The WHIZARD Event Generator

- Universal event generator for lepton and hadron colliders
- Modular package:
  - Phase space parameterization (resonances, collinear emission, Coulomb etc.)
  - O’Mega optimized matrix element generator (tree level, NLO external)
  - VAMP: adaptive multi-channel Monte Carlo integrator
  - CIRCE1/2: generator/simulation tool for lepton collider beam spectra
  - Modules for beam structure, parton shower, matching/merging, event formats, analysis, cascade decays, polarized initial/final states, [NLO subtractions] etc.
  - Interfaces to external packages for Feynman rules, hadronization, tau decays, event formats, analysis, jet clustering etc.
  - SINDARIN: free-format steering language for all inputs (!)
The WHIZARD Event Generator

- Universal event generator for lepton and hadron colliders

- Modular package:
  - **Phase space parameterization** (resonances, collinear emission, Coulomb etc.)
  - **O’Mega optimized matrix element generator** (tree level, NLO external)
  - **VAMP**: adaptive multi-channel Monte Carlo integrator
  - **CIRCE1/2**: generator/simulation tool for lepton collider beam spectra
  - Modules for **beam structure, parton shower, matching/merging, event formats, analysis, cascade decays, polarized initial/final states, [NLO subtractions]** etc.
  - Interfaces to external packages for **Feynman rules, hadronization, tau decays, event formats, analysis, jet clustering** etc.
  - **SINDARIN**: free-format steering language for all inputs (!)

---

**v1.0**  Project start ca. 1999 (parts early 90's): TESLA studies ➔ W, Z, Higgs (+ resp. decays)

**v1.20**  02/2002: optimized matrix elements (O’Mega)

**v1.25**  06/2003: first ever multi-leg implementation of the MSSM

**v1.50**  02/2006: QCD color flow formalism

**v1.95/97**  02/2010: NMSSM, UED, parton shower (alpha), development stop v1

**v2.0.0**  04/2010: OO overhaul (38 months), modern v2 version, faster matrix elements

**v2.1.0**  06/2012: FSR/ISR shower, SINDARIN, unit tests etc., cascade processes

**v2.2.0**  04/2014: 2nd OO overhaul (18 months)

**v2.2.5**  02/2015: production version, LCIO, NLO alpha, POWHEG alpha, top threshold

WHIZARD: Some (technical) facts

WHIZARD v2.2.5 (27.02.2015)  http://whizard.hepforge.org

<whizard@desy.de>

WHIZARD Team:  Wolfgang Kilian, Thorsten Ohl, JRR
Bijan Chokoufè/Marco Sekulla/Christian Weiss + 2 Master + 2 PhD (soon)
(some losses: C. Speckner [software engineering], F. Bach [ESA Space Defense], S. Schmidt [Philosophy])

Publication:  EPJC71 (2011) 1742  (and others for O’Mega, Interfaces, color flow formalism)
WHIZARD: Some (technical) facts

WHIZARD v2.2.5 (27.02.2015)  http://whizard.hepforge.org

<whizard@desy.de>

WHIZARD Team:  Wolfgang Kilian, Thorsten Ohl, JRR
Bijan Chokoufé/Marco Sekulla/Christian Weiss + 2 Master + 2 PhD (soon)
(some losses: C. Speckner [software engineering], F. Bach [ESA Space Defense], S. Schmidt [Philosophy])

Publication: EPJ C71 (2011) 1742 (and others for O’Mega, Interfaces, color flow formalism)
WHIZARD: Some (technical) facts

WHIZARD v2.2.5 (27.02.2015)  http://whizard.hepforge.org

<whizard@desy.de>

WHIZARD Team:  Wolfgang Kilian, Thorsten Ohl, JRR
Bijan Chokoufè/Marco Sekulla/Christian Weiss + 2 Master + 2 PhD (soon)
(some losses: C. Speckner [software engineering], F. Bach [ESA Space Defense], S. Schmidt [Philosophy])

