Production Experience ILD at ILC - Generating the full SM

(largely based on PoS(ICHEP2020)903)

Mikael Berggren[1]

[1]DESY, Hamburg

Generator group meeting, Zoom, Feb. 28 '24

Outline

1. Why: MC requirements for linear colliders
2. How: Generating the full SM
   - Generating beam properties
3. In action: ILC 250 GeV Generation production
4. What next ?
5. Conclusions

MC requirements for linear colliders

- Future LCs aim for extremely high precision measurements.
  - ⇒ Need excellent detector, well controlled machine conditions
  - But also the best possible estimate of backgrounds.
- So: MC statistics or lacking channels must not be a major source of systematic errors ⇒
  - All SM channels yielding at least a few events under the full lifetime of the projects need to be generated, with statistics largely exceeding that of the real data.
  - Also machine conditions need to be accurately taken into account.
- In addition: at an LC ALL events are interesting, and often fully reconstructed. More like a B-factory than LHC!
- This endeavour has been organised as a common effort between ILD and SiD at ILC and CLICdp at CLIC. The work is done within the generator group, LCGG. The common group has been active under both the GDE, the LCC and now under the IDT.

MC requirements for linear colliders

- Future LCs aim for extremely high precision measurements.
  - ⇒ Need excellent detector, well controlled machine conditions
  - But also the best possible estimate of backgrounds.
- So: MC statistics or lacking channels must not be a major source of systematic errors ⇒
  - All SM channels yielding at least a few events under the full lifetime of the projects need to be generated, with statistics largely exceeding that of the real data.
  - Also machine conditions need to be accurately taken into account.
- In addition: at an LC ALL events are interesting, and often fully reconstructed. More like a B-factory than LHC!
- This endeavour has been organised as a common effort between ILD and SiD at ILC and CLICdp at CLIC. The work is done within the generator group, LCGG. The common group has been active under both the GDE, the LCC and now under the IDT.

MC requirements for linear colliders

- Future LCs aim for extremely high precision measurements.
  - $\Rightarrow$ Need excellent detector, well controlled machine conditions
  - But also the best possible estimate of backgrounds.
- So: MC statistics or lacking channels must not be a major source of systematic errors $\Rightarrow$
  - All SM channels yielding at least a few events under the full lifetime of the projects need to be generated, with statistics largely exceeding that of the real data.
  - Also machine conditions need to be accurately taken into account.
- In addition: at an LC ALL events are interesting, and often fully reconstructed. More like a B-factory than LHC!
- This endeavour has been organised as a common effort between ILD and SiD at ILC and CLICdp at CLIC. The work is done within the generator group, LCGG. The common group has been active under both the GDE, the LCC and now under the IDT.

MC requirements for linear colliders

- Future LCs aim for extremely high precision measurements.
  - $\Rightarrow$ Need excellent detector, well controlled machine conditions
  - But also the best possible estimate of backgrounds.
- So: MC statistics or lacking channels must not be a major source of systematic errors $\Rightarrow$
  - All SM channels yielding at least a few events under the full lifetime of the projects need to be generated, with statistics largely exceeding that of the real data.
  - Also machine conditions need to be accurately taken into account.
- In addition: at an LC ALL events are interesting, and often fully reconstructed. More like a B-factory than LHC!
- This endeavour has been organised as a common effort between ILD and SiD at ILC and CLICdp at CLIC. The work is done within the generator group, LCGG. The common group has been active under both the GDE, the LCC and now under the IDT.

Main generator: Whizard

- Whizard remains the generator of choice for e^+e^-.
- Full matrix-element evaluation. Only at tree-level but:
  - Can do $2 \to 8$ processes.
  - Polarised beams.
  - Full helicity treatment.
  - Full colour flow, passed from the hard interaction to the P.S. code.
  - Can handle beam-spectrum, using Circe2.
- ... which is as important as NLO for e^+e^- !
- The subsequent parton-shower and hadronisation is done by PYTHIA6.4.
  - LCGG has tuned hadronisation using input from OPAL at LEPII.
- The process-definition given in the Whizard steering file (aka the *sindarin*) is also the driver for the scripts that organises the production: One ring to rule them all.
- Use powerful grouping and aliasing capabilities of sindarin to assure that no processes are over-looked.

Main generator: Whizard

- Whizard remains the generator of choice for e^+e^-.
- Full matrix-element evaluation. Only at tree-level but:
  - Can do $2 \to 8$ processes.
  - Polarised beams.
  - Full helicity treatment.
  - Full colour flow, passed from the hard interaction to the P.S. code.
  - Can handle beam-spectrum, using Circe2.
- ... which is as important as NLO for e^+e^- !
- The subsequent parton-shower and hadronisation is done by PYTHIA6.4.
  - LCGG has tuned hadronisation using input from OPAL at LEPII.
- The process-definition given in the Whizard steering file (aka the *sindarin*) is also the driver for the scripts that organises the production: One ring to rule them all.
- Use powerful grouping and aliasing capabilities of sindarin to assure that no processes are over-looked.

Main generator: Whizard

- Whizard remains the generator of choice for e^+e^-.
- Full matrix-element evaluation. Only at tree-level but:
  - Can do $2 \to 8$ processes.
  - Polarised beams.
  - Full helicity treatment.
  - Full colour flow, passed from the hard interaction to the P.S. code.
  - Can handle beam-spectrum, using Circe2.
- ... which is as important as NLO for e^+e^- !
- The subsequent parton-shower and hadronisation is done by PYTHIA6.4.
  - LCGG has tuned hadronisation using input from OPAL at LEPII.
- The process-definition given in the Whizard steering file (aka the *sindarin*) is also the driver for the scripts that organises the production: One ring to rule them all.
- Use powerful grouping and aliasing capabilities of sindarin to assure that no processes are over-looked.

Main generator: Whizard

- Whizard remains the generator of choice for e^+e^-.
- Full matrix-element evaluation. Only at tree-level but:
  - Can do $2 \rightarrow 8$ processes.
  - Polarised beams.
  - Full helicity treatment.
  - Full colour flow, passed from the hard interaction to the P.S. code.
  - Can handle beam-spectrum, using Circe2.
- ... which is as important as NLO for e^+e^- !
- The subsequent parton-shower and hadronisation is done by PYTHIA6.4.
  - LCGG has tuned hadronisation using input from OPAL at LEPII.
- The process-definition given in the Whizard steering file (aka the *sindarin*) is also the driver for the scripts that organises the production: One ring to rule them all.
- Use powerful grouping and aliasing capabilities of sindarin to assure that no processes are over-looked.

