Automated NLO calculations with **GoSam**

Gionata Luisoni

gionata.luisoni@durham.ac.uk

Institute for Particle Physics Phenomenology
University of Durham
Max-Planck-Institut für Physik
München

In collaboration with:
G.Cullen, N. Greiner, G.Heinrich, P.Mastrolia, G.Ossola, T.Reiter, F. Tramontano

GoSam release: arXiv:1111.2034 [hep-ph] | http://gosam.hepforge.org/

HP2: High Precision for Hard Processes
München, 04.09.2012

Motivation

- Progresses in NLO calculation:
  - pp $\longrightarrow W+3$ jets — Blackhat (09) / Rocket (09)
  - pp $\longrightarrow t\bar{t}b\bar{b}$ — Denner-Dittmaier (09) / HELAC-NLO (09)
  - pp $\longrightarrow Z(\gamma)+3$ jets — Blackhat (10)
  - pp $\longrightarrow t\bar{t}jj$ — HELAC-NLO (10)
  - pp $\longrightarrow W^+W^-b\bar{b}$ — Denner-Dittmaier (10) / HELAC-NLO (10)
  - $e^+e^- \longrightarrow 5$ jets — Rocket (10)
  - pp $\longrightarrow W^+W^+jj$ — Rocket (10)
  - pp $\longrightarrow Z(\gamma)\,/\,W+4$ jets — Blackhat (11)
  - pp $\longrightarrow b\bar{b}b\bar{b}$ — Golem / Samurai (11)
  - pp $\longrightarrow W^+W^-jj$ — Rocket (11) / GoSam (12)
  - pp $\longrightarrow 4$ jets — Blackhat (11)

Motivation

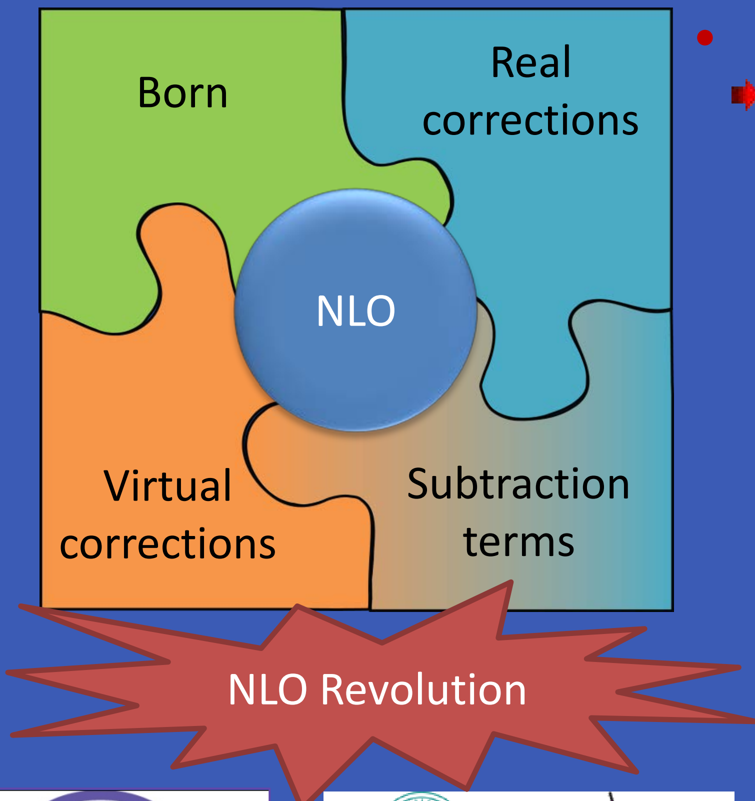
- Progresses in NLO calculation:
  - $\mathrm{pp} \longrightarrow W + 3\ \mathrm{jets}$ Blackhat (09) / Rocket (09)
  - $\mathrm{pp} \longrightarrow t\bar{t}b\bar{b}$ Denner-Dittmaier (09) / HELAC-NLO (09)
  - $\mathrm{pp} \longrightarrow Z(\gamma) + 3\ \mathrm{jets}$ Blackhat (10)
  - $\mathrm{pp} \longrightarrow t\bar{t}jj$ HELAC-NLO (10)
  - $\mathrm{pp} \longrightarrow W^+W^-b\bar{b}$ Denner-Dittmaier (10) / HELAC-NLO (10)
  - $e^+e^- \longrightarrow 5\ \mathrm{jets}$ Rocket (10)
  - $\mathrm{pp} \longrightarrow W^+W^+jj$ Rocket (10)
  - $\mathrm{pp} \longrightarrow Z(\gamma)\,/\,W + 4\ \mathrm{jets}$ Blackhat (11)
  - $\mathrm{pp} \longrightarrow b\bar{b}b\bar{b}$ Golem / Samurai (11)
  - $\mathrm{pp} \longrightarrow W^+W^-jj$ Rocket (11) / GoSam (12)
  - $\mathrm{pp} \longrightarrow 4\ \mathrm{jets}$ Blackhat (11)

Key Concept

AUTOMATION

Automation in NLO calculations

- Different ingredients of a NLO calculation have also different levels of automation according to their complexity:



- Virtual corrections
  - Automatized recently:
    - FEYNARTS/FORMCALC/LOOPTOOLS (public) [Hahn et al.]
    - HELAC-NLO (public) [Bevilacqua, Czakon, van Hameren, Papadopoulos, Pittau, Worek, 11]
    - MadLoop [Hirschi,Frederix,Frixione,Garzelli, Maltoni,Pittau ,11]
    - OpenLoops [Cascioli, Maierhöfer,Pozzorini , 12]
    - GoSam (public) [Cullen, Greiner, Heinrich, GL, Mastrolia, Ossola, Reiter, Tramontano, 11]

Dedicated programs also involve high level of automation:
Denner-Dittmaier et al., VBFNLO (public), MCFM (public), NGLUON (public), BLACKHAT, ROCKET.

NLO evolution

- Evolution from collection of pre-coded processes…

… to generation of full NLO processes by the user "on the fly"!

- Possible thanks to pioneering works:
  - improvements on the computation of tensor integrals,
    [Binoth et al. GOLEM95; Denner, Dittmaier et al.]
  - application of unitarity to the computation of the one loop amplitudes,
    [Bern, Dixon, Kosower; Britto, Cachazo, Feng]
  - reduction at the integrand level.
    [Ossola, Papadopoulos, Pittau; Ellis, Giele, Kunszt, Melnikov]

- Automation allows
  - Self-organization / Process-independent framework / Avoid human mistakes / Focus on Pheno

The GoSam Project: phylosophy

GoSam

Golem (General One Loop Evaluator of Matrix elements)

Samurai (Scattering Amplitudes from Unitarity based Reduction At Integrand level)

An automated amplitude generation based on Feynman diagrams

- Based upon:
  - Algebraic generation of D-dimensional integrands via Feynman diagrams
  - Reduction at the integrand level via D-dimensional extension of the OPP method
  - Generation on the fly of the full rational term

The GoSam Project: goals

- Main targets:
  - Provide an automated tool for stable evaluation of one-loop matrix elements
  - Be general and model independent (QCD, EW, MSSM, …)
  - Interface with existing tools (MadEvent, Sherpa, POWHEG BOX, …)
  - Build upon open source tools only (next slide)
  - Support open standards (for interfacing)

The GoSam Project: the codes

GoSam Project

GoSam: Python package to write code (fortran95)

Code generation

- Diagram generation:

  QGRAF [Nogueira 92]
- Algebra:

  FORM [Vermaseren 91]

  SPINNEY [Cullen, Koch-Janusz, Reiter 10]
- Code generator:

  HAGGIES [Reiter 09]

Yellow codes distributed separately

Generated code execution

- Loop integral reduction:

  SAMURAI [Mastrolia, Ossola, Reiter, Tramontano 10]

  GOLEM95 [Binoth, Cullen, Guillet, Heinrich, Pilon, Reiter 08]

  PJFRY [Yundin]
- Scalar integral evaluation:

  AVHOLO [van Hameren]

  QCDLOOP [Ellis, Zanderighi]

  GOLEM95C [Cullen, Guillet, Heinrich, Kleinschmidt, Pilon, Reiter, Rodgers 11]

All codes in gosam-contrib package

3-Steps to the Loop Amplitude

GOSAM

Code set-up

- Python program writes code files from template
  - QGRAPH writes structure of diagrams