Publication:  EPJ C71 (2011) 1742  (and others for O’Mega, Interfaces, color flow formalism)

support junior developers

2nd WHIZARD Workshop Würzburg, 03/2015
WHIZARD: Some (technical) facts

- **Programming Languages**: Fortran2003/2008 (gfortran $\geq 4.7.4$), OCaml $\geq 3.12.0$
- **Standard conformance to autotools**: libtool/autoconf/automake
- **Standard installation**: configure <FLAGS>, make, [make check], make install
- **Modern OO programming**: abstract modules, polymorphism, inheritance etc. etc.
- **Version control system** (subversion @ Hepforge), internal ticket system
- **Large self test suite, unit tests** [module tests], regression testing
- **Continuous integration system** (jenkins @ Siegen)
- **NEW**: ticketing system (JIRA @ DESY) for user support issues (questions, bug reports)
WHIZARD: Some (technical) facts

- Programming Languages: Fortran2003/2008 (gfortran ≥4.7.4), OCaml (≥3.12.0)
- Standard conformance to autotools: libtool/autoconf/automake
- Standard installation: configure <FLAGS>, make, [make check], make install
- Modern OO programming: abstract modules, polymorphism, inheritance etc. etc.
- Version control system (subversion @ Hepforge), internal ticket system
- Large self test suite, unit tests [module tests], regression testing
- Continous integration system (jenkins @ Siegen)
- NEW: ticketing system (JIRA @ DESY) for user support issues (questions, bug reports)
WHIZARD: Some (technical) facts

- Programming Languages: Fortran2003/2008 (gfortran ≥4.7.4), OCaml (≥3.12.0)
- Standard conformance to autotools: libtool/autoconf/automake
- Standard installation: configure <FLAGS>, make, [make check], make install
- Modern OO programming: abstract modules, polymorphism, inheritance etc. etc.
- Version control system (subversion @ Hepforge), internal ticket system
- Large self test suite, unit tests [module tests], regression testing
- Continuous integration system (jenkins @ Siegen)
- NEW: ticketing system (JIRA @ DESY) for user support issues (questions, bug reports)
General structure of SINDARIN input

model = SM

alias ll = “e-” :”e+” :”mu+” :”mu-“
alias nu = n1:N1:n2:N2:n3:N3
alias jet = u:U:d:D:s:S:g

process tth = e1, E1 => t, tbar, h
process tthfull =
   e1, E1 => ll, nu, ll, nu, b, bbar, jet, jet
process inclusive =
   e1, E1 => (Z, h) + (Z, Z) + (Wp, Wm)
process t_dec = t => E1, nubar, b

sqrts = 500 GeV
beams = e1, E1 => circe1 => ISR

cuts = all M > 10 GeV [jet, jet]

integrate (tthfull)
   { iterations = 15:500000, 5:1000000 }

n_events = 10000

unstable t (t_dec)

sample_format = lhef, stdhep, hepmc
sample = “mydata”
General structure of SINDARIN input

model = SM

alias ll = “e-””:”e+”:”mu+”:”mu-”
alias nu = n1:N1:n2:N2:n3:N3
alias jet = u:U:d:D:s:S:g

process tth = e1, E1 => t, tbar, h
process tthfull =
    e1, E1 => ll, nu, ll, nu, b, bbar, jet, jet
process inclusive =
    e1, E1 => (Z, h) + (Z, Z) + (Wp, Wm)
process t_dec = t => E1, nubar, b

sqrts = 500 GeV
beams = e1, E1 => circe1 => ISR

cuts = all M > 10 GeV [jet, jet]

integrate (tthfull)
    { iterations = 15:500000, 5:1000000 }

n_events = 10000

unstable t (t_dec)

sample_format = lhef, stdhep, hepmc
sample = “mydata”
General structure of SINDARIN input

model = SM

alias ll = "e-"":"e+"":"mu+"":"mu-"
alias nu = n1:N1:n2:N2:n3:N3
alias jet = u:U:d:D:s:S:g