Main generator: Whizard

- Whizard remains the generator of choice for e^+e^-.
- Full matrix-element evaluation. Only at tree-level but:
  - Can do $2 \rightarrow 8$ processes.
  - Polarised beams.
  - Full helicity treatment.
  - Full colour flow, passed from the hard interaction to the P.S. code.
  - Can handle beam-spectrum, using Circe2.
- ... which is as important as NLO for e^+e^- !
- The subsequent parton-shower and hadronisation is done by PYTHIA6.4.
  - LCGG has tuned hadronisation using input from OPAL at LEPII.
- The process-definition given in the Whizard steering file (aka the *sindarin*) is also the driver for the scripts that organises the production: One ring to rule them all.
- Use powerful grouping and aliasing capabilities of sindarin to assure that no processes are over-looked.

Process classification

The classes

1. Initial state
   - ee, eγ or $\gamma\gamma$
2. Final state multiplicity
   - Number of fermions (0 to 8)
3. Final state flavours for up to 5 fermions
   - Flavour-grouping: W or Z, or ambiguous
4. Final state lepton/hadron mix
   - leptonic, hadronic, semi-leptonic (+ neutrino only, for Z-leptonic)
5. Beam-polarisation/ γ nature
   - LR, RL, RR, LL (100% always implied)/ W (EPA), B(real)

- Special Considerations
  - Eg. 4f with $|L_e|$=2 $\Rightarrow$ dominated by single W or single Z (t-channel !)

Process classification

The classes

1. Initial state
   - ee, eγ or $\gamma\gamma$
2. Final state multiplicity
   - Number of fermions (0 to 8)
3. Final state flavours for up to 5 fermions
   - Flavour-grouping: W or Z, or ambiguous
4. Final state lepton/hadron mix
   - leptonic, hadronic, semi-leptonic (+ neutrino only, for Z-leptonic)
5. Beam-polarisation/ γ nature
   - LR, RL, RR, LL (100% always implied)/ W (EPA), B(real)

- Special Considerations
  - Eg. 4f with $|L_e|$=2 $\Rightarrow$ dominated by single W or single Z (t-channel !)

Process classification

The classes

1. Initial state
   - ee, eγ or $\gamma\gamma$
2. Final state multiplicity
   - Number of fermions (0 to 8)
3. Final state flavours for up to 5 fermions
   - Flavour-grouping: W or Z, or ambiguous
4. Final state lepton/hadron mix
   - leptonic, hadronic, semi-leptonic (+ neutrino only, for Z-leptonic)
5. Beam-polarisation/ γ nature
   - LR, RL, RR, LL (100% always implied)/ W (EPA), B(real)

- Special Considerations
  - Eg. 4f with $|L_e|$=2 $\Rightarrow$ dominated by single W or single Z (t-channel !)

Process classification

The classes

1. Initial state
   - ee, eγ or $\gamma\gamma$
2. Final state multiplicity
   - Number of fermions (0 to 8)
3. Final state flavours for up to 5 fermions
   - Flavour-grouping: W or Z, or ambiguous
4. Final state lepton/hadron mix
   - leptonic, hadronic, semi-leptonic (+ neutrino only, for Z-leptonic)
5. Beam-polarisation/ γ nature
   - LR, RL, RR, LL (100% always implied)/ W (EPA), B(real)

- Special Considerations
  - Eg. 4f with $|L_e|$=2 $\Rightarrow$ dominated by single W or single Z (t-channel !)

Process classification

The classes

1. Initial state
   - ee, eγ or $\gamma\gamma$
2. Final state multiplicity
   - Number of fermions (0 to 8)
3. Final state flavours for up to 5 fermions
   - Flavour-grouping: W or Z, or ambiguous
4. Final state lepton/hadron mix
   - leptonic, hadronic, semi-leptonic (+ neutrino only, for Z-leptonic)
5. Beam-polarisation/ γ nature
   - LR, RL, RR, LL (100% always implied)/ W (EPA), B(real)

- Special Considerations
  - Eg. 4f with $|L_e|$=2 $\Rightarrow$ dominated by single W or single Z (t-channel !)

Process classification

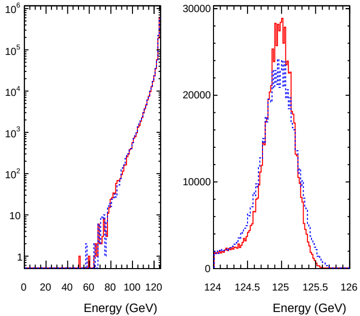
The classes

1. Initial state
   - ee, eγ or $\gamma\gamma$
2. Final state multiplicity
   - Number of fermions (0 to 8)
3. Final state flavours for up to 5 fermions
   - Flavour-grouping: W or Z, or ambiguous
4. Final state lepton/hadron mix
   - leptonic, hadronic, semi-leptonic (+ neutrino only, for Z-leptonic)
5. Beam-polarisation/ γ nature
   - LR, RL, RR, LL (100% always implied)/ W (EPA), B(real)

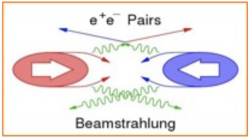
- Special Considerations
  - Eg. 4f with $|L_e|$=2 $\Rightarrow$ dominated by single W or single Z (t-channel !)

Beam effects

- Beam-spectrum.
  1. Incoming beam-spread
  2. But also: very strongly focused beams ⇒ Beam-beam interactions

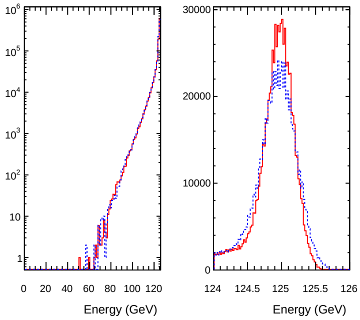


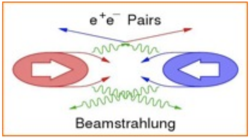
- Photons
  1. How many photons?
  2. Are they virtual or real?
- Need beam-beam interaction simulation input.
- Simulate interaction region: GuineaPig [CERN-PS-99-014-LP]. Gives:
  - Beam-spectrum for electrons and positrons independently
  - Amount and spectrum of real photons
  - Distribution of interaction point



Beam effects

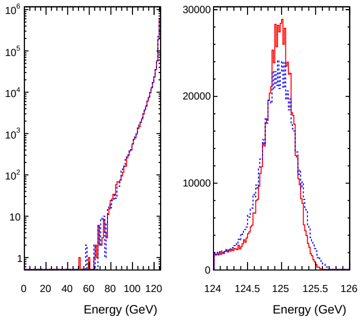
- Beam-spectrum.
  1. Incoming beam-spread
  2. But also: very strongly focused beams ⇒ Beam-beam interactions
- Photons
  1. How many photons?
  2. Are they virtual or real?
- Need beam-beam interaction simulation input.
- Simulate interaction region: GuineaPig [CERN-PS-99-014-LP]. Gives:
  - Beam-spectrum for electrons and positrons independently
  - Amount and spectrum of real photons
  - Distribution of interaction point