Code creation and compilation

- Fortran code is compiled and amplitudes are written
  - FORM & SPINNEY read and manipulates amplitudes
  - HAGGIES optimizes expressions and writes fortran code

Code execution

- Fortran code computes amplitudes
  - SAMURAI/GOLEM95/… called to reduce integrals
  - AVHOLO/QCDLOOP/… called to evaluate scalar integrals

Reduction methods

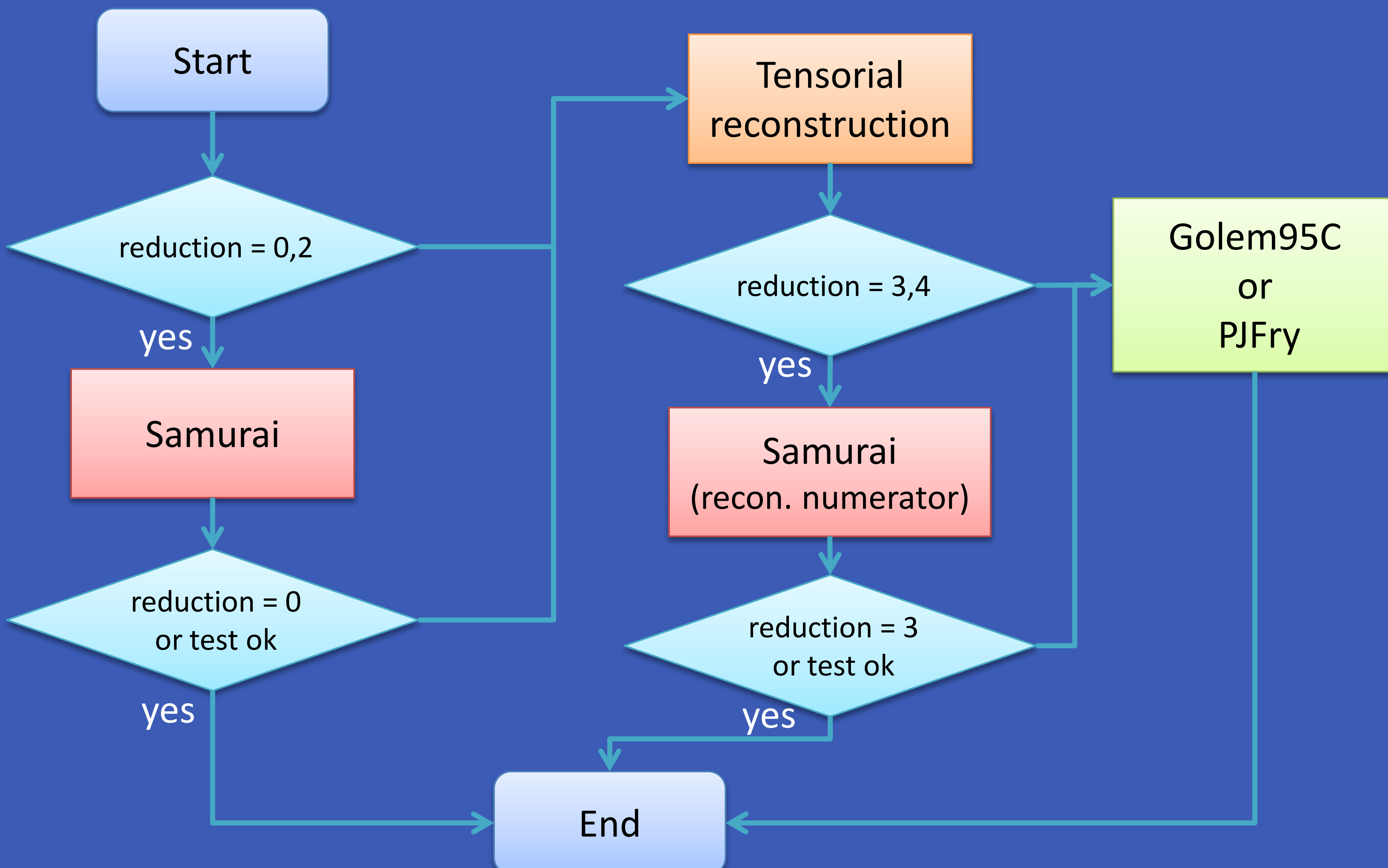
- SAMURAI
  [Mastrolia, Ossola, Reiter, Tramontano 10]

Reduction method can be choosen at runtime

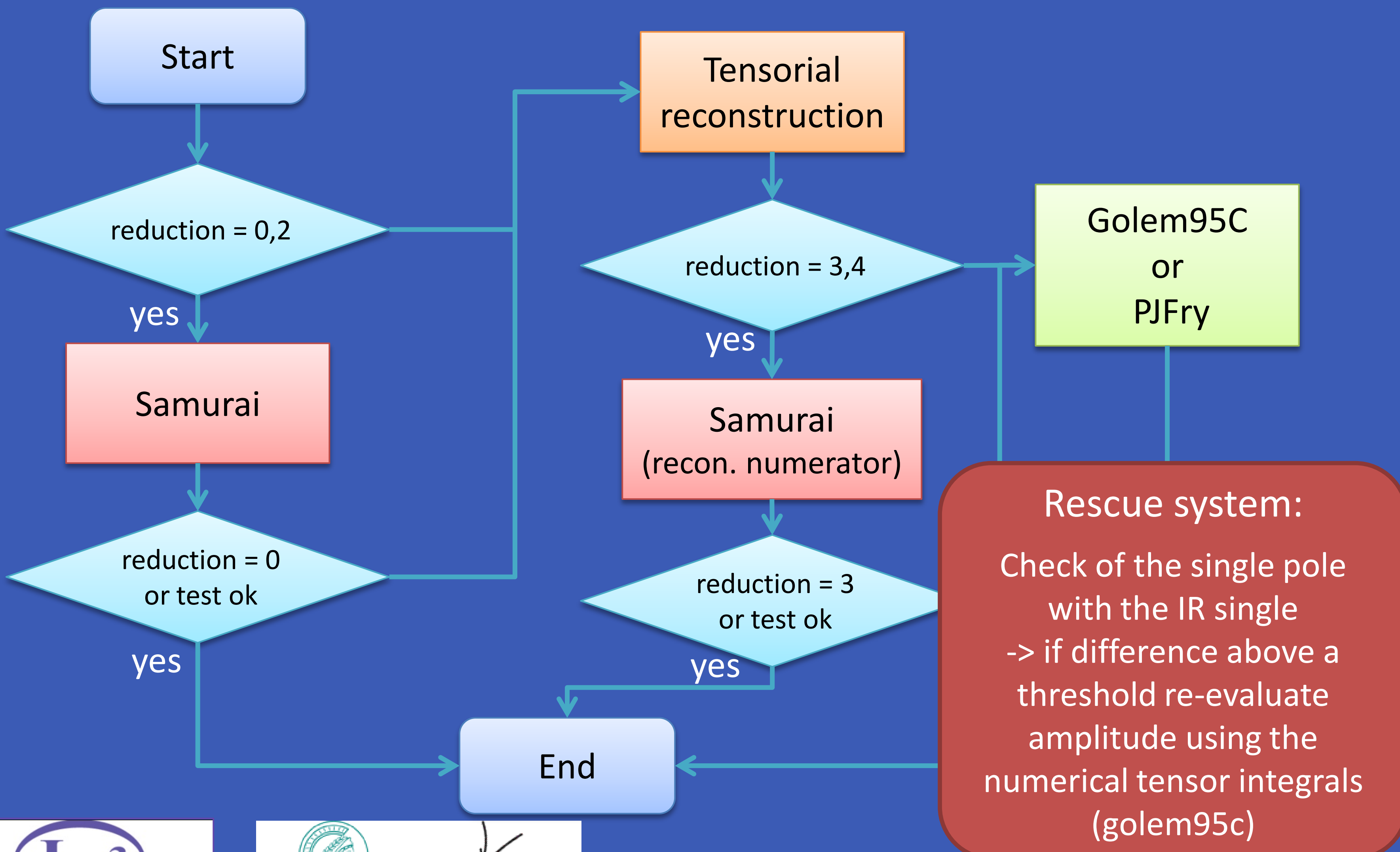
- Tensorial integrand-level reconstruction
  [Heinrich, Ossola, Reiter, Tramontano 10]

  with
  - GOLEM95C [Binoth, Cullen, Guillet, Heinrich, Kleinschmidt, Pilon, Reiter, Rodgers 11]
  - SAMURAI [Mastrolia, Ossola, Reiter, Tramontano 10]
  - PJFry [Yundin]

