert. expansion to resum large logarithms
  $\rightarrow$ parton showers are one solution
  $\rightarrow$ see tomorrow’s lecture

• experimental definitions rely on observable hadrons
  $\rightarrow$ need hadron level generators
Monte Carlo event generation: Introduction I

1. Event generators
   - A hadron collider event
   - Divide et impera

2. Fixed-order calculation
   - Leading order matrix elements
   - Next-to-leading order matrix elements

3. Monte Carlo integration
   - Text book methods
   - Monte Carlo integration

4. Summary
Aim

\[ \langle O \rangle = \int d\Phi \int_0^1 dx_a \int_0^1 dx_b \ f_a(x_a, \mu_F^2) f_b(x_b, \mu_F^2) \left| M(ab \to X; \mu_R^2, \mu_F^2) \right|^2 O(\Phi) \]

So far, we have seen how to construct the integrand. But to compute observables we need to do the integral. As the phase space structure of the observable might be infinitely complex analytic integration is out of the question.

Turn to numerical methods:

a) text book methods
b) Monte Carlo integration

Start with simple example

\[ I_f^{(a,b)} = \int_a^b dx \ f(x) \]

Integral of an arbitrary function on 1 dim. phase space with boundaries
Text book methods

Find solution to

\[ I_f^{(a,b)} = \int_a^b \, dx \, f(x) \]

Rectangle method

- divide interval \([a, b]\) in \(n\) subintervals of size \(\Delta x = (b - a)/n\)

\[ I_f^{(a,b)} \approx \sum_{i=0}^{n-1} f(x_i) \Delta x \quad \text{with} \quad x_i \in \left[a + i\Delta x, a + (i + 1)\Delta x\right] \]

⇒ replace integration by sum over rectangles

- error scales with \(\frac{1}{n}\) (\(n\) number of function evaluations)
Text book methods

Find solution to

$$I_f^{(a, b)} = \int_a^b dx f(x)$$

Trapezoid method

- divide interval \([a, b]\) in \(n\) subintervals of size \(\Delta x = (b - a)/n\)

$$I_f^{(a, b)} \approx \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2} \Delta x \quad \text{with} \quad x_i = a + i\Delta x$$

⇒ replace integration by sum over trapezoids
- error scales with \(\frac{1}{n^2}\) (\(n\) number of function evaluations)
Text book methods

Find solution to

\[ I_f^{(a,b)} = \int_a^b dx \ f(x) \]

Trapezoid method

- divide interval \([a, b]\) in \(n\) subintervals of size \(\Delta x = (b - a)/n\)

\[ I_f^{(a,b)} \approx \frac{f(a) + f(b)}{2} \Delta x + \sum_{i=1}^{n-1} f(x_i) \Delta x \quad \text{with} \quad x_i = a + i \Delta x \]

⇒ replace integration by sum over trapezoids

- error scales with \(\frac{1}{n^2}\) (\(n\) number of function evaluations)
Text book methods

Find solution to

\[ I_f^{(a,b)} = \int_a^b dx \, f(x) \]

Simpson’s rule

• divide interval \([a, b]\) in \(n/2\) subintervals of size \(\Delta x = 2(b - a)/n\)

\[ I_f^{(a,b)} \approx \sum_{i=0}^{n-2} \frac{f(x_i) + 4f(x_{i+1}) + f(x_{i+2})}{6} \Delta x \quad \text{with} \quad x_i = a + \frac{i \, \Delta x}{2} \]

⇒ replace integration by sum over areas under parabolas

• error scales with \(\frac{1}{n^4}\) (\(n\) number of function evaluations)
Textbook methods

Find solution to

\[ I_f^{(a,b)} = \int_a^b dx \ f(x) \]

Simpson’s rule

- divide interval \([a, b]\) in \(n/2\) subintervals of size \(\Delta x = 2(b - a)/n\)

\[ I_f^{(a,b)} \approx \frac{f(a) + f(b)}{6} \Delta x + \sum_{i=1}^{(n-1)/2} \frac{2f(x_{2i}) + f(x_{2i+1})}{3} \Delta x \quad \text{with} \quad x_i = a + i \frac{\Delta x}{2} \]

⇒ replace integration by sum over areas under parabolas

- error scales with \(\frac{1}{n^4}\) (\(n\) number of function evaluations)
Text book methods

