WHIZARD 2.x: A Monte Carlo Event Generator Generator for ILC (& LHC)

Thorsten Ohl

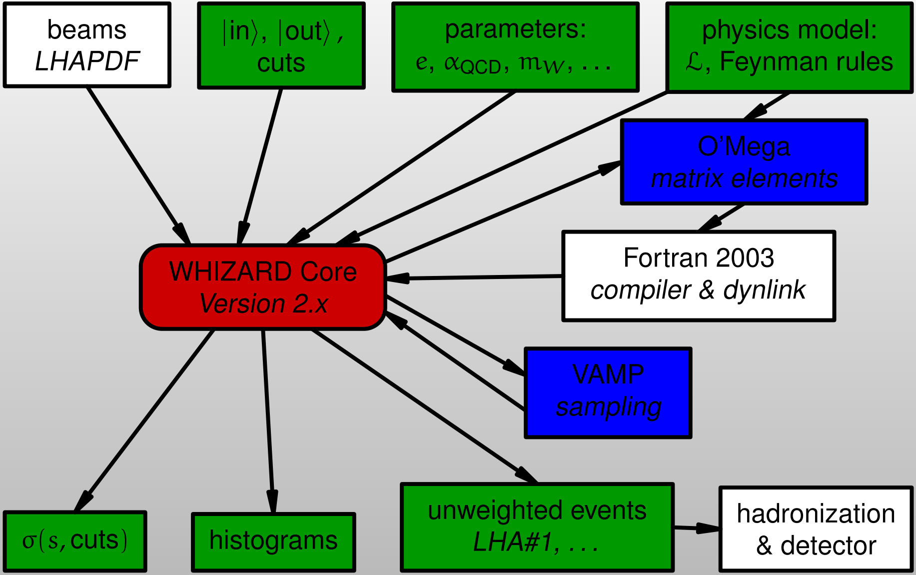
Institute for Theoretical Physics and Astrophysics, University of Würzburg

International Workshop on Linear Colliders 2010
Geneva, Switzerland
October 18-22, 2010

Contents

- O'Mega matrix element compiler:
  - maximally factorized matrix elements
  - exponential, not factorial, complexity (recursion relations)
  - model independent algorithm
- VAMP adaptive multi channel sampling:
  - multiple independent VEGAS grids
  - efficient integration of complicated phase spaces
  - unweighted event generation
- WHIZARD core:
  - phase space parametrization
  - unstable particle decays with spin correlations
  - interfaces, I/O
  - supervisor
  - scripting language

Architecture Version 2.x



- QED
- QCD
- Standard Model
- SM with anomalous top and gauge couplings
- Z'
- Supersymmetry: MSSM (cross checked with MadGraph and Sherpa), NMSSM, PSSSM
- Extra Dimensions, UED
- 3-Site Higgsless Model
- Little Higgs: Littlest, Simplest
- all FeynRules models
- your own …

- color build in from the ground up (finally!)
- compiles with a free Fortran 2003 compiler with reasonable performance:

  `gfortran` from version 4.5.0 on

- NAG Fortran useful for validation, not for production
- build environment fully GNU autotoolized
  - libtool
  - autoconf
  - automake
- standalone installable executable, dynamic linking of compiled matrix elements (no more `make` and `perl` glue required)
- flexible scripting language for integration, generation and analysis
- LHAPDF structure functions
- SLHA SUSY parameters

Simulate $e^+e^- \to ZH \to 4f$

- select the standard model

```
model = SM
```

- define aliases for (anti-)neutrinos, light (anti-)quarks

```
alias n = n1:n2:n3
alias N = N1:N2:N3
alias q = u:d:s:c
alias Q = U:D:S:C
```

- $b\bar{b} + E_{\text{missing}}$, i. e. $\sum_i b\bar{b}\nu_i\bar{\nu}_i$

```
process nnbb = e1, E1 => n, N, b, B
```

- $b\bar{b}jj$

```
process qqbb = e1, E1 => b, B, q, Q
```

- call O'Mega, compile and dynlink the matrix elements

```
compile
```

- travel back in time by 10 years:

```
sqrts = 209 GeV
```

- particle masses

```
mH = 115 GeV
wH = 3.228 MeV
mb = 2.9 GeV
me = 0
ms = 0
mc = 0
```