Reduction: strategies



Reduction: strategies



A Walk through GoSam…

- GoSam as a standalone code

- Interfacing with an external Monte Carlo program:
  - The BLHA-interface [Comput.Phys.Commun. 181 (2010) 1612-1622, arXiv:1001.1307 [hep-ph]]

    Sherpa | Powheg Box | …
  - An example with Sherpa/Powheg Box
  - The GoSam+Sherpa process packages

GoSam standalone: input card

- Preparation of the input card “myprocess.rc“:

```
#!/bin/env /mt/home/luisonig/bin/gosam.py

process_name=ttH
process_path=./ttH_virtual

#### physics options  #####

in=g,g                  # accepts also PDG codes
out=H,t,t~
order=gs, 2, 4

model=smdiag
model.options= masses: mT mH, width: none, \
    alpha: 0.0072973525376, mZ: 92.0584, mW: 80.376, \
    mT: 172.6, mH: 130.0, Nf:5, Nfgen:1

zero=mU,mD,mC,mS,mB,wT,wB,wW,wZ,wH
one=gs,e
symmetries=family,generation
helicities=[+-][+-]0[+-][+-]

qgraf.options=onshell,notadpole,nosnail
qgraf.verbatim=true=iprop[D,S,C,B, 0, 0]
```

Look for example at Higgs-top-antitop production

Models can be imported from FeynRules or LanHEP

Specify which parameters should be set ALGEBRAICALLY to zero or one

Options on helicity and loop diagrams

GoSam standalone: input card

- Preparation of the input card “myprocess.rc“ (continued):

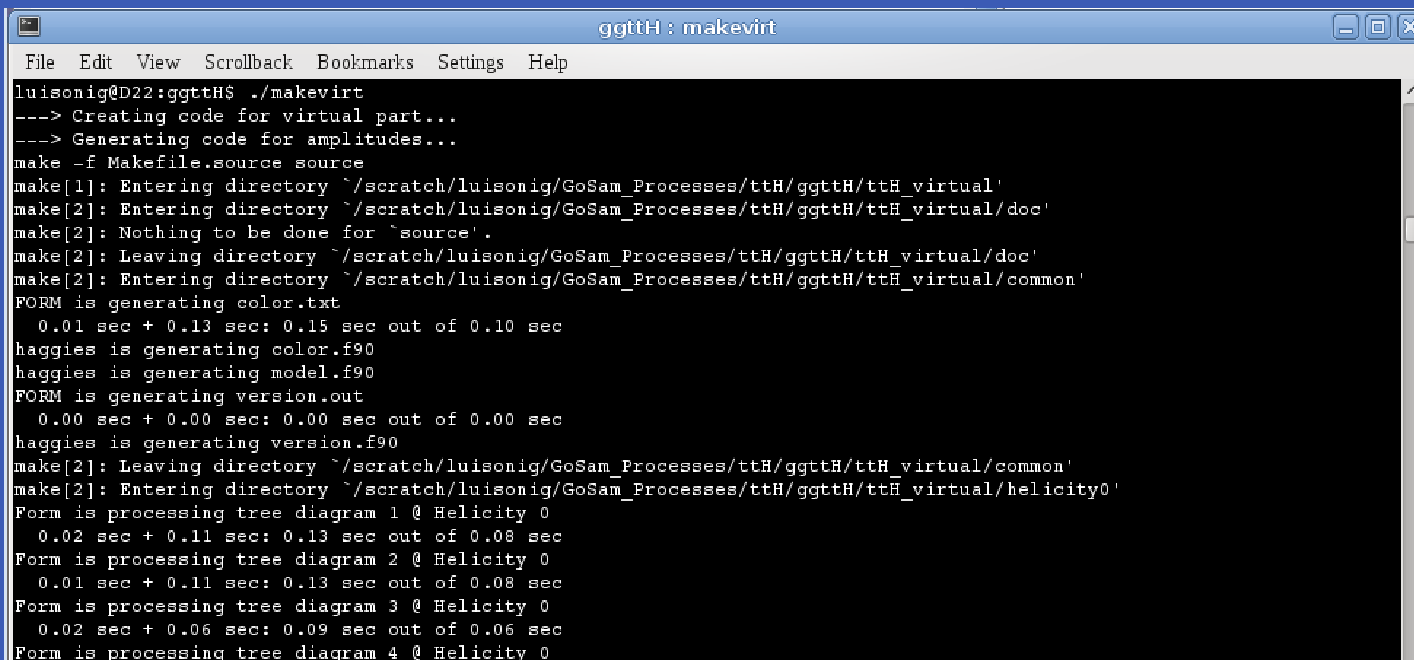
```
#### program options ####

extensions=samurai, golem95, dred

# abbrev.level=helicity  # group , diagram
# abbrev.limit=0

form.bin=tform
form.tempdir=/tmp
fc.bin=gfortran -O2

samurai.fcflags=-I${HOME}/include/samurai
samurai.ldflags=-L${HOME}/lib/ -lqcdloop -lavh_olo \
                -lsamurai
samurai.version=2.1.1

golem95.fcflags=-I${HOME}/include/golem95
golem95.ldflags=-L${HOME}/lib/ -lgolem
```

Several other extension and options available.
For further details check our user manual:
http://www.hepforge.org/archive/gosam/gosam-1.0.pdf

GoSam standalone: generation/compilation

- Generate code and compile

$ gosam.py myprocess.rc	python code generates fortran95 code
$ make source	Form & Haggies process diagrams to write code
$ make compile	fortran95 code in compiled



GoSam standalone: documentation

- Check produced code with automatic generated documentation before the full generation/run
  - Documentation contains information about
    - the generated helicities
    - the colour basis
  - Loop diagrams are grouped into sets of diagrams which share loop propagators

GoSam 1.0: $gg \to Ht\bar{t}$

luisonig

2012-02-19 (22:10:59)

Abstract

This process consists of 8 tree-level diagrams and 160 NLO diagrams. Golem has identified 15 groups of NLO diagrams by analyzing their one-loop integrals.

Index	1	2	3	4	5
0	−	−	0	−	−
1	−	−	0	−	+
2	−	−	0	+	−
3	−	−	0	+	+
4	−	+	0	−	−
5	−	+	0	−	+
6	−	+	0	+	−
7	−	+	0	+	+
$8 \to 4$	+	−	0	−	−
$9 \to 5$	+	−	0	−	+
$10 \to 6$	+	−	0	+	−
$11 \to 7$	+	−	0	+	+
12	+	+	0	−	−
13	+	+	0	−	+
14	+	+	0	+	−
15	+	+	0	+	+

GoSam standalone: documentation

5.4 Group 3 (5-Point)

General Information

The maximum effective rank in this group is 4.

$$
\begin{aligned}
r_1 &= -k_2 + k_5, \quad m_1 = m_t\\
r_2 &= -k_2\\
r_3 &= 0\\
r_4 &= -k_4, \quad m_4 = m_t\\
r_5 &= -k_3 - k_4, \quad m_5 = m_t
\end{aligned}
$$

$$
S = \begin{pmatrix}
S_{1,1} & 0 & S_{1,3} & S_{1,4} & S_{1,5}\\
0 & 0 & 0 & S_{2,4} & S_{2,5}\\
S_{3,1} & 0 & 0 & 0 & S_{3,5}\\
S_{4,1} & S_{4,2} & 0 & S_{4,4} & S_{4,5}\\
S_{5,1} & S_{5,2} & S_{5,3} & S_{5,4} & S_{5,5}
\end{pmatrix}
$$

$$
\begin{aligned}
S_{1,1} &= -2m_t^2\\
S_{1,3} &= -s_{51} + s_{34} - s_{12}\\
S_{1,4} &= -2m_t^2 + s_{45} + m_H^2 - s_{23} - s_{12}\\
S_{1,5} &= -2m_t^2\\
S_{2,4} &= s_{51} - s_{23} - s_{34} + m_H^2\\
S_{2,5} &= s_{51} - m_t^2\\
S_{3,5} &= -m_t^2 + s_{34}\\
S_{4,4} &= -2m_t^2\\
S_{4,5} &= -2m_t^2 + m_H^2\\
S_{5,5} &= -2m_t^2
\end{aligned}
$$