process tth = e1, E1 => t, tbar, h
process tthfull =
   e1, E1 => ll, nu, ll, nu, b, bbar, jet, jet
process inclusive =
   e1, E1 => (Z, h) + (Z, Z) + (Wp, Wm)
process t_dec = t => E1, nubar, b

sqrts = 500 GeV
beams = e1, E1 => circe1 => ISR

cuts = all M > 10 GeV [jet, jet]

integrate (tthfull)
   { iterations = 15:500000, 5:1000000 }

n_events = 10000

unstable t (t_dec)

sample_format = lhef, stdhep, hepmc
sample = "mydata"
General structure of SINDARIN input

model = SM

alias ll = “e-”：“e+”：“mu+”：“mu-”
alias nu = n1:N1:n2:N2:n3:N3
alias jet = u:U:d:D:s:S:g

process tth = e1, E1 => t, tbar, h
process tthfull =
  e1, E1 => ll, nu, ll, nu, b, bbar, jet, jet
process inclusive =
  e1, E1 => (Z, h) + (Z, Z) + (Wp, Wm)
process t_dec = t => E1, nubar, b

sqrts = 500 GeV
beams = e1, E1 => circe1 => ISR
cuts = all M > 10 GeV [jet, jet]
integrate (tthfull) { iterations = 15:500000, 5:1000000 }
n_events = 10000
unstable t (t_dec)
sample_format = lhef, stdhep, hepmc

---

LCWS ’14, Belgrade, Simulation summary talk:
WHIZARD Task to implement LCIO format

WHIZARD v2.2.4, 02/2015:

sample_format = lcio
simulate (<process>)
Lepton Collider Beam Simulation

• Another demand: adapt GuineaPig beam spectra for WHIZARD v2
• For WHIZARD v1.95 simulations done by Lumilinker [T. Barklow]
• TESLA/SLC spectra were rather simple
• Fits with 6 or 7 parameters possible [CIRCE1]
• Beams not factorizable: \[ D_{B_1 B_2}(x_1, x_2) \neq D_{B_1}(x_1) \cdot D_{B_2}(x_2) \]
• No simple power law: \[ D_{B_1 B_2}(x_1, x_2) \neq x_1^{\alpha_1} (1 - x_1)^{\beta_1} x_2^{\alpha_2} (1 - x_2)^{\beta_2} \]
Lepton Collider Beam Simulation

- Another demand: adapt GuineaPig beam spectra for WHIZARD v2
- For WHIZARD v1.95 simulations done by Lumilinker [T. Barklow]
- TESLA/SLC spectra were rather simple
- Fits with 6 or 7 parameters possible [CIRCE1]
- Beams not factorizable: \( D_{B_1B_2}(x_1, x_2) \neq D_{B_1}(x_1) \cdot D_{B_2}(x_2) \)
- No simple power law: \( D_{B_1B_2}(x_1, x_2) \neq x_1^{\alpha_1}(1 - x_1)^{\beta_1} x_2^{\alpha_2}(1 - x_2)^{\beta_2} \)

Dalena/Esbjerg/Schulte [LCWS 2011]

Tails @ CLIC much more complicated (wakefields)
Lepton Collider Beam Simulation

- Another demand: adapt GuineaPig beam spectra for WHIZARD v2
- For WHIZARD v1.95 simulations done by Lumilinker [T. Barklow]
- TESLA/SLC spectra were rather simple
- Fits with 6 or 7 parameters possible [CIRCE1]
- Beams not factorizable: \( D_{B_1B_2}(x_1, x_2) \neq D_{B_1}(x_1) \cdot D_{B_2}(x_2) \)
- No simple power law: \( D_{B_1B_2}(x_1, x_2) \neq x_1^{\alpha_1}(1 - x_1)^{\beta_1} x_2^{\alpha_2}(1 - x_2)^{\beta_2} \)

CIRCE2 algorithm (WHIZARD 2.2.5, 02/15)

- Adapt 2D factorized variable width histogram to steep part of distribution
- Smooth correlated fluctuations with moderate Gaussian filter [suppresses artifacts from limited GuineaPig statistics]
- Smooth continuum/boundary bins separately [avoid artificial beam energy spread]

Tails @ CLIC much more complicated (wakefields)
1. Run Guinea-Pig++ with

```plaintext
do_lumi=7; num_lumi=100000000; num_lumi_eg=100000000; num_lumi_gg=100000000;

```
to produce `lumi.[eg][eg].out` with $(E_1, E_2)$ pairs.