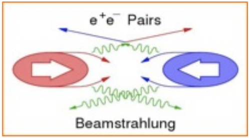




Beam effects

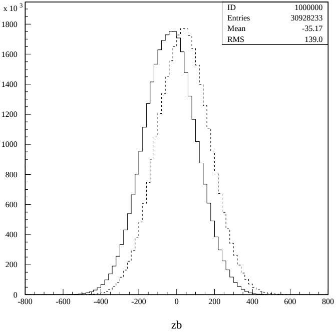
- Beam-spectrum.
  1. Incoming beam-spread
  2. But also: very strongly focused beams ⇒ Beam-beam interactions
- Photons
  1. How many photons?
  2. Are they virtual or real?
- Need beam-beam interaction simulation input.
- Simulate interaction region: GuineaPig [CERN-PS-99-014-LP]. Gives:
  - Beam-spectrum for electrons and positrons independently
  - Amount and spectrum of real photons
  - Distribution of interaction point

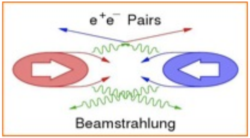




Beam effects

- Beam-spectrum.
  1. Incoming beam-spread
  2. But also: very strongly focused beams ⇒ Beam-beam interactions
- Photons
  1. How many photons?
  2. Are they virtual or real?
- Need beam-beam interaction simulation input.
- Simulate interaction region: GuineaPig [CERN-PS-99-014-LP]. Gives:
  - Beam-spectrum for electrons and positrons independently
  - Amount and spectrum of real photons
  - Distribution of interaction point

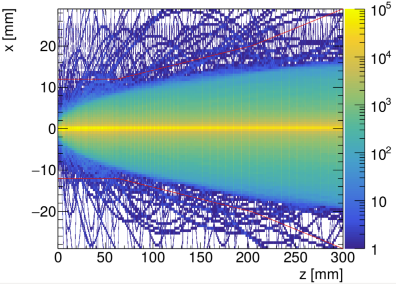




Spurious interactions ("pile-up")

Two types:

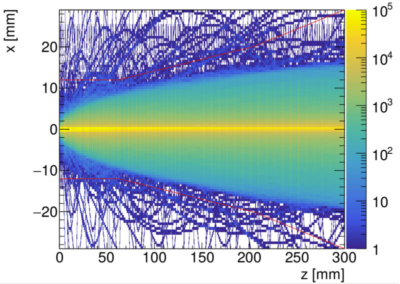
- Pair-background: Pair-creation of photons in the beam by the strong fields. GuineaPig can generate the full activity during a beam-crossing (a "BX").
- low-$p_\perp$ hadrons, ie. $\gamma^{(*)}\gamma^{(*)}$ interaction with small $M_{\gamma\gamma}$ and multiplicity. NB: only $\mathcal{O}(1)$/BX !
  - ME can't do this, and PYTHIA is good down to $M_{\gamma\gamma} \sim 2$ GeV.
  - Below: fit to data - Custom generator developed by LCGG.

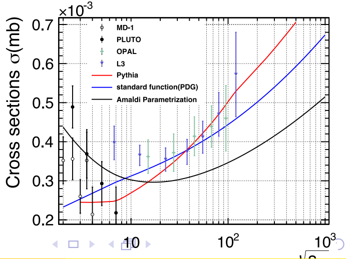


Spurious interactions ("pile-up")

Two types:

- Pair-background: Pair-creation of photons in the beam by the strong fields. GuineaPig can generate the full activity during a beam-crossing (a "BX").
- low-$p_\perp$ hadrons, ie. $\gamma^{(*)}\gamma^{(*)}$ interaction with small $M_{\gamma\gamma}$ and multiplicity. NB: only $\mathcal{O}(1)$/BX !
    - ME can't do this, and PYTHIA is good down to $M_{\gamma\gamma} \sim 2$ GeV.
    - Below: fit to data - Custom generator developed by LCGG.





Spurious interactions (“pile-up”)

- These backgrounds need to be passed on to simulation, but in a different mode.
- Eg. can’t simulate $\sim 10^5$ pairs on each physics event.
- Actually, can’t generate that either: time for 1 BX 5-10 minutes
- Find the few tracks that do hit the tracking (< 100/BX). Do $\sim$ 100000 BXes, and pick a random one from the pool to overlay to each physics event.
  - Done using the fast detector simulation code SGV, which faithfully evaluates detector acceptance.
- Also, use some ($\mathcal{O}(100)$) BXes to simulate pairs hitting the BeamCal, to build a map of the background, to be used in the BeamCal simulation.
- Similar for low-$p_\perp$ hadrons, but here also the number per BX is random, and their production point.

Spurious interactions ("pile-up")

- These backgrounds need to be passed on to simulation, but in a different mode.
- Eg. can't simulate $\sim 10^5$ pairs on each physics event.
- Actually, can't generate that either: time for 1 BX 5-10 minutes
- Find the few tracks that do hit the tracking ($<$ 100/BX). Do $\sim$ 100000 BXes, and pick a random one from the pool to overlay to each physics event.
  - Done using the fast detector simulation code SGV, which faithfully evaluates detector acceptance.
- Also, use some ($\mathcal{O}(100)$) BXes to simulate pairs hitting the BeamCal, to build a map of the background, to be used in the BeamCal simulation.
- Similar for low-$p_\perp$ hadrons, but here also the number per BX is random, and their production point.

Spurious interactions ("pile-up")

- These backgrounds need to be passed on to simulation, but in a different mode.
- Eg. can't simulate $\sim 10^5$ pairs on each physics event.
- Actually, can't generate that either: time for 1 BX 5-10 minutes
- Find the few tracks that do hit the tracking ($<$ 100/BX). Do $\sim$ 100000 BXes, and pick a random one from the pool to overlay to each physics event.
  - Done using the fast detector simulation code SGV, which faithfully evaluates detector acceptance.
- Also, use some ($\mathcal{O}(100)$) BXes to simulate pairs hitting the BeamCal, to build a map of the background, to be used in the BeamCal simulation.
- Similar for low-$p_\perp$ hadrons, but here also the number per BX is random, and their production point.