Loop diagrams are grouped into sets of diagrams which share loop-propagators. A loop integral can be written as

$$
\int \frac{\mathrm{d}^n k}{i\pi^{\frac{n}{2}}} \frac{\mathcal{N}(q)}{\prod_{j=1}^{N} \left[(k + r_j)^2 - m_j^2 + i m_j \Gamma_j + i\delta\right]} \qquad (16)
$$

For each group we list r_j, m_j and Γ_j. For m_j and Γ_j only non-vanishing symbols are listed. Furthermore, we give the matrix S which is defined as

$$
S_{\alpha\beta} = (r_\alpha - r_\beta)^2 - m_\alpha^2 + i m_\alpha \Gamma_\alpha - m_\beta^2 + i m_\beta \Gamma_\beta. \qquad (17)
$$

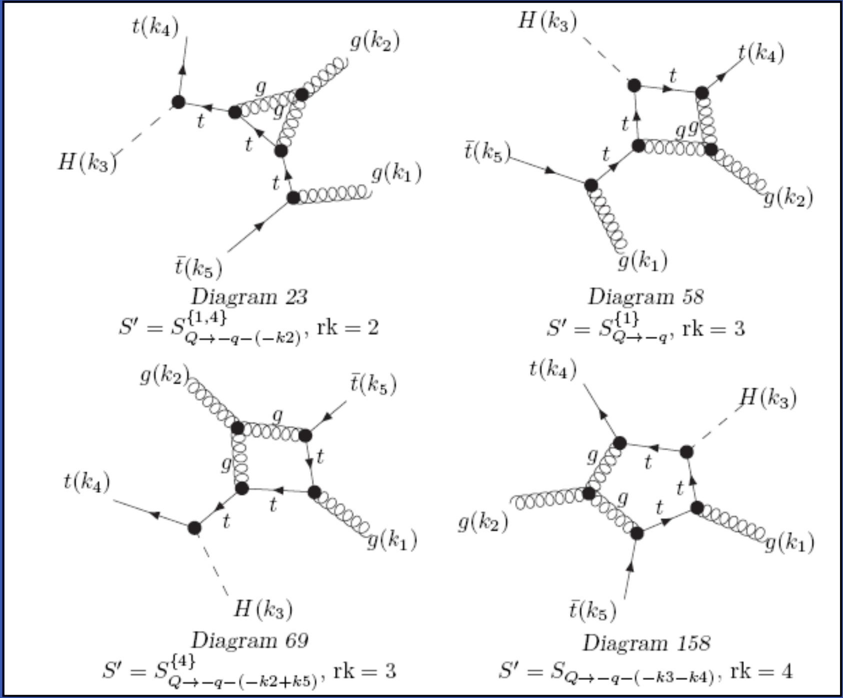


Diagram 23 $S' = S^{\{1,4\}}_{Q\to -q-(-k2)}$, rk = 2

Diagram 58 $S' = S^{\{1\}}_{Q\to -q}$, rk = 3

Diagram 69 $S' = S^{\{4\}}_{Q\to -q-(-k2+k5)}$, rk = 3

Diagram 158 $S' = S_{Q\to -q-(-k3-k4)}$, rk = 4

GoSam standalone: code ready to use

- Contributions divided in directories by helicity

ttH_virtual : bash

File Edit View Scrollback Bookmarks Settings Help

```
luisonig@D22:ttH_virtual$ ls
codegen    diagrams-0.hh   diagrams-1.log  helicity1   helicity14  helicity3  helicity6  Makefile.conf    model.hh
common     diagrams-0.log  doc             helicity12  helicity15  helicity4  helicity7  Makefile.source
config.sh  diagrams-1.hh   helicity0       helicity13  helicity2   helicity5  Makefile   matrix
luisonig@D22:ttH_virtual$
```

- Many running options in **common/config.f90**
- Model parameters in **common/model.f90**

matrix : bash

File Edit View Scrollback Bookmarks Settings Help

```
luisonig@D22:matrix$ ls
debug.xml  Makefile      Makefile.source  matrix.f90  test.exe  test.f90~  tth_matrix.mod
ltest.dat  Makefile.dep  matrix.a         matrix.o    test.f90  test.o
luisonig@D22:matrix$
```

Generated code comes with **test.exe** routine which allows to check cancellation of poles with built-in integrated dipoles: **make test.exe**

➢Output of ***test.exe***:

```
#                   LO:  0.6918083437862626E-04
# NLO, finite part:   17.78339386183546
# NLO, single pole:  -9.484483151742308
# NLO, double pole:  -6.000000000000000
# IR,  single pole:  -9.484483151741490
# IR,  double pole:  -5.999999999999999
#  Time/Event [ms]:               280.000
```

Example: $pp \to H t \bar{t}$

Generation time: 1h 20min   Compilation time: 3h 6min   Time for 1 PS point: 280 ms

Machine: Intel Core Quad CPU Q6600 @ 2.4 GHz / 6 GB RAM

Process generated in DRED and converted to CDR at runtime

	E	p_x	p_y	p_z
u/g	250.0	0.0	0.0	250.0
$\bar{u}/g$	250.0	0.0	0.0	-250.0
H	136.35582793693018	15.133871809486299	27.986733991031045	26.088703626953386
t	181.47665951104506	20.889486679044587	-50.105625289561424	14.002628607367491
$\bar{t}$	182.16751255202476	-36.023358488530903	22.118891298530357	-40.091332234320859

parameters			
$\sqrt{s}$	500.0	N_f	5
μ	m_t	$N_{f,h}$	1
m_t	172.6	α_s	0.107639510785814 5
m_H	130	v	246.21835258713082

result $gg \to t\bar{t}H$		
	GoSam	Ref. [39]
$a_0 \cdot 10^5$	6.127399805961155	6.127400074872043
c_0/a_0	9.006680638719660	9.006680836410272
c_{-1}/a_0	2.986347664537282	2.9863477301662056
c_{-2}/a_0	−6.000000000000004	−6.000000131659877

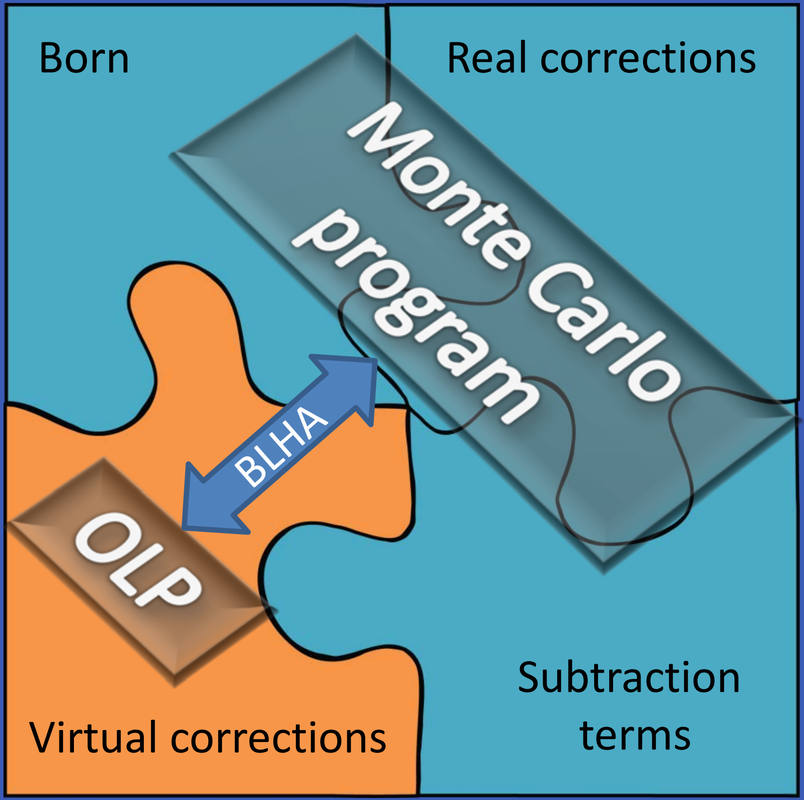
Comparison with MadLoop [Hirschi,Frederix,Frixione,Garzelli, Maltoni,Pittau 11]