Generalisation: Newton-Cotes method

• divide interval \([a, b]\) in \(n\) subintervals of size \(\Delta x = (b - a)/n\), approximate \(f(x)\) on interval by a polynomial \(P_p\) of degree \(p\), supported at \(p + 1\) equally spaced points, and weights \(w_i = w_i(P_p)\)

\[
I_f^{(a,b)} \approx \sum_{i=0}^{n} w_i f(x_i) \Delta x \quad \text{with} \quad x_i = a + i\Delta x
\]

• error scales with \(\frac{1}{n^{2p}}\) (\(n\) number of function evaluations)
• support points remain equally spaced
Text book methods

Gaussian quadrature

\[
I_f^{(a,b)} = \int_a^b \, dx \, g(x) \, w(x)
\]

\[
\approx \int_a^b \, dx \, P_m(x) \, w(x) = \sum_{i=1}^{m} \alpha_i \, g(x_i)
\]

where \( x_i \) roots of orthogonal polynomials \( P_m(x) \) of degree \( m \)

- common choices for \( P_m \): Legendre polynomials, Chebyshev polynomials, Laguerre polynomials, Hermite polynomials, etc.
- weights \( \alpha_i \) are specific for choice of polynomials and degree \( m \)
  \( \Rightarrow \) calculate once and tabulate
- error scales with \( n^{-2m-1} \)
  \( \Rightarrow \) variably spaced support points \( x_i \), pos. \( w_i \)
  \( \Rightarrow \) very well suited for one-dimensional integrals
Text book methods

**Issue:**

- Newton-Cotes and Gauss quadrature work very well in one dimension → methods of choice if you need to know the value of a one-dimensional integral
- rely on grid of support points to interpolate a polynomial
  ⇒ in $d$ dimensions need $n^d$ support points to achieve the same convergence behaviour
- the $N$ particle phase space has $d = 3N - 2$ dimensions
  → number of function calls for support grid grows as $n^{3N-2}$

⇒ **turn to Monte Carlo integration**
Monte Carlo integration – hit & miss method

\[ I_f^{(a,b)} = \int_a^b \, dx \, f(x) \]

- find \( f_{\text{max}} > f(x) \) on \([a, b]\), then \((b - a)f_{\text{max}} > I_f(a, b)\)
- sample random points \( \bar{x}_i = (x, y) \) on \([a, b] \otimes [0, f_{\text{max}}]\) and count
  * \(N_{\text{hit}}\) – number of points with \( y_i \leq f(x_i) \)
  * \(N_{\text{miss}}\) – number of points with \( y_i > f(x_i) \)
- then
  \[ I_f^{(a,b)} \approx \frac{N_{\text{hit}}}{N_{\text{hit}} + N_{\text{miss}}} \sqrt{\sigma} \propto \frac{1}{\sqrt{N_{\text{hit}} + N_{\text{miss}}}} \]
- works independent of the number of dimensions of integration range
Monte Carlo integration – hit & miss method – example

Consider \( f(x) = \sqrt{1 - x^2} \) on \([a, b] = [0, 1]\):

- sample random points on \([0, 1] \times [0, 1]\)
- count \( N_{\text{hit}} \) and \( N_{\text{miss}} \), \( V = 1 \)

<table>
<thead>
<tr>
<th>( N )</th>
<th>( N_{\text{hit}} )</th>
<th>( N_{\text{miss}} )</th>
<th>( I_f^{(a,b)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>7</td>
<td>3</td>
<td>0.7</td>
</tr>
<tr>
<td>100</td>
<td>79</td>
<td>21</td>
<td>0.79</td>
</tr>
<tr>
<td>( \downarrow )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \downarrow )</td>
</tr>
<tr>
<td>( N \to \infty )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \pi/4 )</td>
</tr>
</tbody>
</table>
Monte Carlo integration

**Mean value theorem:** if $f$ continuous, then $\exists \xi \in [a, b]: I_f^{(a,b)} = (b - a) f(\xi)$

$\rightarrow$ not interested in $\xi$, but identify $f(\xi) = \langle f \rangle$

$$I_f^{(a,b)} = (b-a)\langle f \rangle$$ with $$\langle f \rangle \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$ with $x_i \in [a, b]$ equally distributed

