

MC Generators for Future Colliders in Key4hep

Thomas Madlener (DESY) Alan Price (Cracow),
Dirk Zerwas (DMLab)
MCNet
March 16, 2026

- Introduction
- Comparing
- Integrating
- Open Questions



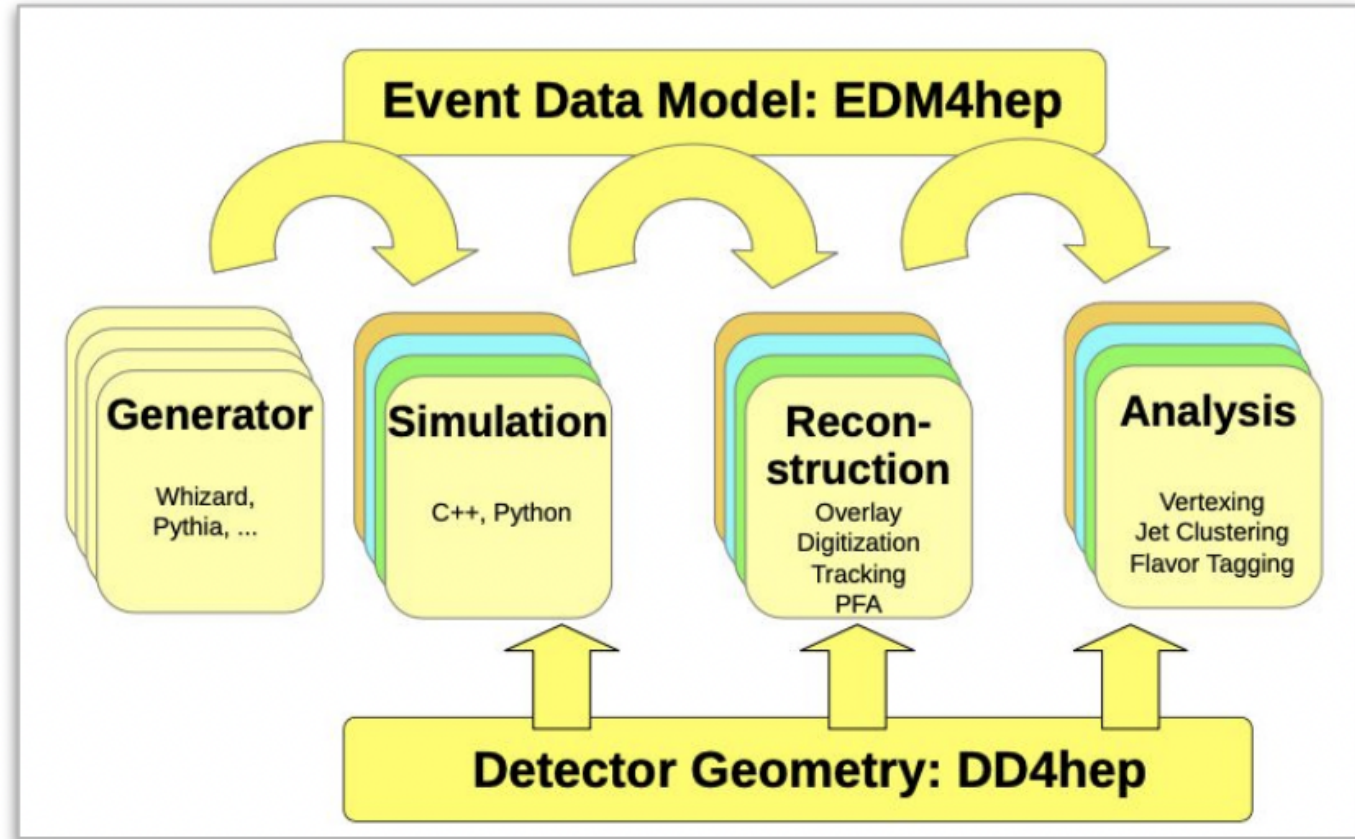
Introduction

k4GeneratorsConfig started in
ECFA Higgs/TopEW Factories WG2: Physics Analysis Tools

- Patrizia Azzi
- Fulvio Piccinini
- Dirk Zerwas

Key4hep Ecosystem (Thomas, Frank et al):

- **edm4hep**
- Geometry DD4HEP
- Gaudi
- DIRAC
- ILCSoft
- FCC
- CEPCSW
- ACTS....