[Large event numbers, as Guinea-Pig++ will produce only a small fraction!]

2. Run `circe2_tool.opt` with steering file

```plaintext
{ file="ilc500/beams.circe"
  { design="ILC" roots=500 bins=100 scale=250 # E in [0,1]
    { pid/1=electron pid/2=positron pol=0 # unpolarized e-/e+
      events="ilc500/lumi.ee.out" columns=2 # <= Guinea-Pig
      lumi = 1564.763360 # <= Guinea-Pig
      iterations = 10 # adapting bins
      smooth = 5 [0,1) [0,1) smooth = 5 [0,1) [1] } } }
```
to produce correlated beam description

3. Run WHIZARD with SINDARIN input:

```plaintext
beams = e1, E1 => circe2
$circe2_file = "ilc500.circe"
$circe2_design = "ILC"
?circe_polarized = false
```
1. Run **Guinea-Pig++ with**

   ```
   do_lumi=7; num_lumi=100000000; num_lumi_eg=100000000; num_lumi_gg=100000000;
   ```

   to produce `lumi_[eg]_[eg].out` with \((E_1, E_2)\) pairs.

   [Large event numbers, as Guinea-Pig++ will produce only a small fraction!]

2. Run **circe2_tool.opt with steering file**

   ```
   { file="ilc500/beams.circe" 
     { design="ILC" roots=500 bins=100 scale=250 # E in [0,1] 
       { pid/1=electron pid/2=positron pol=0 # unpolarized e-/e+ 
         events="ilc500/lumi.ee.out" columns=2 # <= Guinea-Pig 
         lumi = 1564.763360 # <= Guinea-Pig 
         iterations = 10 # adapting bins 
         smooth = 5 [0,1) [0,1) smooth = 5 [0,1) [1] } } } 
   ```

   to produce correlated beam description

3. Run **WHIZARD with SINDARIN input:**

   ```
   beams = e1, E1 => circe2
   $circe2_file = "ilc500.circe"
   $circe2_design = "ILC"
   ?circe_polarized = false
   ```

   polarized spectra on demand
Iterations of Beam Spectrum

(171,306 GuineaPig events in 10,000 bins)
Iterations of Beam Spectrum

- **iterations** = 0 and **smooth** = 0, 3, 5:

- **iterations** = 2 and **smooth** = 0, 3, 5:

- **iterations** = 4 and **smooth** = 0, 3, 5:
NLO Development in WHIZARD

- Need for precision predictions that match (sub-) percent experimental accuracy
- Scary challenge for the theory community [ok, we have some time still …]
- Mostly electroweak corrections, but also QCD and pure QED

**Binoth Les Houches Interface (BLHA): Workflow**

1. Process definition in SINDARIN (contract to One-Loop Program [OLP])
2. OLP generates code (Born/NLO interference), WHIZARD reads contract
3. NLO matrix element loaded into WHIZARD
NLO Development in WHIZARD

- Need for precision predictions that match (sub-) percent experimental accuracy
- Scary challenge for the theory community [ok, we have some time still …]
- Mostly electroweak corrections, but also QCD and pure QED

Binoth Les Houches Interface (BLHA): Workflow

1. Process definition in SINDARIN (contract to One-Loop Program [OLP])
2. OLP generates code (Born/NLO interference), WHIZARD reads contract
3. NLO matrix element loaded into WHIZARD

Working NLO interfaces to:

- GoSam [G. Cullen et al.]
- OpenLoops [J. Lindert et al.]