- calculate $\langle f \rangle$: evaluate integrand at random points
  $\rightarrow$ need to be equally distributed
  $\rightarrow$ (pseudo-)random number generators
- error scales with $\sqrt{N}$ independent of the dimension of the phase space

$$\sigma = (b - a) \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N - 1}}$$

$\Rightarrow$ simultaneously gives an error estimate
Monte Carlo integration

\[ \langle O \rangle = \int d\Phi_n \int_0^1 dx_a \int_0^1 dx_b \ f_a(x_a, \mu^2_F) \ f_b(x_b, \mu^2_F) \ |M(ab \rightarrow n; \mu^2_R, \mu^2_F)|^2 \ O(\Phi) \]

- Monte Carlo integration now means sampling the \( n \) particle (3\( n \) − 2 dimensional) phase space with random points \( \Phi_n \)

\[ \Rightarrow \] produce weight \( w_i = f_a(x_a, \mu^2_F) f_b(x_b, \mu^2_F) \ |M(ab \rightarrow n; \mu^2_R, \mu^2_F)|^2 \) for every \( \Phi_{n,i} \)

- Inteprete as event of configuration \( \Phi_{n,i} \) with weight \( w_i \)

- **Benefit:** simultaneously project onto arbitrary many observables

\[ \Rightarrow \] can compute the expectation value for arbitrary many observables simultaneously
Importance sampling

- **realistic scenario**: peaked integrand
  
  ⇒ most of the equally distributed points in regions that contribute little few points that carry the bulk of the integral
  → large variance

- **solution**: variable transform
  
  \[
  I = \int_a^b dx \ g(x) \ \frac{f(x)}{g(x)} = \int_{G(a)}^{G(b)} dG(x) \ w(x) \quad \text{with} \quad w(x) = \frac{f(x)}{g(x)}
  \]

  approximate \( f(x) \) with \( g(x) \) with known integral \( G(x) \)

  ⇒ throw more points were there is a large contribution to integral

- **Note**: \( I \) is independent of \( g, \sigma \) is not

  \[
  I = \left[ G(b) - G(a) \right] \langle w \rangle \quad \sigma = \left[ G(b) - G(a) \right] \sqrt{\frac{\langle w^2 \rangle - \langle w \rangle^2}{N - 1}}
  \]
Importance sampling – example

Consider $f(x) = \cos\left(\frac{\pi}{2} x\right)$ on $[a, b] = [0, 1]$

- choose $g(x) = 1 - x^2$, $G(x) = x - \frac{x^3}{3}$

$$I_f^{(0,1)} = (1 - 0) \cdot \langle f \rangle = \left(\frac{2}{3} - 0\right) \cdot \langle w \rangle$$

$\Rightarrow$ with knowledge of $\langle f \rangle$, $\langle f^2 \rangle$, $\langle w \rangle$, $\langle w^2 \rangle$

$$I_f^{(0,1)} = 0.637 \pm 0.308/\sqrt{N}$$

$\Rightarrow 0.637 \pm 0.032/\sqrt{N}$
Selecting from known distribution

- Random Number Generators (RNGs) produce uniformly distributed pseudo random numbers in $[0, 1]$.
- Assume we want points following distribution $g(x)$ with known integral $G(x)$ and its inverse $G^{-1}$ instead.
  - Probability to produce point in $[x, x + dx]$ is $g(x)dx$.
- Generate $x$ according to

$$\int_a^x d\bar{x} g(\bar{x}) = R \int_a^b d\bar{x} g(\bar{x})$$

where $R$ is a uniform random number in $[0, 1]$, then

$$x = G^{-1}\left[G(a) + R(G(b) - G(a))\right]$$
Stratified sampling

- **Problem:** Limited information on integrand
- decompose integral in $M$ sub-integrals

\[
\langle f \rangle = \sum_{i=1}^{M} \langle f_i \rangle \quad \sigma^2 = \sum_{i=1}^{M} \sigma_i^2
\]

- overall variance smallest, if variance in each contribution equal
  $\rightarrow$ sample most where fluctuations are largest
- algorithm:
  - divide integration region into bins (variable bin-size and/or weight)
  - adjust bins such that variance identical in all bins
- Example: VEGAS
Stratified sampling – example

Consider \( f(x) = \cos \left( \frac{\pi}{2} x \right) \) on \([a, b] = [0, 1]\)