Spurious interactions ("pile-up")

- These backgrounds need to be passed on to simulation, but in a different mode.
- Eg. can't simulate $\sim 10^5$ pairs on each physics event.
- Actually, can't generate that either: time for 1 BX 5-10 minutes
- Find the few tracks that do hit the tracking (< 100/BX). Do $\sim$ 100000 BXes, and pick a random one from the pool to overlay to each physics event.
  - Done using the fast detector simulation code SGV, which faithfully evaluates detector acceptance.
- Also, use some ($\mathcal{O}(100)$) BXes to simulate pairs hitting the BeamCal, to build a map of the background, to be used in the BeamCal simulation.
- Similar for low-$p_\perp$ hadrons, but here also the number per BX is random, and their production point.

ILC 250 GeV Generation production

Currently, ILC is doing a new full production at 250 GeV, 10 times larger than the one done for the DBD/IDR, and about twice the expected real data-set. NB: Also includes the Higgs signal.

proposal for statistics of 250 GeV generators

process\pol.	eL.pR	eR.pL	eL.pL	eR.pR
2f_l, 2f_h	5 ab^{-1}	5 ab^{-1}	1 ab^{-1}	1 ab^{-1}
all 4f				
all 6f	10K	10K	10K	10K
2f_bhabhag	1 ab^{-1}	1 ab^{-1}	1 ab^{-1}	1 ab^{-1}
h->inclusive	1 ab^{-1}	1 ab^{-1}	1 ab^{-1}	1 ab^{-1}
h->each mode (5x9 channels)	100K	100K	10K	10K

most of the irreducible background will then have x10 more than expectation at ILC250

aa_2f, aa_4f: 1 ab^{-1} each initial state

Steps for ILC 250 GeV production: Beams

Beam-conditions etc.:

- Beam background with GuineaPig, 100000 BXes
    - Pair background
        - Need to create files with real tracks
        - One event with 1 BX
        - SGV is used to do this.
    - Beam-spectrum and Circe2 parametrisation.
    - Beam-spot size and position.
    - Input for BeamCal background maps.
- aa_lowpt for “pile-up”
    - Events to overlay.
    - Average number per BX evaluated.

Steps for ILC 250 GeV Generation: Setup & integration

- Whizard process definition is parsed to build a directory-tree structure - one unique directory per process.
- Process-specific code is generated and compiled (interactive)
- A "pre-integration" of all channels performed, to flag zero cross-section ones (interactive).
- Full integration of all channels, with error goal 0.1 % submitted (local batch-farm under Condor). Has already been done for $\leq$ 5f: over-night for $\leq$ 5f.
- Pilot generation of all channels, (1k events/channel), to evaluate CPU time and storage needed.
- For $\leq$ 5f: CPU $\sim$ 20 000 h - Disk $\sim$ 10 TB : Much, but not too much...

Steps for ILC 250 GeV Generation: Setup & integration

- Whizard process definition is parsed to build a directory-tree structure - one unique directory per process.
- Process-specific code is generated and compiled (interactive)
- A “pre-integration” of all channels performed, to flag zero cross-section ones (interactive).
- Full integration of all channels, with error goal 0.1 % submitted (local batch-farm under Condor). Has already been done for $\leq$ 5f: over-night for $\leq$ 5f.
- Pilot generation of all channels, (1k events/channel), to evaluate CPU time and storage needed.
- For $\leq$ 5f: CPU $\sim$ 20 000 h - Disk $\sim$ 10 TB : Much, but not too much...

Steps for ILC 250 GeV production: Generation

- In most cases: one process per generation job.
- Some processes alone represent $\sim$ billion events. These must be split in several jobs.
- Disk-space might be an issue, but here the switch from un-compressed stdhep output (DBD) to compressed LCIO output helps.
- At the end of each job, the events, metadata, and input+logfile tarballs are uploaded to the grid, where the simulation and reconstruction system under DIRAC kick in.
- At this point, disk can be freed on the batch cluster.
- Using $\mathcal{O}(100)$ cores, this generation is done in 1 to 2 weeks.

Steps for ILC 250 GeV production: Generation

- In most cases: one process per generation job.
- Some processes alone represent $\sim$ billion events. These must be split in several jobs.
- Disk-space might be an issue, but here the switch from un-compressed stdhep output (DBD) to compressed LCIO output helps.
- At the end of each job, the events, metadata, and input+logfile tarballs are uploaded to the grid, where the simulation and reconstruction system under DIRAC kick in.
- At this point, disk can be freed on the batch cluster.
- Using $\mathcal{O}(100)$ cores, this generation is done in 1 to 2 weeks.

ILC 250 GeV production:Documentation details

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam-polarisation, and cross-section of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, ...
  - Browasble on the Web and uploaded to the Grid.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,...:
  - On the Web, in full (W.I.P ... - exists for previous production)
  - In tar files - parallel to generated files, also stored on the Grid.

ILC 250 GeV production:Documentation details

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam-polarisation, and cross-section of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, ...
  - Browasble on the Web and uploaded to the Grid.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,...:
  - On the Web, in full (W.I.P ... - exists for previous production)
  - In tar files - parallel to generated files, also stored on the Grid.

ILC 250 GeV production:Documentation details

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam-polarisation, and cross-section of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, ...
  - Browasble on the Web and uploaded to the Grid.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,...:
  - On the Web, in full (W.I.P ... - exists for previous production)
  - In tar files - parallel to generated files, also stored on the Grid.

ILC 250 GeV production:Documentation details

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam-polarisation, and cross-section of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, ...
  - Browasble on the Web and uploaded to the Grid.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,...:
  - On the Web, in full (W.I.P ... - exists for previous production)
  - In tar files - parallel to generated files, also stored on the Grid.

ILC 250 GeV production: Generation status (OLD numbers...)

- Done on a batch-farm.
- For $\leq$ 5f : 116 channels. 96 virtual γ induced channels, deferred because of remaining Whizard issues details.
- As of today, 104 channels are done, producing 2.7 billion events in 15788 LCIO files details, occupying 5.4 TB. This used 7233 CPU hours, obtained in $\sim$ 10 days.
- In most cases: one channel = one generation job, but in some processes alone represent $\sim$ billion events: split in several jobs. In total, 478 jobs have been completed.
- At the end of each job, the events (in LCIO format), metadata, and input+log-file tarballs are uploaded to the grid.
- At the end of each channel, summary metadata of all jobs of the channel are uploaded to the grid. This triggers the simulation and reconstruction system under DIRAC to do it's thing.

ILC 250 GeV production: Generation status (OLD numbers...)