GoSam: further tested calculations

- $q\bar{q} \longrightarrow b\bar{b}b\bar{b}$
- $g\bar{g} \longrightarrow b\bar{b}b\bar{b}$
- $q\bar{q} \longrightarrow t\bar{t}b\bar{b}$
- $g\bar{g} \longrightarrow t\bar{t}b\bar{b}$
- $u\bar{d} \longrightarrow W^{+}\,ggg$
- $u\bar{u} \longrightarrow H\,t\bar{t}$
- $g\bar{g} \longrightarrow H\,t\bar{t}$
- $u\bar{d} \longrightarrow W + s\bar{s} \longrightarrow e^{+}\nu_{e}s\bar{s}$
- $u\bar{d} \longrightarrow W + gg \longrightarrow e^{+}\nu_{e}gg$
- $d\bar{d} \longrightarrow Z\,gg \longrightarrow e^{+}e^{-}gg$
- $u\bar{d} \longrightarrow W + b\bar{b} \longrightarrow e^{+}\nu_{e}b\bar{b}$ with massive b‘s
- $u\bar{d} \longrightarrow W + g \longrightarrow e^{+}\nu_{e}g$ EW corrections
- $e^{+}e^{-} \longrightarrow Z \longrightarrow d\bar{d}g$
- $e^{+}e^{-} \longrightarrow Z \longrightarrow b\bar{b}g$ with massive b‘s
- $\gamma\gamma \longrightarrow \gamma\gamma\gamma\gamma$
- $u\bar{d} \longrightarrow W^{+}W^{+}\,s\bar{c} \longrightarrow e^{+}\nu_{e}\mu^{+}\nu_{\mu}s\bar{c}$
- $u\bar{u} \longrightarrow W^{+}W^{+}\,c\bar{c} \longrightarrow e^{+}\nu_{e}\mu^{+}\nu_{\mu}c\bar{c}$
- $u\bar{d} \longrightarrow W^{+}W^{+}\,\bar{s}c \longrightarrow e^{+}\nu_{e}\mu^{+}\nu_{\mu}\bar{s}c$
- plus a large number of 2 to 2 processes

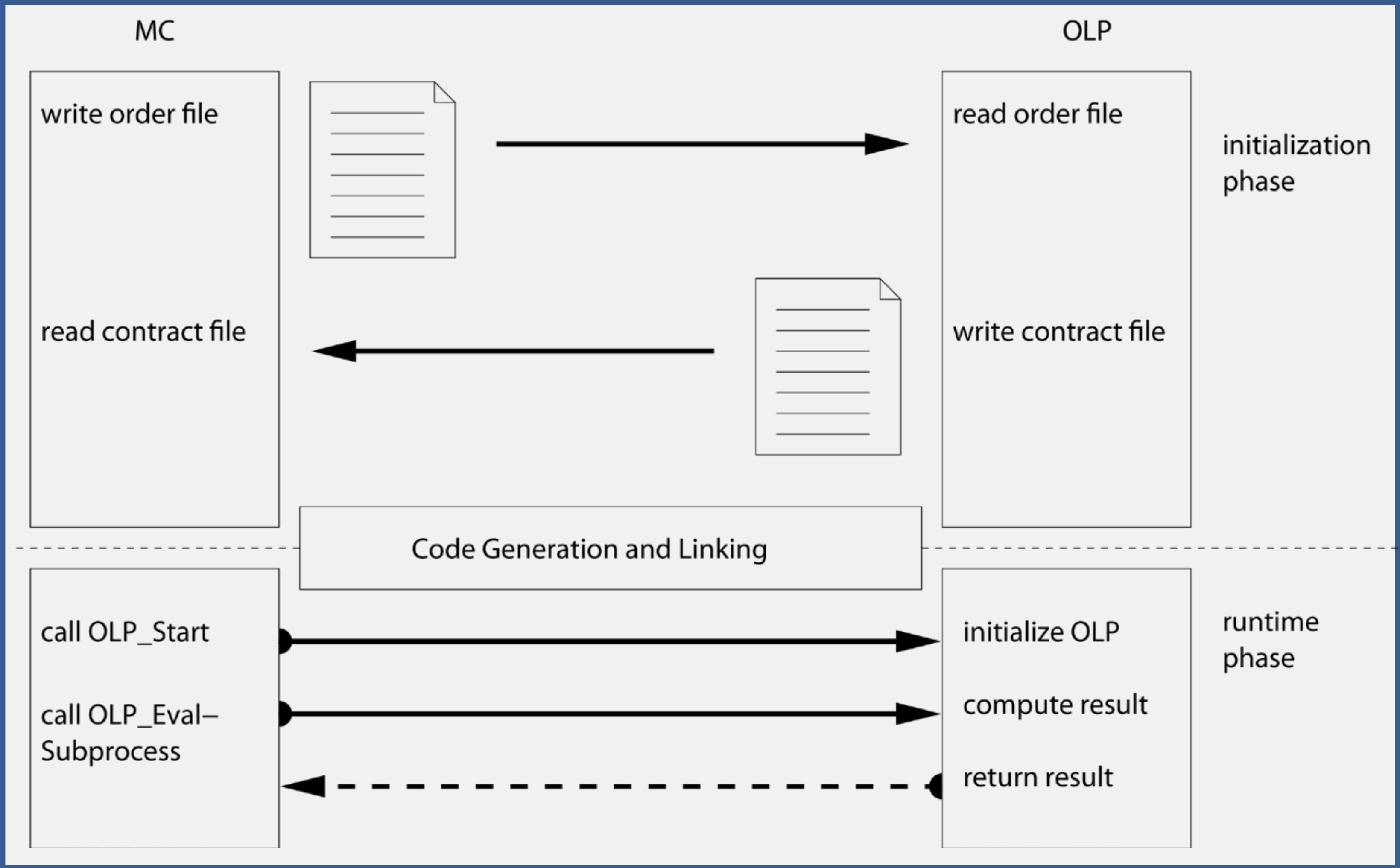
GoSam: interface with MC

- GoSam supports the Binoth-Les-Houches-Accord (BLHA) standards to interface with Monte Carlo generators:
  - Monte Carlo program:
    Born / real corr. / sub. terms
  - One-loop Program (OLP):
    virtual corr.
  - Pre-runtime comunication via “order“ and “contract“ files
  - At runtime:
    - **`OLP_Start()`**
    - **`OLP_EvalSubProcess()`**



[arXiv:1001.1307 [hep-ph]]

BLHA-interface: order & contract



In practice: GoSam+ Sherpa

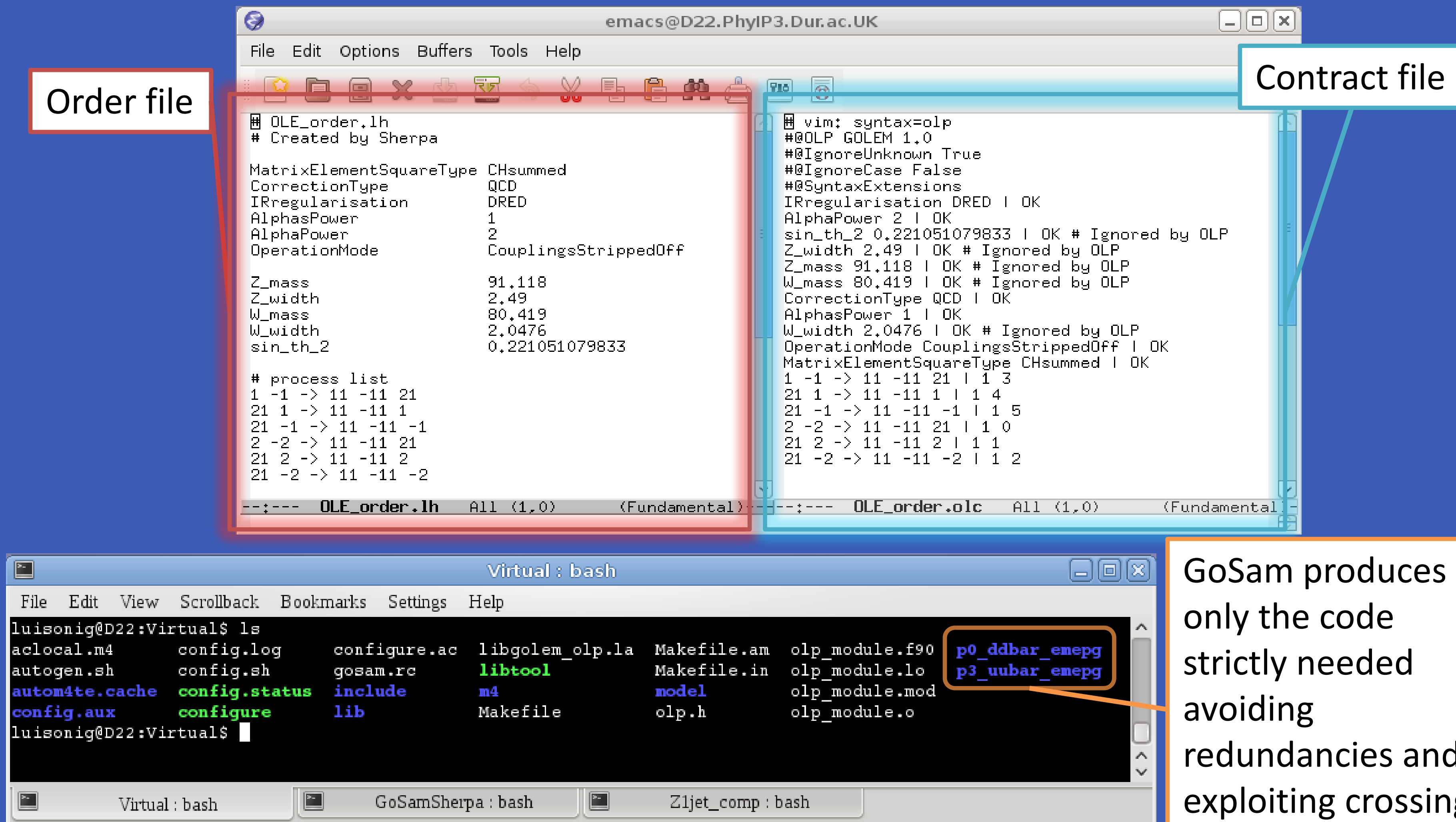
[In collaboration with M.Schonherr]