- divide integration range into two subranges
- introduce weights \( \alpha_1 \) and \( \alpha_2 \) → govern how many points in each subrange
- adjust such, that \( \sigma_1 = \sigma_2 \)

\[
I_f^{(0,1)} = 0.637 \pm 0.308 / \sqrt{N}
\]

\[
\Rightarrow 0.637 \pm 0.147 / \sqrt{N}
\]
Multichannel

- **Problem:** integrand with intricate peak structure
- combine importance sampling and stratified sampling
  - importance sample with “eigenfunctions” $g_i(x)$, each approximating one peak of the integrand, assign weight $\alpha_i$

$$g(x) = \sum_{i=1}^{M} \alpha_i g_i(x)$$

- optimise $\alpha_i$ such that variance for each $\langle f_i \rangle$ is equal

⇒ method of choice when integrating multileg matrix elements
Unweighting

- recall from importance sampling

\[
\langle f \rangle = \frac{G(b) - G(a)}{b - a} \frac{1}{N} \sum_{i=1}^{N} w(x_i)
\]

\[
\sigma \propto \sqrt{\langle w^2 \rangle - \langle w \rangle^2}
\]

\[N - 1\]

\[\Rightarrow \text{variance minimal if } w(x) = w = \text{const.}, \text{i.e. all } x_i \text{ contribute the same to the integral, then}
\]

\[
\langle f \rangle = \frac{G(b) - G(a)}{b - a} w \quad \text{and} \quad \sigma = 0
\]

- however this implies \( g(x) = w \cdot f(x) \) and \( \langle f \rangle \) was known beforehand
Unweighting – Hit-and-miss method

- **realistically**: \( g(x) \) approximates \( f(x) \)
  \[ \rightarrow \] there exists a constant \( c \) such that \( c \cdot g(x) > f(x) \) for all \( x \)

- **Hit-and-miss method**
  - choose \( x_i \) according to \( g(x) \), if necessary replace \( g(x) = \sum_i \alpha_i g_i(x) \)
  - accept \( x_i \) with probability \( \frac{f(x)}{c \cdot g(x)} \), i.e. draw a uniform random number \( R \) on \([0, 1]\) and accept if \( \frac{f(x)}{c \cdot g(x)} < R \)
  - assign it the weight \( \langle f \rangle \)

- **problem**: \( c \) and \( \langle f \rangle \) not known beforehand, two run strategy
  - integrate inclusive observable first, e.g. incl. cross section
  - book keep max \( \frac{f(x)}{g(x)} \) to find minimal \( c \) and \( \langle f \rangle \)
  - in second run unweight and assign constant weight \( \langle f \rangle \)

\[ \Rightarrow \] optimally distributed phase space points of second run can be used to project onto observables, each \( x_i \) can be interpreted as event with same probability

- very costly if \( g(x) \) not a good approximation to \( f(x) \)
## Numerical integration

<table>
<thead>
<tr>
<th></th>
<th>In one dimension</th>
<th>In $d$ dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncertainty as a function of number of points $n$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monte Carlo</td>
<td>$n^{-1/2}$</td>
<td>$n^{-1/2}$</td>
</tr>
<tr>
<td>Rectangle rule</td>
<td>$n^{-1}$</td>
<td>$n^{-1/d}$</td>
</tr>
<tr>
<td>Trapezoid rule</td>
<td>$n^{-2}$</td>
<td>$n^{-2/d}$</td>
</tr>
<tr>
<td>Simpson’s rule</td>
<td>$n^{-4}$</td>
<td>$n^{-4/d}$</td>
</tr>
<tr>
<td>Newton-Cotes</td>
<td>$n^{-(2p)}$</td>
<td>$n^{-(2p)/d}$</td>
</tr>
<tr>
<td>Gauss rule</td>
<td>$n^{-(2m-1)}$</td>
<td>$n^{-(2m-1)/d}$</td>
</tr>
</tbody>
</table>
Summary: MC event generation I & II

- To calculate the probabilities of collider events they are structured/factorised into event stages.
- Fixed order matrix elements largely automated at LO and NLO level.
- Dedicated methods work best for one dimensional integrals: Newton-Cotes or Gauss methods.
- Monte-Carlo integration is the method of choice for multidimensional integration. Side-product: events with a statistical interpretation that can be projected onto arbitrary observables.
<table>
<thead>
<tr>
<th>Event generators</th>
<th>Fixed-order calculation</th>
<th>Monte Carlo integration</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thank you for your attention!