- Done on a batch-farm.
- For $\leq$ 5f : 116 channels. 96 virtual γ induced channels, deferred because of remaining Whizard issues details.
- As of today, 104 channels are done, producing 2.7 billion events in 15788 LCIO files details, occupying 5.4 TB. This used 7233 CPU hours, obtained in $\sim$ 10 days.
- In most cases: one channel = one generation job, but in some processes alone represent $\sim$ billion events: split in several jobs. In total, 478 jobs have been completed.
- At the end of each job, the events (in LCIO format), metadata, and input+log-file tarballs are uploaded to the grid.
- At the end of each channel, summary metadata of all jobs of the channel are uploaded to the grid. This triggers the simulation and reconstruction system under DIRAC to do it's thing.

What next ? : Future developments

Already in Whizard, but not yet tested by us:

- gluon matching between ME and PS. Some work done (Zhije Zhao), more needed.
- Pythia8 instead of Pythia6 for hadronisation: Some work needed. In principle in place (Zhije Zhao), but integration into the work-flow needed.

Medium-term wishes for Whizard:

- γ ISR/FSR matching
- Work out priority processes for EW-NLO.

What next ? : Future developments

Already in Whizard, but not yet tested by us:

- gluon matching between ME and PS. Some work done (Zhije Zhao), more needed.
- Pythia8 instead of Pythia6 for hadronisation: Some work needed. In principle in place (Zhije Zhao), but integration into the work-flow needed.

Medium-term wishes for Whizard:

- γ ISR/FSR matching
- Work out priority processes for EW-NLO.

What next ? : Future developments

Other issues:

- The low-$P_\perp$ hadron generator is ∼ orphaned (written by Tim Barklow long ago, needs Pythia6 + Whizard v1 to produce the gamma-flux)
- Energy-asymmetric collisions does not work with Whizard+Pythia6. Need to work on that...
- Document the work-flow, beyond the comments in the scripts ....
- Would be good to have other generators for cross-checks:
  - Specialised generators: BHWide/BHLumi for better Bhabhas, CAIN for beam-beam effects.
  - General generators: Pythia8, Madgraph, Sherpa... Some activity, both from users, and - at least for Sherpa - by authors.
  - But remember: To be a player, beam-spectrum and polarised beams is a MUST !

What next ? : Future developments

Other issues:

- The low-$P_\perp$ hadron generator is $\sim$ orphaned (written by Tim Barklow long ago, needs Pythia6 + Whizard v1 to produce the gamma-flux)
- Energy-asymmetric collisions does not work with Whizard+Pythia6. Need to work on that...
- Document the work-flow, beyond the comments in the scripts ....
- Would be good to have other generators for cross-checks:
  - Specialised generators: BHWide/BHLumi for better Bhabhas, CAIN for beam-beam effects.
  - General generators: Pythia8, Madgraph, Sherpa... Some activity, both from users, and - at least for Sherpa - by authors.
  - But remember: To be a player, beam-spectrum and polarised beams is a MUST !

Conclusions

- The precision-goal of future e^+e^- colliders are such that it is not permissible that MC statistics or coverage could constitute an major systematic uncertainty.
- We showed how the generation of the full SM, with statistics well above the expected H20@250 data set is achieved.
  - It consists of bringing a large number of different codes together:
    - Whizard, Pythia, and Tauola for physics generation
    - GuineaPig and Circe2 for beam-properties
    - SGV-GuineaPig and the Peskin/Barklow generator for pile-up
    - In addition, input from machine-physics and data from LEP was used.
- This full data is organised and documented in a physics-oriented fashion, for the benefit of the end-user.
- Full simulation and reconstruction has been done. However, some channels remain to be generated, once issues have been resolved.

Conclusions

- The precision-goal of future e^+e^- colliders are such that it is not permissible that MC statistics or coverage could constitute an major systematic uncertainty.
- We showed how the generation of the full SM, with statistics well above the expected H20@250 data set is achieved.
  - It consists of bringing a large number of different codes together:
    - Whizard, Pythia, and Tauola for physics generation
    - GuineaPig and Circe2 for beam-properties
    - SGV+GuineaPig and the Peskin/Barklow generator for pile-up
    - In addition, input from machine-physics and data from LEPII was used.
- This full data is organised and documented in a physics-oriented fashion, for the benefit of the end-user.
- Full simulation and reconstruction has been done. However, some channels remain to be generated, once issues have been resolved.

Conclusions

- The precision-goal of future e^+e^- colliders are such that it is not permissible that MC statistics or coverage could constitute an major systematic uncertainty.
- We showed how the generation of the full SM, with statistics well above the expected H20@250 data set is achieved.
  - It consists of bringing a large number of different codes together:
    - Whizard, Pythia, and Tauola for physics generation
    - GuineaPig and Circe2 for beam-properties
    - SGV+GuineaPig and the Peskin/Barklow generator for pile-up
    - In addition, input from machine-physics and data from LEPII was used.
- This full data is organised and documented in a physics-oriented fashion, for the benefit of the end-user.
- Full simulation and reconstruction has been done. However, some channels remain to be generated, once issues have been resolved.

Conclusions

- The precision-goal of future e^+e^- colliders are such that it is not permissible that MC statistics or coverage could constitute an major systematic uncertainty.
- We showed how the generation of the full SM, with statistics well above the expected H20@250 data set is achieved.
  - It consists of bringing a large number of different codes together:
    - Whizard, Pythia, and Tauola for physics generation
    - GuineaPig and Circe2 for beam-properties
    - SGV+GuineaPig and the Peskin/Barklow generator for pile-up
    - In addition, input from machine-physics and data from LEPII was used.
- This full data is organised and documented in a physics-oriented fashion, for the benefit of the end-user.
- Full simulation and reconstruction has been done. However, some channels remain to be generated, once issues have been resolved.

BACKUP

BACKUP SLIDES

Documentation

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam nature ($e^{+(-)}$ or γ), beam polarisation, beam spectrum, cross-section, run- and event-number of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, total number of events generated, total integrated luminosity, technicalities, ...
  - Browasble on the Web and uploaded to the Grid.
  - Once on the grid, the metadata is read by DIRAC to orchestrate the detector simulation.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,... (Sufficient information to re-run):
  - On the Web, in full.
  - In tar files - parallel to generated files, also stored on the Grid.

Documentation

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam nature ($e^{+(-)}$ or γ), beam polarisation, beam spectrum, cross-section, run- and event-number of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, total number of events generated, total integrated luminosity, technicalities, ...
  - Browasble on the Web and uploaded to the Grid.
  - Once on the grid, the metadata is read by DIRAC to orchestrate the detector simulation.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,... (Sufficient information to re-run):
  - On the Web, in full.
  - In tar files - parallel to generated files, also stored on the Grid.