Adopted by ECFA(well...), CEPC, C³, FCC, ILC :

- Interoperability of algorithms
- comparisons and improvements

Configuration

Attobarn data with N^n LO under control:

- many generators claim the same precision (differing approaches)
- needs high precision in generation
- first step: **cross section**
- second step: **differential distributions**
- **Technical Benchmarks**

k4GeneratorsConfig in key4HEP:

- define process
- define generators
- generates datacards for the generators
 - **dynamic and template based**
- generates event generation script in key4HEP environment (run anywhere, well.....)

In practice:

- python
- dynamic loading of generators by name
- OO usage 2+1 levels deep

```
Generators:
- Sherpa
- Whizard
- Madgraph
- KKMC
- Pythia

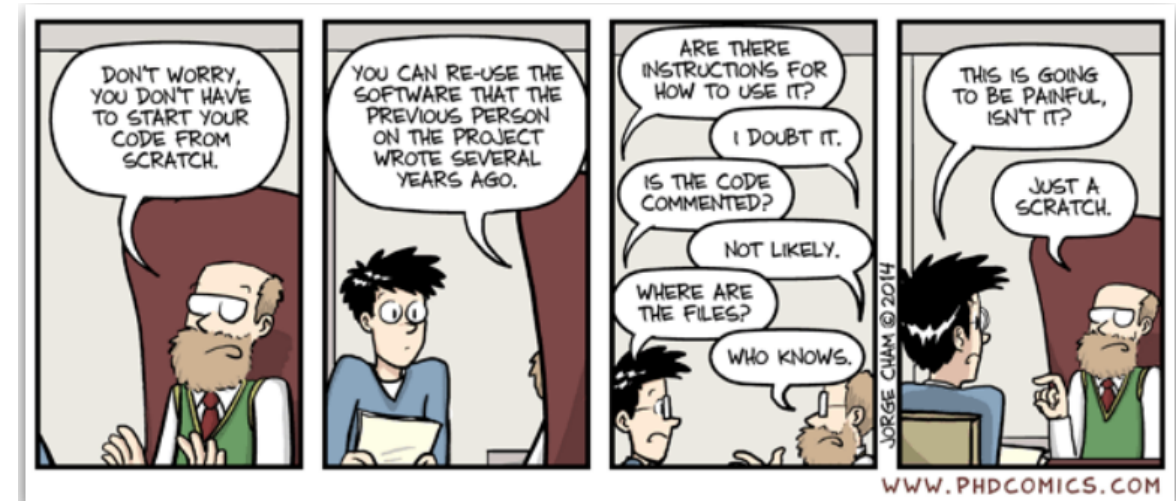
OutputFormat: hepmc3
OutDir: Run-Cards
Events: 10000
EventMode: unweighted

SqrtS: 91.2
Model: SM
ISRMode: 0

Processes:
  Muon:
    Final: [13, -13]
    Order: [2, 0]
  Tau:
    Final: [15, -15]
    Order: [2, 0]

Sherpa:
  Run:
    EW_SCHEME: 3

ParticleData:
  23:
    mass: 91.1876
    width: 2.4952
```



behind the scenes:

- **converter HEPMC/LHE to edm4hep**
- Pythia main with PhaseSpace cuts

Technical Features

Goal:

- detect
- correct
- follow up

key4HEP

- Whizard in Key4hep works on CENTOS, ALMA, UBUNTU
- Whizard does not work on ALMA in CI
- traced to incomplete setup on all OS but UBUNTU
- bugfix (SPACK) provided by Thomas

KKMC:

- cross section not provided
- corrected by others authors

Babayaga

- cross section not provided
- update by author, included in Key4hep and tested (solved)

The current MODEL:

- **SUPER-SPLIT SUPERSYMMETRY** [arXiv hep-th/0503249](https://arxiv.org/abs/hep-th/0503249) Fox et al

MADGRAPH:

- $\sigma(\text{d}\bar{\text{b}}) > \sigma(\text{b}\bar{\text{b}})$
- LHC inspired implicit pT cut (bQuark)
- reset all cuts
- update of MADGRAPH setup (incomplete installation) in Key4hep

k4GeneratorsConfig

- consistent initialization of parameters (de facto SLHA)
- overconstrained input (eg EW scheme)
- database
- generator specific process configuration additions

Conversion hepmc3 to edm4HEP

Goal:

- lossless conversion
- edm4hep evolved from HEPEVT

edm4hep v1.0 released

Implementation in k4GeneratorsConfig

- convertHepMC2EDM4HEP.cxx as extended version of convert_example.cc
- new WriterEDM4HEP.cxx inheriting from (hepmc)Writer