- very inclusive cuts: just kill intermediate photons

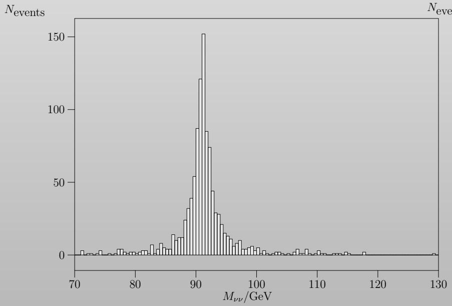
```
cuts = all M >= 10 GeV [q,Q]
```

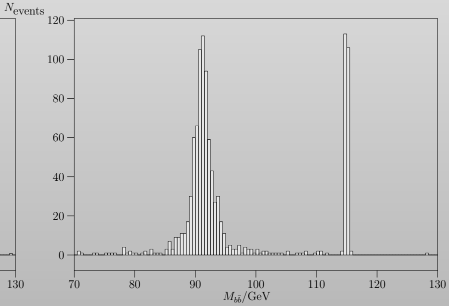
- integrate the cross sections, warming up the grids

```
integrate (nnbb, qqbb) { iterations = 12:20000 }
```

- create histograms for invariant masses and fill them with 10 fb^{-1}

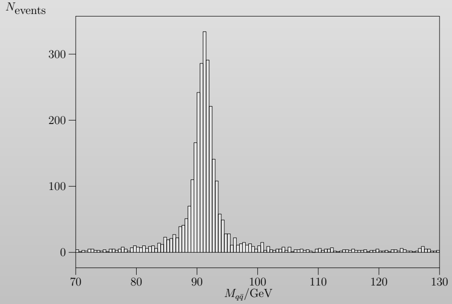
```
histogram m_invisible (70 GeV, 130 GeV, 0.5 GeV)
histogram m_bb (70 GeV, 130 GeV, 0.5 GeV)
luminosity = 10
simulate (nnbb) {
  analysis = record m_invisible (eval M [n,N]);
     record m_bb (eval M [b,B])
}
```





- once more for light quark jets

```
histogram m_jj (70 GeV, 130 GeV, 0.5 GeV)
simulate (qqbb) { analysis = record m_jj (eval M [q,Q]) }
compile_analysis { $out_file = "higgsstrahlung.dat" }
```



Simulate $u\bar{u} \to \bar{\tilde{u}}_1\tilde{u}_1 \to \bar{\tilde{u}}_1 u e^- \bar{\tilde{e}}^+$

- select the minimal supersymmetric standard model

```
model = MSSM
```

- full process

```
process full = u, U => SU1, u, e1, SE12
```

- two particle decays $\tilde{u}_1 \to u\chi_2^0$ and $\chi_2^0 \to \bar{\tilde{e}}^+ e^-$

```
process dec_su_q = su1 => u, neu2
process dec_neu_sl2 = neu2 => SE12, e1
```

- on-shell $u\bar{u} \to \bar{\tilde{u}}_1\tilde{u}_1$ production

```
process onshell = u,U => SU1, su1
```

- call O'Mega, compile and dynlink the matrix elements

```
compile
```

- read the SLHA parameters

```
?slha_read_decays = true
read_slha("sps1ap_decays.slha")
```

- integrate the decays: not many samples needed

```
integrate (dec_su_q, dec_neu_sl2) { iterations = 1:1000 }
```

- LHC in 2013

```
sqrts = 14000
beams = p, p => lhapdf
```

- very inclusive cuts

```
cuts = all Pt > 10 GeV [u]
```

- use more interations to adapt the grids for the full process

```
integrate (onshell) { iterations = 5:10000, 2:10000 }
integrate (full) { iterations = 10:10000, 5:20000 }
```