Documentation

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam nature (e$^{+(-)}$ or γ), beam polarisation, beam spectrum, cross-section, run- and event-number of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, total number of events generated, total integrated luminosity, technicalities, ...
  - Browasble on the Web and uploaded to the Grid.
  - Once on the grid, the metadata is read by DIRAC to orchestrate the detector simulation.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,... (Sufficient information to re-run):
  - On the Web, in full.
  - In tar files - parallel to generated files, also stored on the Grid.

Documentation

Created by generation job, driven by the contents of the process-definition Sindarin script and common conditions. It goes into:

- The event header:
  - Process-id, beam nature ($e^{+(-)}$ or γ), beam polarisation, beam spectrum, cross-section, run- and event-number of each event.
- The generator meta-data files:
  - Condenses job-specific information from Whizard logs.
  - Contains: process, cross section, polarisation, file-names, total number of events generated, total integrated luminosity, technicalities, ...
  - Browasble on the Web and uploaded to the Grid.
  - Once on the grid, the metadata is read by DIRAC to orchestrate the detector simulation.
- Steering-files, logs, pdf:s with diagrams, integration grids, output other than the events,... (Sufficient information to re-run):
  - On the Web, in full.
  - In tar files - parallel to generated files, also stored on the Grid.

The files: where and what

- The files can be found on
  - .../ilc/prod/ilc/mc-2020/generated/250-SetA/*
- Anatomy of the file-name:
  - E250-SetA.P4f_ww_h.Gwhizard-2_8_5.eL.pR.I500066.317.slcio
    - E250-SetA = E(nergy)[E_{cms}]_[beam conditions]
    - P4f_ww_h =P(rocess)[(aa_[1])number of final state particles]_[NC/CC/mixed flavour combination]_[hadronic/leptonic/semi-leptonic final state]
    - Gwhizard-2_8_5 =G(enerator)[name and version of generator] (NB: Unfortunately, this tag is suppressed in the name of the DST files...)
    - eL = e(lectron beam)[polarisation (L/R) or γ type (W/B)]
    - pR = p(ositron beam)[polarisation (l/R) or γ type (W/B)]
    - I = I(dentification number)[id (also run number)]

back

[1] added to γγ events to separate from e^+e^- events with the same number of fermions

The files: where and what

- The files can be found on
  - .../ilc/prod/ilc/mc-2020/generated/250-SetA/*
- Anatomy of the file-name:
  - E250-SetA.P4f_ww_h.Gwhizard-2_8_5.eL.pR.I500066.317.slcio
    - E250-SetA = E(nergy)[E_{CMS}]_[beam conditions]
    - P4f_ww_h =P(rocess)[(aa_[1])number of final state particles]_[NC/CC/mixed flavour combination]_[hadronic/leptonic/semi-leptonicfinal state]
    - Gwhizard-2_8_5 =G(enerator)[name and version of generator] (NB: Unfortunately, this tag is suppressed in the name of the DST files...)
    - eL = e(lectron beam)[polarisation (L/R) or γ type (W/B)]
    - pR = p(ositron beam)[polarisation (l/R) or γ type (W/B)]
    - I = I(detification number)[id (also run number)]

back

[1] added to $\gamma\gamma$ events to separate from e^+e^- events with the same number of fermions

The files: where and what

- The files can be found on
  - .../ilc/prod/ilc/mc-2020/generated/250-SetA/*
- Anatomy of the file-name:
  - E250-SetA.P4f_ww_h.Gwhizard-2_8_5.eL.pR.I500066.317.slcio
    - E250-SetA = E(nergy)[E_{CMS}]_[beam conditions]
    - P4f_ww_h =P(rocess)[(aa_[1])number of final state particles] _[NC/CC/mixed flavour combination]_[hadronic/leptonic/semi-leptonicfinal state]
    - Gwhizard-2_8_5 =G(enerator)[name and version of generator] (NB: Unfortunately, this tag is suppressed in the name of the DST files...)
    - eL = e(lectron beam)[polarisation (L/R) or γ type (W/B)]
    - pR = p(ositron beam)[polarisation (l/R) or γ type (W/B)]
    - I = I(detification number)[id (also run number)]

back

[1] added to $\gamma\gamma$ events to separate from e^+e^- events with the same number of fermions

The files: where and what

- The files can be found on
  - .../ilc/prod/ilc/mc-2020/generated/250-SetA/*
- Anatomy of the file-name:
  - E250-SetA.P4f_ww_h.Gwhizard-2_8_5.eL.pR.I500066.317.slcio
    - E250-SetA = E(nergy)[E_{CMS}]_[beam conditions]
    - P4f_ww_h =P(rocess)[(aa_[1])number of final state particles] _[NC/CC/mixed flavour combination]_[hadronic/leptonic/semi-leptonicfinal state]
    - Gwhizard-2_8_5 =G(enerator)[name and version of generator] (NB: Unfortunately, this tag is suppressed in the name of the DST files...)
    - eL = e(lectron beam)[polarisation (L/R) or γ type (W/B)]
    - pR = p(ositron beam)[polarisation (l/R) or γ type (W/B)]
    - I = I(detification number)[id (also run number)]

back

[1] added to $\gamma\gamma$ events to separate from e^+e^- events with the same number of fermions

The files: where and what

- The files can be found on
  - .../ilc/prod/ilc/mc-2020/generated/250-SetA/*
- Anatomy of the file-name:
  - E250-SetA.P4f_ww_h.Gwhizard-2_8_5.eL.pR.I500066.317.slcio
    - E250-SetA = E(nergy)[E_{CMS}]_[beam conditions]
    - P4f_ww_h =P(rocess)[(aa_[1])number of final state particles] _[NC/CC/mixed flavour combination]_[hadronic/leptonic/semi-leptonicfinal state]
    - Gwhizard-2_8_5 =G(enerator)[name and version of generator] (NB: Unfortunately, this tag is suppressed in the name of the DST files...)
    - eL = e(lectron beam)[polarisation (L/R) or γ type (W/B)]
    - pR = p(ositron beam)[polarisation (l/R) or γ type (W/B)]
    - I = I(detification number)[id (also run number)]

back

[1] added to $\gamma\gamma$ events to separate from e^+e^- events with the same number of fermions

The files: where and what

- The files can be found on
  - .../ilc/prod/ilc/mc-2020/generated/250-SetA/*
- Anatomy of the file-name:
  - E250-SetA.P4f_ww_h.Gwhizard-2_8_5.eL.pR.I500066.317.slcio
    - E250-SetA = E(nergy)[E_{CMS}]_[beam conditions]
    - P4f_ww_h =P(rocess)[(aa_[1])number of final state particles] _[NC/CC/mixed flavour combination]_[hadronic/leptonic/semi-leptonicfinal state]
    - Gwhizard-2_8_5 =G(enerator)[name and version of generator] (NB: Unfortunately, this tag is suppressed in the name of the DST files...)
    - eL = e(lectron beam)[polarisation (L/R) or γ type (W/B)]
    - pR = p(ositron beam)[polarisation (l/R) or γ type (W/B)]
    - I = I(detification number)[id (also run number)]

back

[1] added to $\gamma\gamma$ events to separate from e^+e^- events with the same number of fermions