- Few steps needed to compute e.g. Z+1 jet @NLO:
  - Prepare Sherpa card according to your need and run it once
    - The "order" file and the necessary tree-level code is generated
  - Run GoSam feeding the "order" file and a configuration file with further needed inputs (paths / filtering options / …)
  - After the virtual code is set up, generate and compile it with **`configure / make / make install`**
  - The produced library **`libgolem_olp.so`** must be added to the **`SHERPA_LDADD`** option in the Sherpa card

➔ HAVE FUN WITH PHENOMENOLOGY

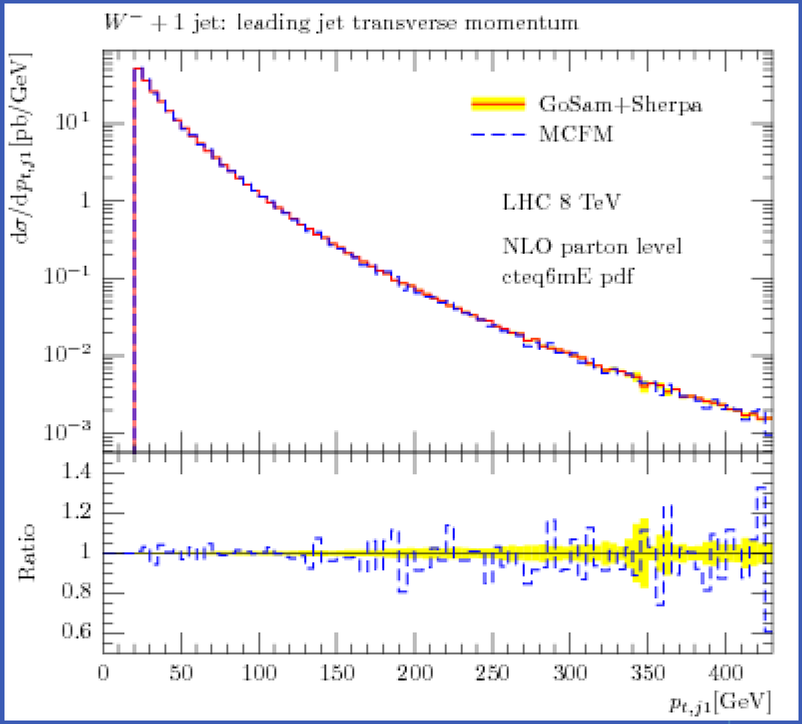
High level of automation and optimization in the generated code

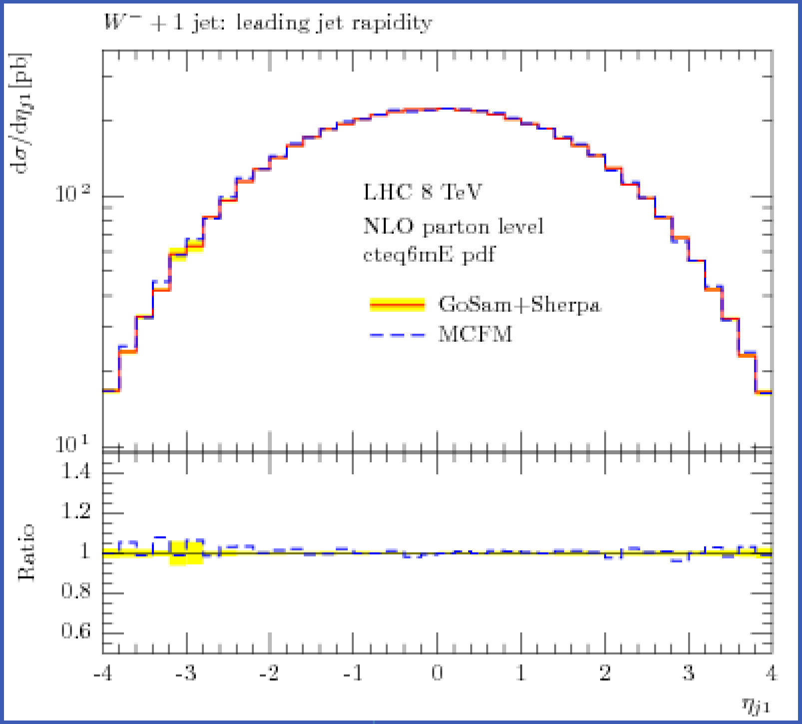
In practice: GoSam+ Sherpa



GoSam produces only the code strictly needed avoiding redundancies and exploiting crossing-symmetry

GoSam+Sherpa vs MCFM: W+1 jet





TIMINGS:

Set-up

Virtual: < 10 sec

Generation & Compilation

Virtual: < 2 min

Running

Born : ~10 min Real : ~ 4h 20 min Virtual: ~ 1h 10 min

MCFM:

ncalls1/2: 100'000

itmx1/2: 10

time: 1h 18 min

Machine Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

NUMBER OF EVENTS:

Born : 5'000'000 x 5

Real : 50'000'000 x 5

Virtual: 5'000'000 x 5

PHYSICS:

LHC 8 TeV

Cuts

pt_jet > 20 GeV

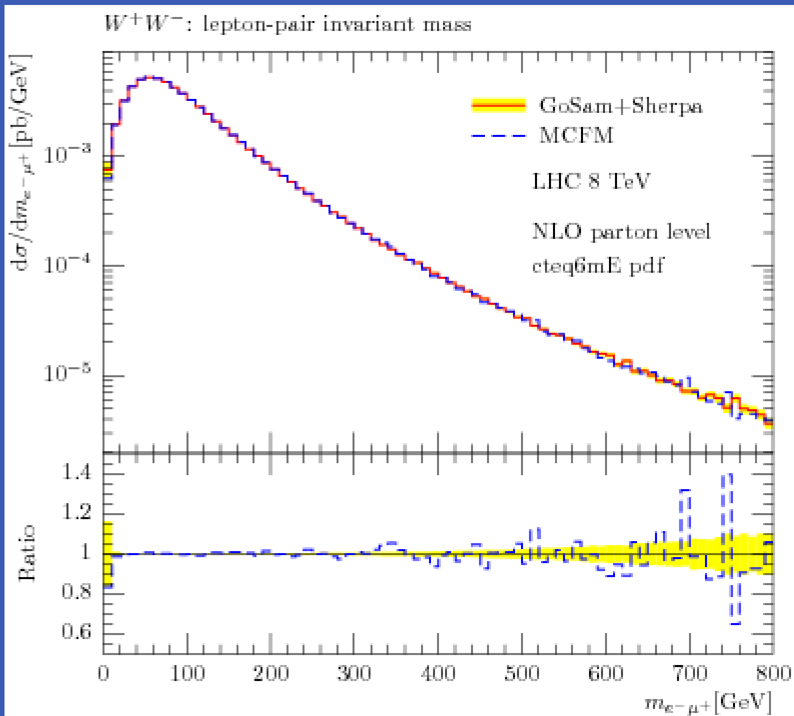
eta_jet < 4.0

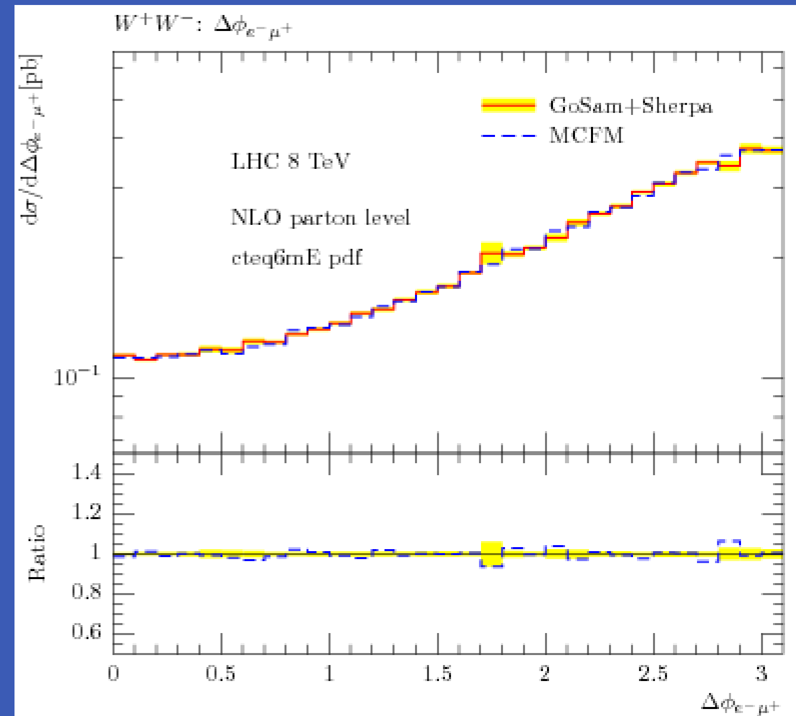
kt_alg, R=0.7

Scale H_T

PDFs cteq6mE.LHgrid

GoSam+Sherpa vs MCFM: $W^+ + W^-$





TIMINGS:
Set-up
Virtual: < 30 sec
Generation & Compilation
Virtual: <20 min
Running
Born : ~15 min Real : ~ 4h 20 min Virtual: ~ 1h 35 min

Machine Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

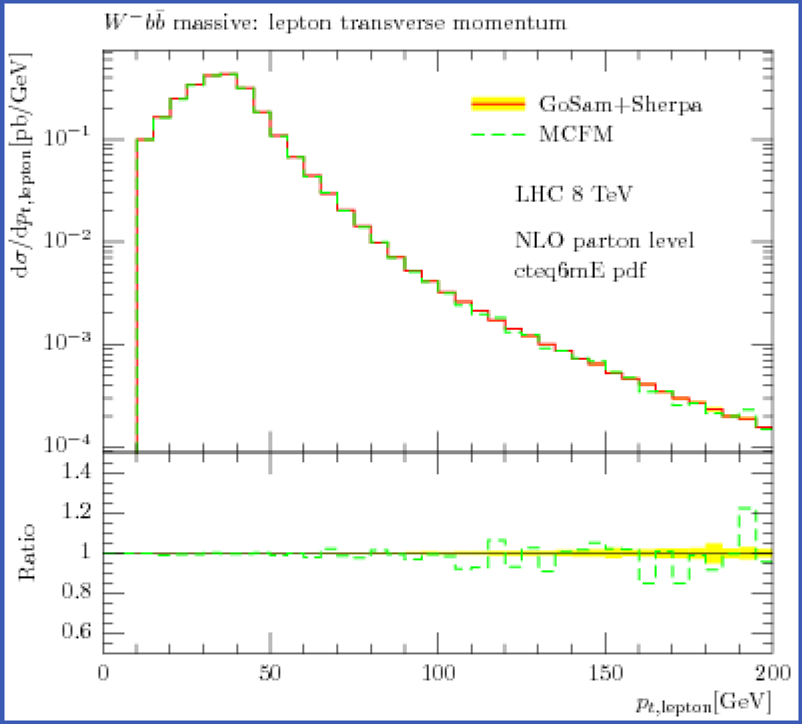
MCFM:
ncalls1/2: 600'000
itmx1/2: 10
time: ~3h

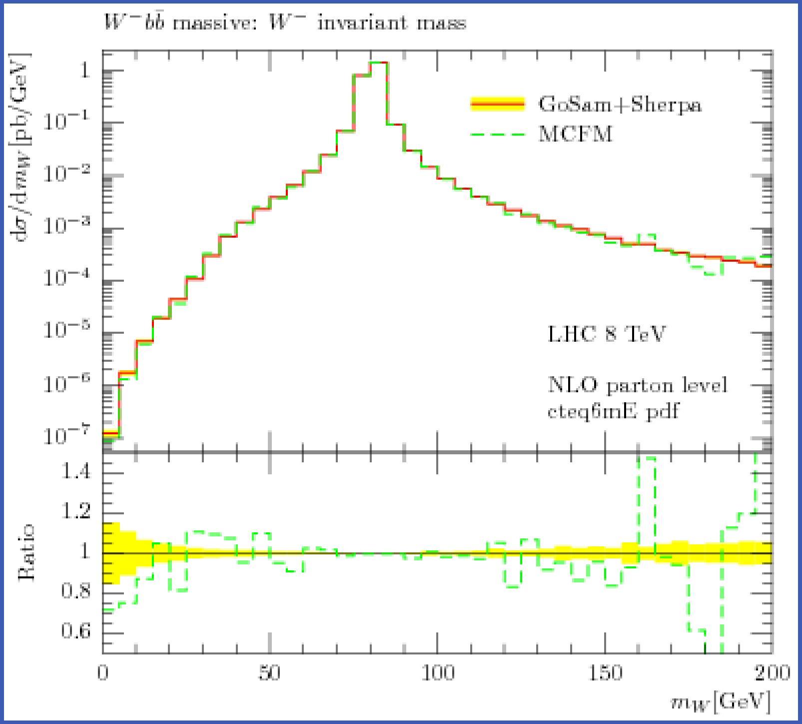
NUMBER OF EVENTS:
Born : 5'000'000 x 5
Real : 50'000'000 x 5
Virtual: 1'000'000 x 5

PHYSICS:
LHC 8 TeV
Cuts no cuts in jets
Scale 80 GeV
PDFs cteq6mE.LHgrid

G.Luisoni, 4th September 2012

GoSam+Sherpa vs MCFM: $W^- + b\bar{b}$ massive





TIMINGS:
Set-up
Virtual: < 10 sec
Generation & Compilation
Virtual: ~ 22 min
Running
Born : ~9 min Real : ~ 5h 20 min Virtual: ~ 11h

Machine Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

MCFM:
ncalls1/2: 100'000
itmx1/2: 10
time: ~7h 10 min

NUMBER OF EVENTS:
Born : 5'000'000 x 5
Real : 50'000'000 x 10
Virtual: 5 '000'000 x 10

PHYSICS:
LHC 8 TeV

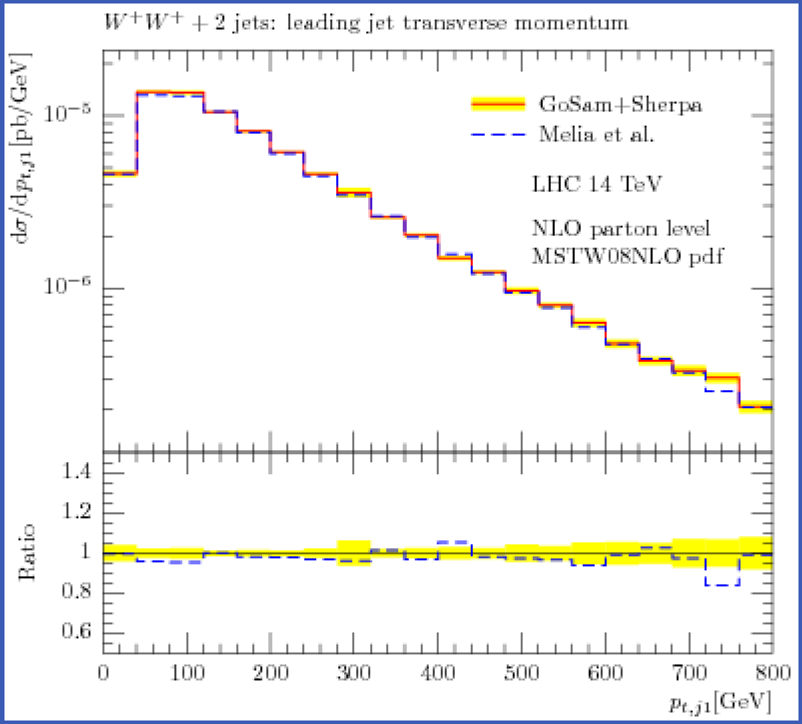
Cuts
pt_miss > 20 GeV
pt_lepton > 10 GeV
inclusive in jets