(first focus on QCD corrections)
NLO Development in WHIZARD

- Need for precision predictions that match (sub-) percent experimental accuracy
- Scary challenge for the theory community [ok, we have some time still …]
- Mostly electroweak corrections, but also QCD and pure QED

Binoth Les Houches Interface (BLHA): Workflow

1. Process definition in SINDARIN (contract to One-Loop Program [OLP])
2. OLP generates code (Born/NLO interference), WHIZARD reads contract
3. NLO matrix element loaded into WHIZARD

Working NLO interfaces to:

- GoSam [G. Cullen et al.]
- OpenLoops [J. Lindert et al.]

(First focus on QCD corrections)

WHIZARD v2.2.5 contains alpha version

QCD corrections (massless and massive emitters)

```plaintext
alpha_power = 2
alphas_power = 0

process eett = e1,E1 => t, tbar
   { nlo_calculation = "full" }
```
NLO Development in WHIZARD

- Need for precision predictions that match (sub-) percent experimental accuracy
- Scary challenge for the theory community [ok, we have some time still …]
- Mostly electroweak corrections, but also QCD and pure QED

Binoth Les Houches Interface (BLHA): Workflow

1. Process definition in SINDARIN (contract to One-Loop Program [OLP])
2. OLP generates code (Born/NLO interference), WHIZARD reads contract
3. NLO matrix element loaded into WHIZARD

Working NLO interfaces to:
- GoSam [G. Cullen et al.]
- OpenLoops [J. Lindert et al.]

(First focus on QCD corrections)

WHIZARD v2.2.5 contains alpha version

QCD corrections (massless and massive emitters)

```
alpha_power = 2
alphas_power = 0
process eett = e1,E1 => t, tbar
{ nlo_calculation = "full" }
```
FKS Subtraction \textit{(Frixione/Kunszt/Signer)}

Subtraction formalism to make real and virtual contributions separately finite

$$d\sigma^{\text{NLO}} = \int_{n+1} \left( d\sigma^R - d\sigma^S \right) + \int_{n+1} d\sigma^S + \int_n d\sigma^V$$

\text{finite}

\text{finite}
FKS Subtraction (Frixione/Kunszt/Signer)

Subtraction formalism to make real and virtual contributions separately finite

\[ d\sigma^{\text{NLO}} = \int_{n+1} \left( d\sigma^R - d\sigma^S \right) + \int_{n+1} d\sigma^S + \int_{n} d\sigma^V \]

Automated Subtraction algorithm:

- Find all singular pairs
  \[ \mathcal{I} = \{(1, 5), (1, 6), (2, 5), (2, 6), (5, 6)\} \]
- Partition phase space according to singular regions
  \[ 1 = \sum_{\alpha \in \mathcal{I}} S_\alpha(\Phi) \]
- Generate subtraction terms for singular regions
FKS Subtraction \textit{(Frixione/Kunszt/Signer)}

Subtraction formalism to make real and virtual contributions separately finite

\begin{equation}
\frac{d\sigma^{\text{NLO}}}{d^{d}R} = \int_{n+1} \left( d\sigma^{R} - d\sigma^{S} \right) + \int_{n+1} d\sigma^{S} + \int_{n} d\sigma^{V}
\end{equation}

Automated Subtraction algorithm:

\begin{itemize}
\item Find all singular pairs
\[ I = \{(1, 5), (1, 6), (2, 5), (2, 6), (5, 6)\} \]
\item Partition phase space according to singular regions
\[ 1 = \sum_{\alpha \in I} S_{\alpha}(\Phi) \]
\item Generate subtraction terms for singular regions
\end{itemize}

\begin{itemize}
\item \textbf{Soft subtraction} involves color-correlated matrix elements:
\[ B_{kl} \sim - \sum_{\text{color}} A^{(n)}(\mathcal{I}_k) \cdot \mathcal{Q}(\mathcal{I}_l) A^{(n)*}, \]
\end{itemize}

\begin{itemize}
\item \textbf{Collinear subtraction} involves spin-correlated matrix elements:
\[ B_{+-} \sim \text{Re} \left\{ \frac{\langle k_{em} k_{rad} \rangle}{[k_{em} k_{rad}]} \sum_{\text{color}} \mathcal{A}_{+}^{(n)} \mathcal{A}_{-}^{(n)*} \right\} \]
\end{itemize}
Examples and Validation