- allocate histograms

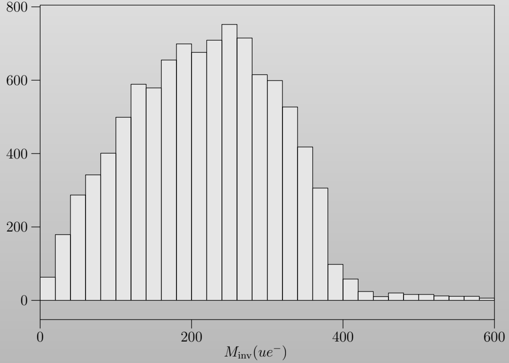
```
histogram inv_mass_full (0, 600, 20)
histogram inv_mass_off  (0, 600, 20)
histogram inv_mass_diag (0, 600, 20)
histogram inv_mass_iso  (0, 600, 20)
```

- WHIZARD integration log:

```
| 10000 calls, 49 channels, 10 dimensions, 20 bins, stratified = T
|=======================================================================|
| It   Calls Integral[fb]  Error[fb] Err[%]   Acc  Eff[%] Chi2 N[It] |
   1   10000 3.0581558E-01  5.70E-02 18.65  18.65*   1.25
   2   10000 3.1043792E-01  2.40E-02  7.74   7.74*   1.17
...
   6   10000 3.1822828E-01  3.44E-03  1.08   1.08*  10.61
   7   10000 3.1514664E-01  2.86E-03  0.91   0.91*  14.52
   8   10000 3.1580746E-01  3.02E-03  0.96   0.96   14.18
   9   10000 3.1467397E-01  2.75E-03  0.87   0.87*  12.90
  10   10000 3.1639558E-01  3.09E-03  0.98   0.98   12.85
|-----------------------------------------------------------------------|
  10 100000 3.1588868E-01  1.22E-03  0.39   1.22   12.85  0.53   10
|-----------------------------------------------------------------------|
  11   20000 3.2003341E-01  1.93E-03  0.60   0.85*  10.96
...
  15   20000 3.1792470E-01  7.81E-04  0.25   0.35*   7.68
|-----------------------------------------------------------------------|
  15 100000 3.1803887E-01  4.61E-04  0.14   0.46    7.68  0.36    5
```

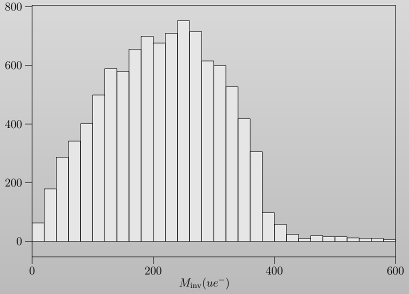
- generate 1000 events using the full matrix element and record the u, e^--invariant mass:

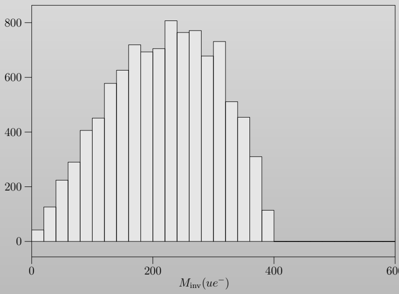
```
n_events = 10000
simulate (full) {
   analysis = record inv_mass_full (eval M [u,e1])
}
```



- the same using on-shell production and cascade decays with off diagonal spin correlations:

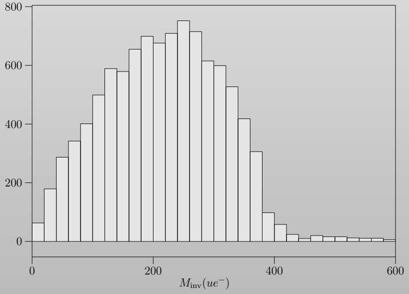
```
unstable su1 (dec_su_q)
unstable neu2 (dec_neu_sl2)
simulate (onshell) {
    analysis = record inv_mass_off (eval M [u,e1])
}
```





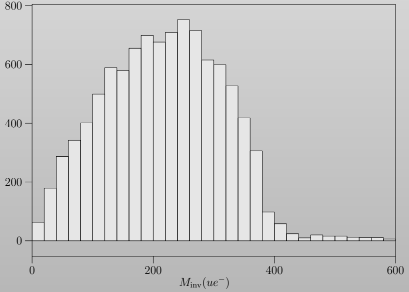
- keep only the diagonal spin correlations:

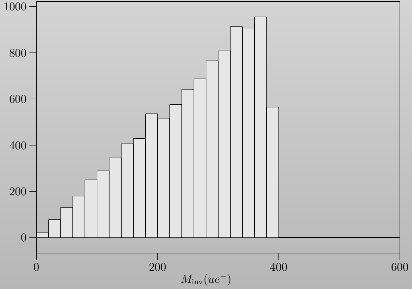
```
?diagonal_decay = true
unstable su1 (dec_su_q)
unstable neu2 (dec_neu_sl2)
simulate (onshell) {
   analysis = record inv_mass_diag (eval M [u,e1])
}
```



- isotropic decays (“it was 20 years ago . . . ”)

```
?isotropic_decay = true
unstable su1 (dec_su_q)
unstable neu2 (dec_neu_sl2)
simulate (onshell) {
   analysis = record inv_mass1_iso (eval M [u,e1])
}
write_analysis
compile_analysis { $out_file = "cascade_decays.dat" }
```





Simulate the W^- endpoint distribution

- select the Standard Model

```
model = SM
```

- set up the parton level processes $q\bar{q} \to \ell\nu_\ell j$

```
alias parton = u:U:d:D:g
alias jet = parton
alias lepton = e1:e2
alias neutrino = n1:N1:n2:N2
process enj = parton, parton => lepton, neutrino, jet
```

- call O'Mega, the Fortran compiler and the dynamic linker:

```
compile
```

- choose the LHC design energy

```
sqrts = 14 TeV
beams = p, p => lhapdf { $lhapdf_file = "cteq5l.LHgrid" }
```

- define reasonable phase space cuts

```
cuts = all Pt >= 10 GeV [jet:lepton]
```

- integrate the cross section in order to initialize the phase space grids for simulation

```
integrate (enj) { iterations = 5:20000 }
```

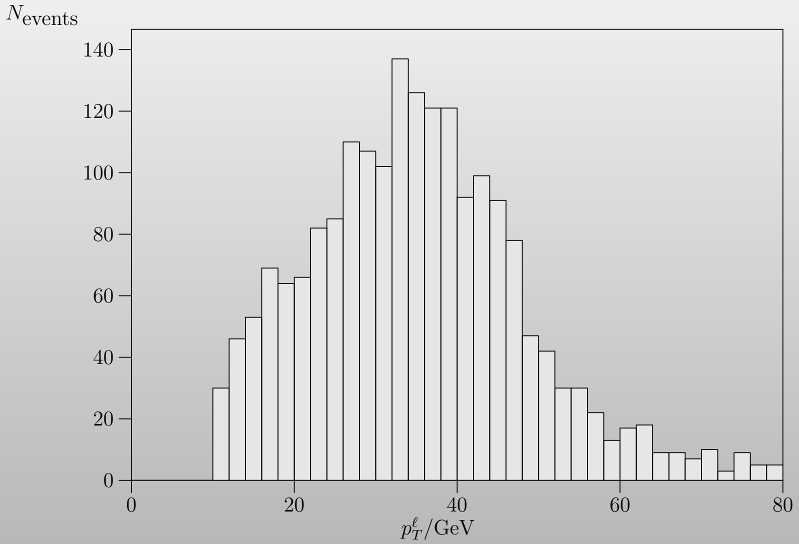
- allocate plots

```
$title = "$W$ Endpoint in $pp\to \ell\bar\nu j$"
$ylabel = "$N_{\textrm{events}}$"
$xlabel = "$p_T^\ell$/GeV"
histogram pt_lepton (0 GeV, 80 GeV, 2 GeV)
analysis =
  record pt_lepton
    (eval Pt [extract index 1 [sort by Pt [lepton]]])
```

- generate 1000 events and write the results

```
simulate (enj) { n_events = 1000 }
write_analysis
```

- Resulting plot



Remaining Challenges

- efficient and complete implementations of the Fortran 2003 standard
  - procedure pointers (required for dynamic linking) only correcty implemented by NAG and gfortran 4.5
- use more symmetries to reduce the code size
  - compiled code for multi jet cross sections at LHC can become larger than a giga[sic!]byte
- allow completely general vertex structures (MadGraph is working on this too)
- loops (proof-of-principle implementations by other groups exist, but completely general and fully automatic implementations are still science fiction . . .)
- get it from

  `http://projects.hepforge.org/whizard/`