Particle

modified after Chris and Andy's visit to DESY:

- 4 vector (changed to double)
- Charge
- Vertex position & time
- End Vertex position
- Helicity (replaced 3d spin)
- PDGID
- Charge
- Generator Status
- Removed colorflow

Run:

- Vector edm4HEPTool:
 - hepMCTool.name
 - hepMCTool.version
 - hepMCTool.description
- weight names

Event

- event number
- weights
- cross section(s)
- sqrts (with a bit of gymnastics)
- signalVertex from signal_vertex_id fallback signal_process_vertex
- PDF: partonIDs from pdfinfo (stripped down after Chris and Andy's visit)

Conversion Move to hepmc3

Reference:

- [MergeRequest](#)
- Location ConvertExample
- new compilation flag: -DHEPMC3_ENABLE_KEY4HEP=ON
- needs compilation flag: -DHEPMC3_BUILD_EXAMPLES=ON

Implementation

- several updates to code thanks to **Andriis** suggestions/questions and corrections
- protect a la Francaise (ceinture et bretelles) against nonsense configurations
- sanity checks in CMakeLists.txt

Tests done:

- Compilation against Key4hep nightlies: **ok**
- running conversion: **ok**
- running analysis on converted files: **ok**

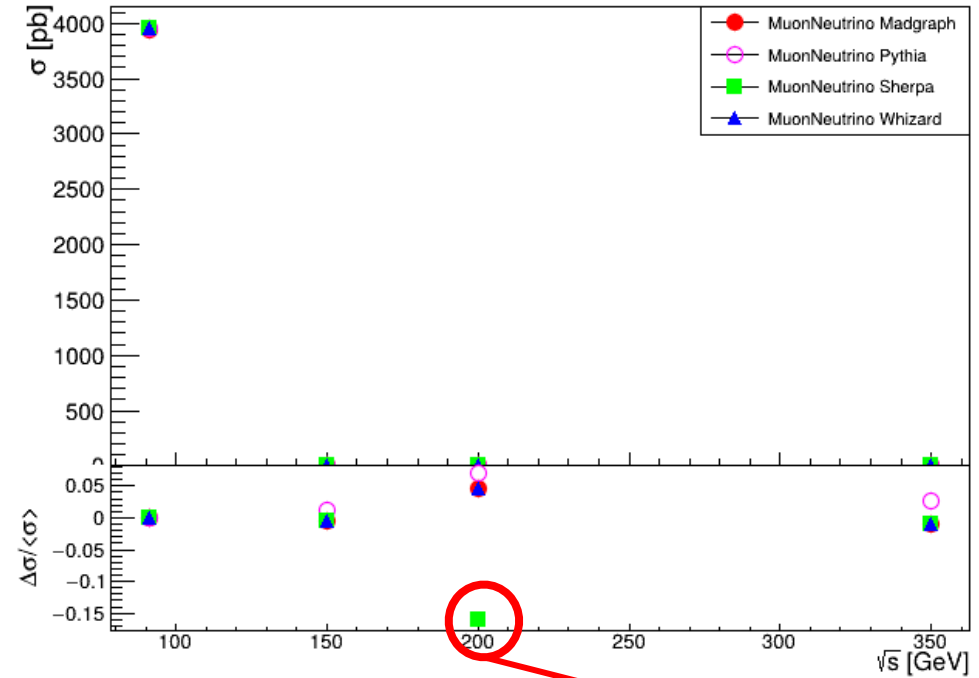
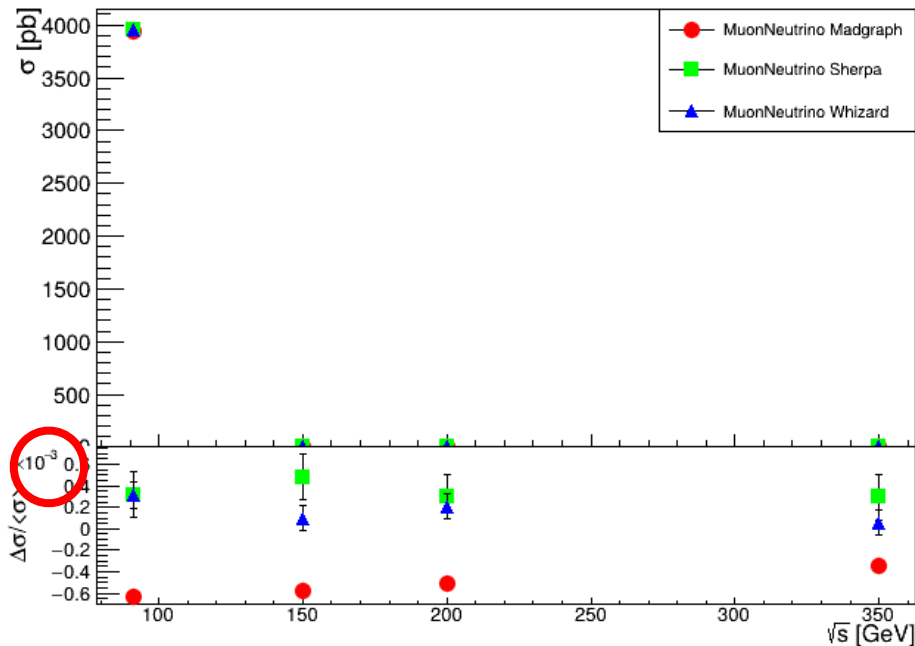
CI coded:

- Compilation with cvmfs: **ok**
- **Andriis: compilation with lcg**

Generators: Fermion Pairs

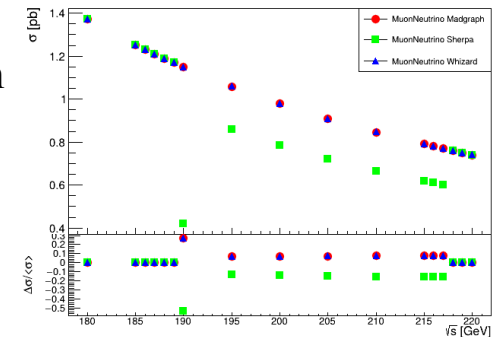
BottomUp:

- fermion pairs
- LO
- no ISR
- MuonNeutrino: 1 diagram only, no interference



Processing:

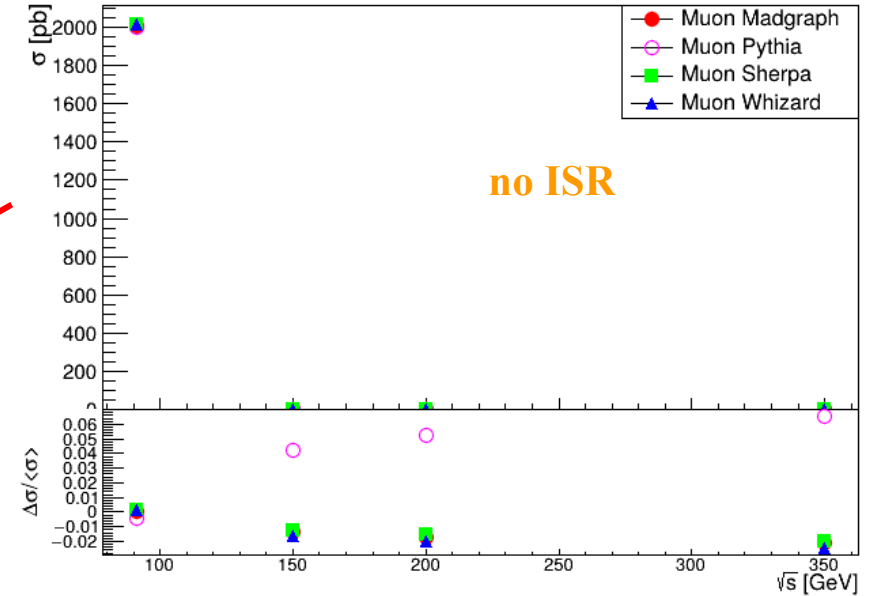
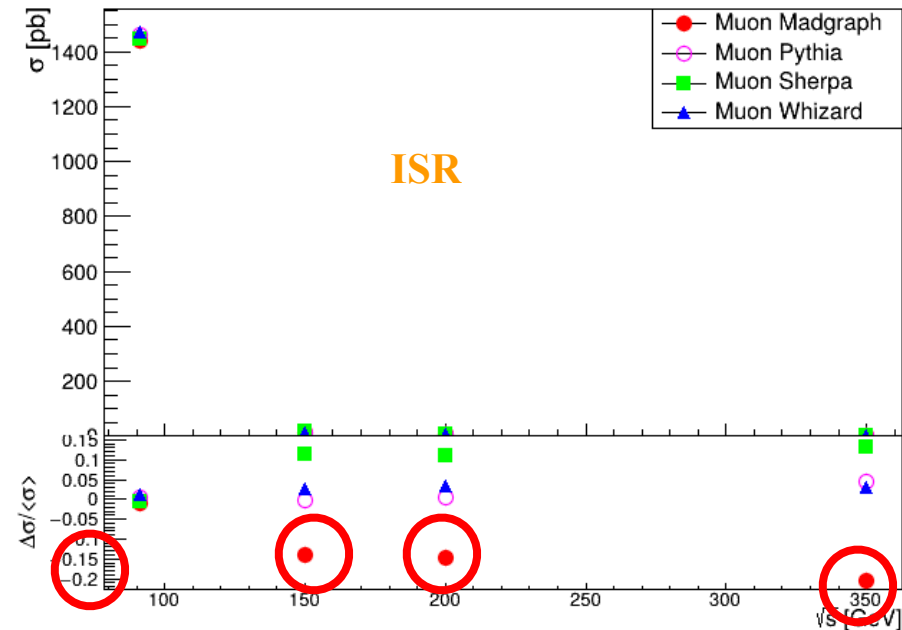
- 3% agreement Madgraph/Pythia/Sherpa/Whizard
- Drop of -15% for Sherpa@200GeV (in all fermion pairs)
- Alan and Daniel investigated: fix in Sherpa 3.0.X
- Fix confirmed in key4HEP
- Madgraph/Sherpa/Whizard agreement: <0.1%
- Sherpa/Whizard agreement: ~0.01%