Simplest benchmark process:
\[ e^+ e^- \to q\bar{q} \quad \text{with} \quad \frac{(\sigma^{\text{NLO}} - \sigma^{\text{LO}})}{\sigma^{\text{LO}}} = \frac{\alpha_s}{\pi} \]

Plot for total cross section for fixed strong coupling constant

List of validated QCD NLO processes

- \[ e^+ e^- \to q\bar{q} \]
- \[ e^+ e^- \to q\bar{q}g \]
- \[ e^+ e^- \to \ell^+ \ell^- q\bar{q} \]
- \[ e^+ e^- \to \ell^+ \nu_\ell q\bar{q} \]
- \[ e^+ e^- \to t\bar{t} \]
- \[ e^+ e^- \to tW^- \bar{b} \]
- \[ e^+ e^- \to W^+ W^- b\bar{b} \]

Caveat: no fixed-order NLO event generation due to missing counter-event infrastructure

- Cross-checks with Madgraph5_aMC@NLO (except for ee \( \to WbWb \))
Examples and Validation

Simplest benchmark process:

\[ e^+ e^- \rightarrow q\bar{q} \quad \text{with} \quad (\sigma^{NLO} - \sigma^{LO}) / \sigma^{LO} = \alpha_s / \pi \]

Plot for total cross section for fixed strong coupling constant

List of validated QCD NLO processes

- \[ e^+ e^- \rightarrow q\bar{q} \]
- \[ e^+ e^- \rightarrow q\bar{q}g \]
- \[ e^+ e^- \rightarrow \ell^+ \ell^- q\bar{q} \]
- \[ e^+ e^- \rightarrow \ell^+ \nu_\ell q\bar{q} \]
- \[ e^+ e^- \rightarrow t\bar{t} \]
- \[ e^+ e^- \rightarrow tW^- b \]
- \[ e^+ e^- \rightarrow W^+ W^- b\bar{b} \]

Caveat: no fixed-order NLO event generation due to missing counter-event infrastructure

- Cross-checks with Madgraph5_aMC@NLO (except for \( ee \rightarrow WbWb \))

✦ First working infrastructure for QCD NLO in pp
✦ First attempts on electroweak corrections, interfacing the RECOLA code [Denner et al.]
POWHEG Matching in WHIZARD

- Soft gluon emission before hard emission generate large logs
- Perturbative $\alpha_s$:
  \[ |M_{\text{soft}}|^2 \sim \frac{1}{k_T^2} \rightarrow \log \frac{k_T^{\text{max}}}{k_T^{\text{min}}} \]
- Matrix element + parton shower has to take this into account
- POWHEG method: hardest emission first [Nason et al.]
POWHEG Matching in WHIZARD

- Soft gluon emission before hard emission generate large logs
- Perturbative $\alpha_s$: $|\mathcal{M}_{\text{soft}}|^2 \sim \frac{1}{k_T^2} \rightarrow \log \frac{k_T^{\text{max}}}{k_T^{\text{min}}}$
- Matrix element + parton shower has to take this into account
- POWHEG method: hardest emission first [Nason et al.]

Complete NLO events

$$\bar{B}(\Phi_n) = B(\Phi_n) + V(\Phi_n) + \int d\Phi_{\text{rad}} R(\Phi_{n+1})$$

POWHEG generate events according to the formula:

$$d\sigma = \bar{B}(\Phi_n) \left[ \Delta_{R}^{\text{NLO}}(k_{T}^{\text{min}}) + \Delta_{R}^{\text{NLO}}(k_{T}) \frac{R(\Phi_{n+1})}{B(\Phi_n)} d\Phi_{\text{rad}} \right]$$