The files: where and what

- The files can be found on
  - .../ilc/prod/ilc/mc-2020/generated/250-SetA/*
- Anatomy of the file-name:
  - E250-SetA.P4f_ww_h.Gwhizard-2_8_5.eL.pR.I500066.317.slcio
    - E250-SetA = E(nergy)[E_{CMS}]_[beam conditions]
    - P4f_ww_h =P(rocess)[(aa_[1])number of final state particles] _[NC/CC/mixed flavour combination]_[hadronic/leptonic/semi-leptonicfinal state]
    - Gwhizard-2_8_5 =G(enerator)[name and version of generator] (NB: Unfortunately, this tag is suppressed in the name of the DST files...)
    - eL = e(lectron beam)[polarisation (L/R) or γ type (W/B)]
    - pR = p(ositron beam)[polarisation (l/R) or γ type (W/B)]
    - I = I(detification number)[id (also run number)]

back

[1]added to $\gamma\gamma$ events to separate from e^+e^- events with the same number of fermions

The files: News

- The format is LCIO, the contents is one MCParticle collection/event
  - This is not the MCParticle you will see on the DST: Obviously, on the DST, particles created in the detector simulation and from overlay are added. But not only:
    - The crossing-angle boost is added.
    - The production point has been swimmed around.
    - For flying particles, the effect of the detector B-field and energy loss is added, so that the decay properties are modified.
- The event numbers are unique, without gaps.
  - The event header contains run and events number which is now unique, and carried on all the way to the DST.
- The header also contains the cross-section of the process. A GOTCHA: the true cross-section. It does not take into account that the luminosity for e^+e^-, $e^{+(-)}\gamma$ and $\gamma\gamma$ processes are different!
- The early parts of the event record (initial state, resonance insertions, status codes) is different.

The files: News

- The format is LCIO, the contents is one MCParticle collection/event
  - This is not the MCParticle you will see on the DST: Obviously, on the DST, particles created in the detector simulation and from overlay are added. But not only:
    - The crossing-angle boost is added.
    - The production point has been swimmed around.
    - For flying particles, the effect of the detector B-field and energy loss is added, so that the decay properties are modified.
- The event numbers are unique, without gaps.
  - The event header contains run and events number which is now unique, and carried on all the way to the DST.
- The header also contains the cross-section of the process. A GOTCHA: the true cross-section. It does not take into account that the luminosity for e^+e^-, $e^{+(-)}\gamma$ and $\gamma\gamma$ processes are different!
- The early parts of the event record (initial state, resonance insertions, status codes) is different.

The files: News

- The format is LCIO, the contents is one MCParticle collection/event
  - This is not the MCParticle you will see on the DST: Obviously, on the DST, particles created in the detector simulation and from overlay are added. But not only:
    - The crossing-angle boost is added.
    - The production point has been swimmed around.
    - For flying particles, the effect of the detector B-field and energy loss is added, so that the decay properties are modified.
- The event numbers are unique, without gaps.
  - The event header contains run and events number which is now unique, and carried on all the way to the DST.
- The header also contains the cross-section of the process. A GOTCHA: the true cross-section. It does not take into account that the luminosity for e^+e^-, $e^{+(-)}\gamma$ and $\gamma\gamma$ processes are different!
- The early parts of the event record (initial state, resonance insertions, status codes) is different.

The files: News

- The format is LCIO, the contents is one MCParticle collection/event
  - This is not the MCParticle you will see on the DST: Obviously, on the DST, particles created in the detector simulation and from overlay are added. But not only:
    - The crossing-angle boost is added.
    - The production point has been swimmed around.
    - For flying particles, the effect of the detector B-field and energy loss is added, so that the decay properties are modified.
- The event numbers are unique, without gaps.
  - The event header contains run and events number which is now unique, and carried on all the way to the DST.
- The header also contains the cross-section of the process. A GOTCHA: the true cross-section. It does not take into account that the luminosity for e^+e^-, $e^{+(-)}\gamma$ and $\gamma\gamma$ processes are different!
- The early parts of the event record (initial state, resonance insertions, status codes) is different.

Whizard 1.x $\to$ 2.x issues

Virtual incoming γ:s

- $\mathrm{e}^+\mathrm{e}^- \to e^+e^- f\bar{f}$ is generated in different ways, depending on the Q^2 between in- and out-going $e^{+(-)}$:
  1. Both high: as M.E. of $\mathrm{e}^+\mathrm{e}^- \to e^+e^- f\bar{f}$
  2. One high, the other low: as M.E. of $e^{+(-)}\gamma^* \to e^{+(-)}f\bar{f}$
  3. Both low: as M.E. of $\gamma^*\gamma^* \to f\bar{f}$

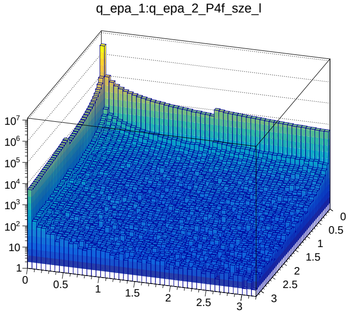
  where the γ^* is generated with the EPA off an incoming $e^{+(-)}$. The scattered $e^{+(-)}$ is present in the final state (the "beam-remnant").
- In Whizard2, the cross-section of 2 (3) is two (four) times too low, *if circe is used*.
- For now, these channels are being deferred in the production.

Whizard 1.x → 2.x issues

Virtual incoming γ:s

- $\mathrm{e}^+\mathrm{e}^- \rightarrow e^+e^-f\bar{f}$ is generated in different ways, depending on the Q^2 between in- and out-going $e^{+(-)}$:
  1. Both high: as M.E. of $\mathrm{e}^+\mathrm{e}^- \rightarrow e^+e^-f\bar{f}$
  2. One high, the other low: as M.E. of $e^{+(-)}\gamma^* \rightarrow e^{+(-)}f\bar{f}$
  3. Both low: as M.E. of $\gamma^*\gamma^* \rightarrow f\bar{f}$

  where the γ^* is generated with the EPA off an incoming $e^{+(-)}$. The scattered $e^{+(-)}$ is present in the final state (the "beam-remnant").



- In Whizard2, the cross-section of 2 (3) is two (four) times too low, *if circe is used*.
- For now, these channels are being deferred in the production.