Generators: Fermion Pairs

BottomUp:

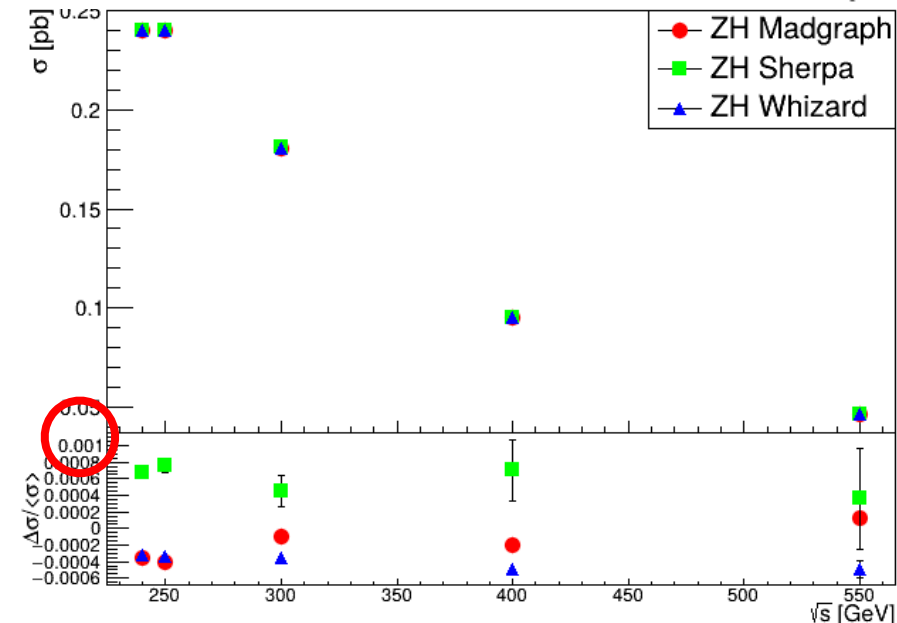
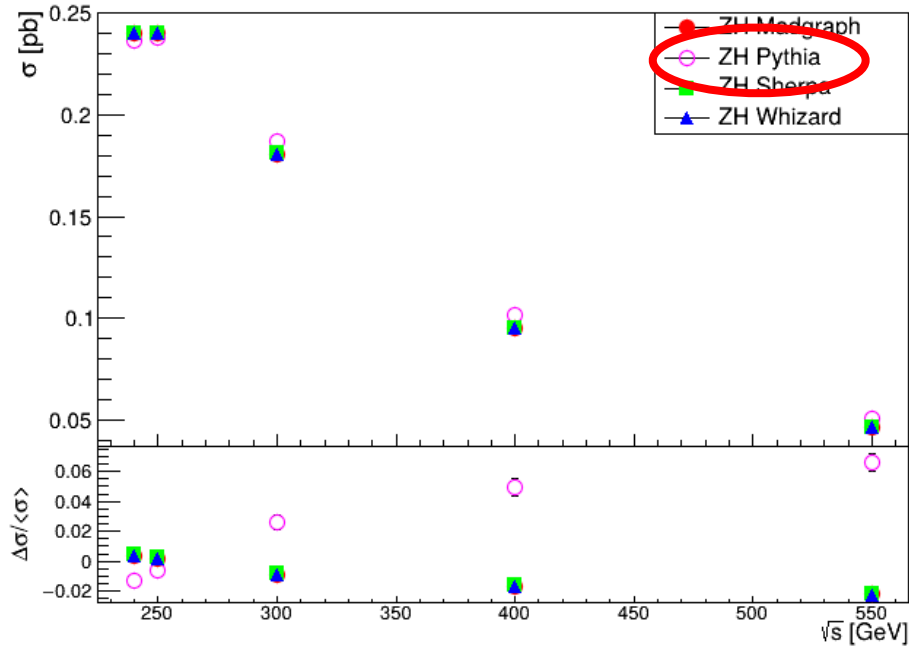
- Muons: 2 diagram, interference
- LO
- ISR:
 - YFS eg Sherpa, KKMC
 - PDF eg Pythia, Madgraph