Uses the modified Sudakov form factor:

$$\Delta_{R}^{\text{NLO}}(k_{T}) = \exp \left[ - \int d\Phi_{\text{rad}} \frac{R(\Phi_{n+1})}{B(\Phi_n)} \theta(k_{T}(\Phi_{n+1}) - k_{T}) \right]$$
POWHEG Matching in WHIZARD

- Soft gluon emission before hard emission generate large logs
- Perturbative $\alpha_s$: $|\mathcal{M}_\text{soft}|^2 \sim \frac{1}{k_T^2} \to \log \frac{k_T^{\max}}{k_T^{\min}}$
- Matrix element + parton shower has to take this into account
- POWHEG method: hardest emission first [Nason et al.]

• Complete NLO events

$$\overline{B}(\Phi_n) = B(\Phi_n) + V(\Phi_n) + \int d\Phi_{\text{rad}} R(\Phi_{n+1})$$

• POWHEG generate events according to the formula:

$$d\sigma = \overline{B}(\Phi_n) \left[ \Delta_{R}^{\text{NLO}}(k_T^{\min}) + \Delta_{R}^{\text{NLO}}(k_T) \frac{R(\Phi_{n+1})}{B(\Phi_n)} d\Phi_{\text{rad}} \right]$$

• Uses the modified Sudakov form factor:

$$\Delta_{R}^{\text{NLO}}(k_T) = \exp \left[ -\int d\Phi_{\text{rad}} \frac{R(\Phi_{n+1})}{B(\Phi_n)} \theta(k_T(\Phi_{n+1}) - k_T) \right]$$

• Hardest emission: $k_T^{\max}$; shower with imposing a veto:
- $\overline{B} < 0$ if virtual and real terms larger than Born: shouldn’t happen in perturbative regions
- Reweighting such that $\overline{B} > 0$ for all events
- POWHEG: Positive Weight Hardest Emission Generator now implemented in WHIZARD
POWHEG Matching in $e^+e^-$ to dijets

- Comparison of LO+Pythia8 and POWHEG+Pythia8 for various quantities.
- Plots show ratios and distributions for $1-T$ and Major/Minor axes.

WhIZARD+Omega/GoSam results are highlighted in each plot.
Top Threshold at lepton colliders

ILC top threshold scan best-known method to measure top quark mass, $\Delta M \sim 100$ MeV

Heavy quark production at lepton colliders

Threshold region (quantitatively)
Implement resummed threshold effects as effective tab vertex [form factor] in WHIZARD

\[ G^{v,a}(0, p_t, E + i\Gamma_t, \nu) \] from TOPPIK code [Jezabek/Teubner], included in WHIZARD
Top Threshold in WHIZARD

- Implement resummed threshold effects as effective tab vertex [form factor] in WHIZARD
- \( G^{v,a}(0, p_t, E + i\Gamma_t, \nu) \) from TOPPIK code [Jezabek/Teubner], included in WHIZARD

\[
R^{\gamma,Z}(s) = F^{v}(s)R^{v}(s) + F^{a}(s)R^{a}(s)
\]

s-wave: LL+NLL  \quad p-wave:\sim v^2:NNLL

**BUT: differentially** \( p \)-wave at NLL!

- Default parameters:
  \[
  M^{1S} = 172 \text{ GeV}, \quad \Gamma_t = 1.54 \text{ GeV},
  \]
  \[
  \alpha_s(M_Z) = 0.118
  \]

Threshold/Continuum Matching: WIP
Top Threshold in WHIZARD

- Implement resummed threshold effects as effective tab vertex [form factor] in WHIZARD
- $G^{v,a}(0, p_t, E + i\Gamma_t, \nu)$ from TOPPIK code [Jezabek/Teubner], included in WHIZARD