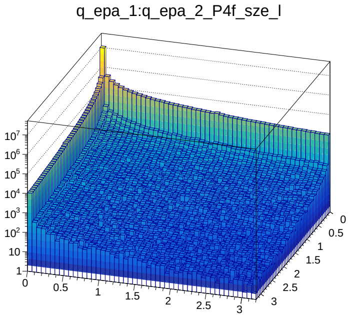
back

Whizard 1.x → 2.x issues

Virtual incoming γ:s

- $\mathrm{e^+e^-} \rightarrow e^+e^-f\bar{f}$ is generated in different ways, depending on the Q^2 between in- and out-going $e^{+(-)}$:
  1. Both high: as M.E. of $\mathrm{e^+e^-} \rightarrow e^+e^-f\bar{f}$
  2. One high, the other low: as M.E. of $e^{+(-)}\gamma^* \rightarrow e^{+(-)}f\bar{f}$
  3. Both low: as M.E. of $\gamma^*\gamma^* \rightarrow f\bar{f}$

  where the γ^* is generated with the EPA off an incoming $e^{+(-)}$. The scattered $e^{+(-)}$ is present in the final state (the "beam-remnant").



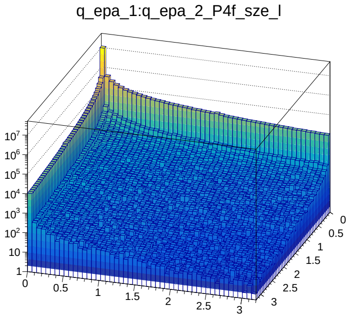
- In Whizard2, the cross-section of 2 (3) is two (four) times too low, *if circe is used*.
- For now, these channels are being deferred in the production.

Whizard 1.x $\rightarrow$ 2.x issues

Virtual incoming γ:s

- $\mathrm{e}^+\mathrm{e}^- \rightarrow e^+e^-f\bar{f}$ is generated in different ways, depending on the Q^2 between in- and out-going $e^{+(-)}$:
  1. Both high: as M.E. of $\mathrm{e}^+\mathrm{e}^- \rightarrow e^+e^-f\bar{f}$
  2. One high, the other low: as M.E. of $e^{+(-)}\gamma^* \rightarrow e^{+(-)}f\bar{f}$
  3. Both low: as M.E. of $\gamma^*\gamma^* \rightarrow f\bar{f}$

  where the γ^* is generated with the EPA off an incoming $e^{+(-)}$. The scattered $e^{+(-)}$ is present in the final state (the "beam-remnant").



- In Whizard2, the cross-section of 2 (3) is two (four) times too low, *if circe is used*.
- For now, these channels are being deferred in the production.

ILC 250 GeV production: Whizard 1.x $\rightarrow$ 2.x

- DBD was done with v. 1.95
- v2.x is a major re-write; we finally use v2.8.5. Many new features.
  - New, better steering
  - Things done by us now part of the main code:
    - Interface to PYTHIA (parton-shower and hadronisation).
    - Interface TAUOLA (polarised τ-decays)
    - Beam-spectrum CIRCE2: Easier to study of impact of beam parameters
  - Generated events directly in LCIO format.
  - Samples from new BSM models much easier to create, using tools like UFO.
  - 8 fermion final states possible (ttH !). Was not (practically) possible with Whizard 1.95.

ILC 250 GeV production: Whizard 1.x → 2.x

- DBD was done with v. 1.95
- v2.x is a major re-write; we finally use v2.8.5. Many new features.
  - New, better steering
  - Things done by us now part of the main code:
    - Interface to PYTHIA (parton-shower and hadronisation).
    - Interface TAUOLA (polarised τ-decays)
    - Beam-spectrum CIRCE2: Easier to study of impact of beam parameters
  - Generated events directly in LCIO format.
  - Samples from new BSM models much easier to create, using tools like UFO.
  - 8 fermion final states possible (ttH !). Was not (practically) possible with Whizard 1.95.

ILC 250 GeV production: Whizard 1.x $\rightarrow$ 2.x

- DBD was done with v. 1.95
- v2.x is a major re-write; we finally use v2.8.5. Many new features.
  - New, better steering
  - Things done by us now part of the main code:
    - Interface to PYTHIA (parton-shower and hadronisation).
    - Interface TAUOLA (polarised τ-decays)
    - Beam-spectrum CIRCE2: Easier to study of impact of beam parameters
  - Generated events directly in LCIO format.
  - Samples from new BSM models much easier to create, using tools like UFO.
  - 8 fermion final states possible (*ttH* !). Was not (practically) possible with Whizard 1.95.

Whizard 1.x (DBD) $\rightarrow$ Whizard 2.x (MC2020).

Polarised τ decays

- Whizard makes polarised τ:s $\Rightarrow$ must decay polarised τ:s $\Rightarrow$ TAUOLA.
- But: Pythia doesn't handle polarised decays $\Rightarrow$ tweak needed.
- This was done by us with Whizard1, Whizard2 implementation of this was not correct.
- We helped the authors to get this right.

Colour-connection

- Whizard colour connects quarks $\Rightarrow$ transfer to Pythia P.S. code.
- This was done by us with Whizard1, Whizard2 did not do it.
- We worked with the authors to fix this: Mark connection by artificial resonances in the event before handing over to Pythia.

Whizard 1.x (DBD) $\rightarrow$ Whizard 2.x (MC2020).

Polarised τ decays

- Whizard makes polarised τ:s $\Rightarrow$ must decay polarised τ:s $\Rightarrow$ TAUOLA.
- But: Pythia doesn't handle polarised decays $\Rightarrow$ tweak needed.
- This was done by us with Whizard1, Whizard2 implementation of this was not correct.
- We helped the authors to get this right.

Colour-connection

- Whizard colour connects quarks $\Rightarrow$ transfer to Pythia P.S. code.
- This was done by us with Whizard1, Whizard2 did not do it.
- We worked with the authors to fix this: Mark connection by artificial resonances in the event before handing over to Pythia.

Whizard 1.x (DBD) $\rightarrow$ Whizard 2.x (MC2020).

Polarised τ decays

- Whizard makes polarised τ:s $\Rightarrow$ must decay polarised τ:s $\Rightarrow$ TAUOLA.
- But: Pythia doesn't handle polarised decays $\Rightarrow$ tweak needed.
- This was done by us with Whizard1, Whizard2 implementation of this was not correct.
- We helped the authors to get this right.

Colour-connection

- Whizard colour connects quarks $\Rightarrow$ transfer to Pythia P.S. code.
- This was done by us with Whizard1, Whizard2 did not do it.
- We worked with the authors to fix this: Mark connection by artificial resonances in the event before handing over to Pythia.