Processing:

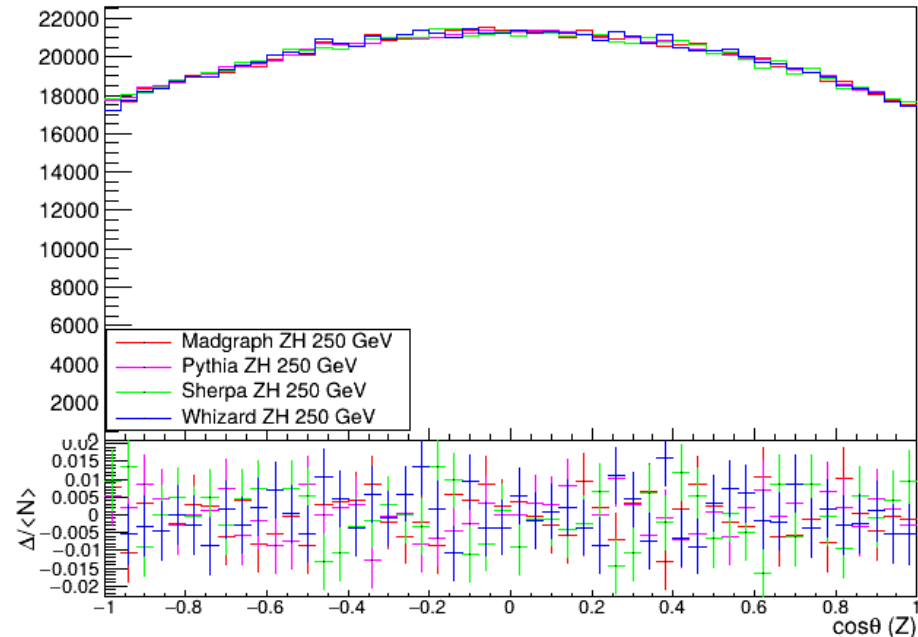
- no ISR
 - Pythia versus Madgraph, Sherpa, Whizard
 - kind of understood: $\sin^2\theta$ is intelligently set by Pythia
- ISR
 - Reasonable agreement Pythia/Sherpa/Whizard (also at 200GeV)
 - Madgraph off up to 20%
 - SM ok-ish, SM-FULL
 - $x=1$ effect: 10^{-6} precision in ISR
 - $\hat{s} = x_1 * x_2 * s$ (imprecise in 3 terms (massive initial state))
 - boost with massive initial state particles
 - **ad hoc correction 3.5.9**

Generators: ZH



ZH LO noISR:

- several percent agreement level
- differential distribution ($\cos\theta$) roughly ok
- Sherpa versus Madgraph/Whizard
 - globally: permil level agreement
 - locally: 0.01 % level
 - Sherpa being checked by Alan



Open Questions

Full circle:

- **next HEPMC release:**
 - **installation in compilation in key4HEP (SPAC recipe)**
 - **test in k4GeneratorsConfig**
 - **removal of duplicate code**

Helicity:

- MCNet provides structures to store
- Could MCNet provide guidance? (a de facto Standard)?

Status Codes:

- DO WE HAVE a proposal/request/example?