Threshold/Continuum Matching: WIP

$R_{\gamma,Z}^{\gamma,Z}(s) = F_v(s)R_v(s) + F_a(s)R_a(s)$

**s-wave: LL+NLL**

**p-wave $\sim v^2$:NNLL**

**BUT: differentially p-wave at NLL!**

**Default parameters:**

$M_1^{1S} = 172 \text{ GeV}$, $\Gamma_t = 1.54 \text{ GeV}$,

$\alpha_s(M_Z) = 0.118$
Top Threshold in WHIZARD

- Implement resummed threshold effects as effective tab vertex [form factor] in WHIZARD
- $G^{v,a}(0, p_t, E + i\Gamma_t, \nu)$ from TOPPIK code [Jezabek/Teubner], included in WHIZARD

\[ R^{\gamma,Z}(s) = \underbrace{F^{v}(s)R^{v}(s)}_{\text{s-wave: LL+NLL}} \underbrace{+ F^{a}(s)R^{a}(s)}_{\text{p-wave} \sim v^2: \text{NNLL}} \]

BUT: differentially p-wave at NLL!

- Default parameters:
  \[ M^{1S} = 172 \text{ GeV}, \quad \Gamma_t = 1.54 \text{ GeV}, \]
  \[ \alpha_s(M_Z) = 0.118 \]

Threshold/Continuum Matching: WIP
Spin Correlation and Polarization in Cascades

Cascade decay, factorize production and decay

simulate (fullproc)

simulate (casc)

?diagonal_decay = true

?isotropic_decay = true
Spin Correlation and Polarization in Cascades

Cascade decay, factorize production and decay

simulate (fullproc)

simulate (casc)

?diagonal_decay = true

?isotropic_decay = true

NEW: possibility to select specific helicity in decays!

unstable "W+" { decay_helicity = 0 }
Projects, Plans, Performance and all that

- **O'Mega Virtual Machine (OVM):** ME via bytecode interpreter than compiled code ✓
- Parton shower: LO merging (MLM ✓), NLO matching
- QED shower (FSR)
- QED shower (ISR): exclusive part of ISR spectrum
- pT spectrum of ISR radiation
- **automated massless/massive QCD NLO corrections:** FS ✓ / Initial state in preparation
  → WHIZARD 3.0
- QED/electroweak NLO automation: longer time scale
- complete NLL NRQCD top threshold/NLO continuum matching; extension to ttH
- **POWHEG matching** implemented ✓; maybe also MC@NLO or Nagy-Soper matching
- Monte Carlo over helicities and colors
- Modified algorithm for multi-leg (tree) matrix elements: includes high-color flow amplitudes, QCD/EW coupling orders, general Lorentz structures
- Automatic generation of decays (and calculation of decay widths)
- New syntax for nested decay chains

```plaintext
process = e1, E1 => (t => (Wp => E2, nu2), b), tbar
```
Conclusions & Outlook

- WHIZARD 2.2 excellent tool for Linear Collider Physics
- [WHIZARD 2.2 excellent tool for LHC Physics]
- Great effort on the demands for mass production for LCs
- Beamspectra, LCIO, LC top threshold
- Main focus in physics: NLO automation $\rightarrow$ WHIZARD 3.0
- Performance: many developments to come
- Tell us what is missing, insufficient, annoying, desirable
Conclusions & Outlook

- WHIZARD 2.2 excellent tool for Linear Collider Physics
- WHIZARD 2.2 excellent tool for LHC Physics
- Great effort on the demands for mass production for LCs
- Beamspectra, LCIO, LC top threshold
- Main focus in physics: NLO automation $\rightarrow$ WHIZARD 3.0
- Performance: many developments to come
- Tell us what is missing, insufficient, annoying, desirable

  even if it is in a LCWS summary talk $\implies$ Challenge accepted!
Conclusions & Outlook

- WHIZARD 2.2 excellent tool for Linear Collider Physics
- Great effort on the demands for mass production for LCs
- Beamspectra, LCIO, LC top threshold
- Main focus in physics: NLO automation $\rightarrow$ WHIZARD 3.0
- Performance: many developments to come
- Tell us what is missing, insufficient, annoying, desirable

even if it is in a LCWS summary talk $\implies$ Challenge accepted!

(Personal) Memory to LCWS 2013: 金閣寺
ありがとうございます。