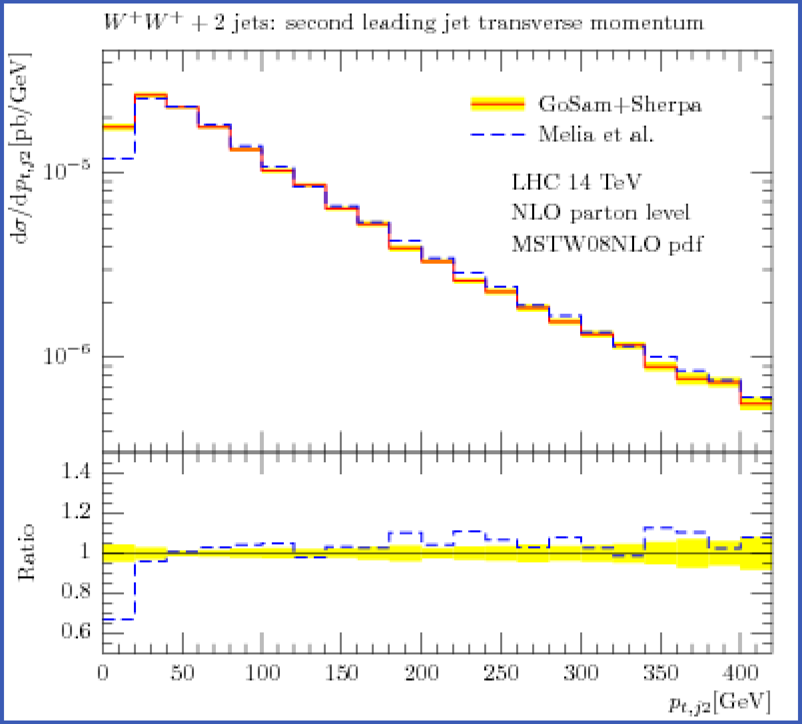
Scale H_T

PDFs cteq6mE.LHgrid

G.Luisoni, 4th September 2012

GoSam+Sherpa vs Melia et al.: W^+W^+ + 2 jets





TIMINGS:
Set-up
Virtual: ~1 min
Generation & Compilation
Virtual: ~ 5h 45 min
Running
Real : ~ 14h 15 min Born+Virtual: ~ 14h 40 min

Machine Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

NUMBER OF EVENTS:
Born : 1'000'000 x 5
Real : 50'000'000 x 5
Virtual: 1'000'000 x 5

Comparison with:
Melia, Melnikov,
Roentsch, Zanderighi;
JHEP 1012 (2010) 053;
[arXiv:1007.5313]

G.Luisoni, 4th September 2012

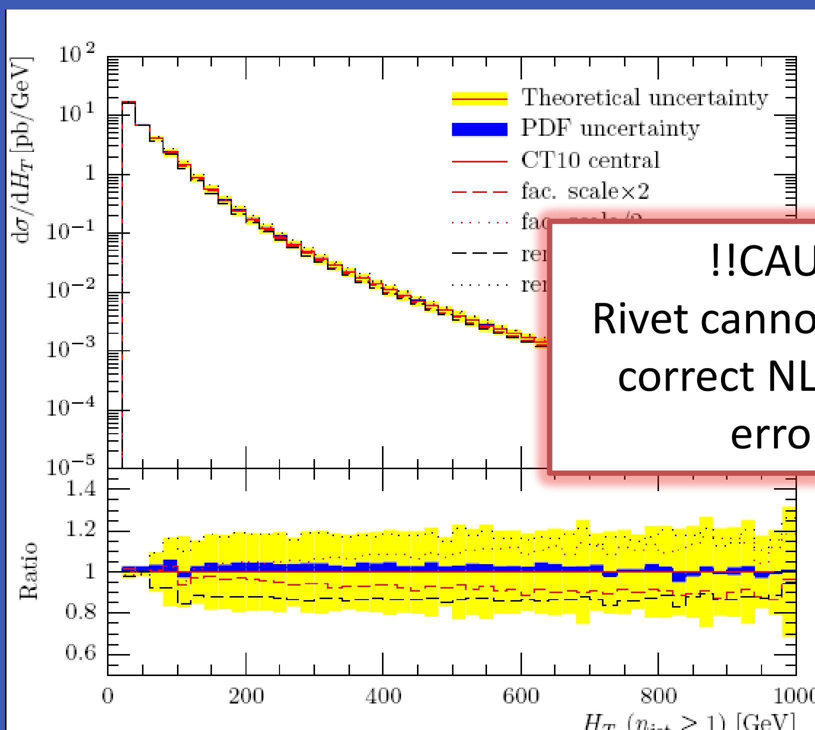
PHYSICS:
LHC 14 TeV

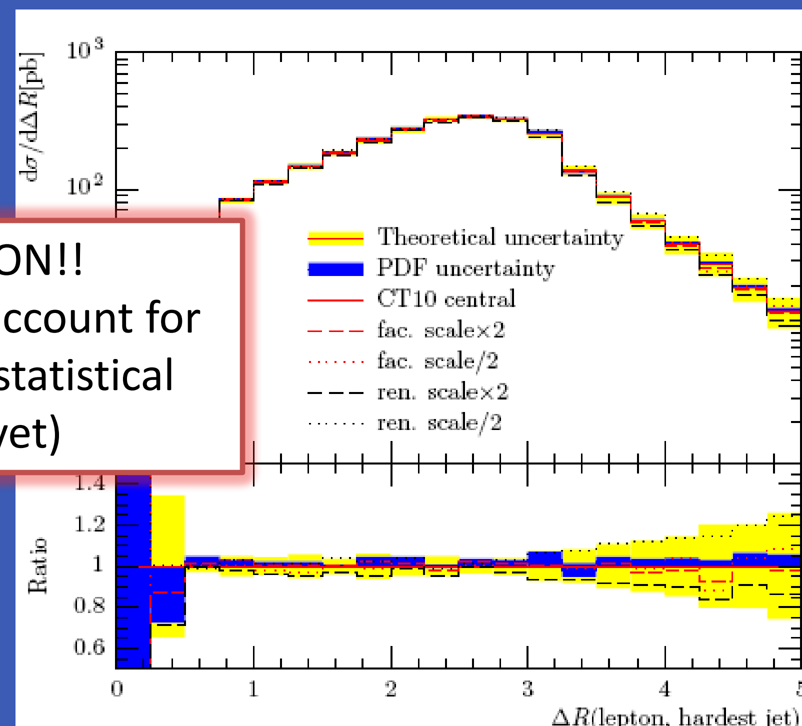
Cuts
pt_lep > 20 GeV
|eta_lep| < 2.4
pt_miss > 30 GeV
antikt_alg, R=0.4

Scale 150 GeV
PDFs MSTW2008nlo.LHgrid

NLO analyses with Rivet

- Easy to perform phenomenological NLO analysis using e.g. GoSam+Sherpa in association with Rivet
  - LH-uncertainty study of W+1 jet [LH2011-proceedings]





!!CAUTION!!
Rivet cannot account for correct NLO statistical error (yet)

GoSam+Sherpa Process Packages

http://gosam.hepforge.org/proc/

Process List:

- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) +\ jet$, wm1jet.tar.gz (437K)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) +\ jet$, wp1jet.tar.gz (431K)
- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) +\ b\bar{b}$, wmbb.tar.gz (772K)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) +\ b\bar{b}$, wpbb.tar.gz (771K)
- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) +\ 2\,jets$, wm2jets.tar.gz (3.49M)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) +\ 2\,jets$, wp2jets.tar.gz (3.46M)
- $pp/p\bar{p} \to W^+(\to \mu^+ + \nu_\mu) + W^-(\to e^- + \bar{\nu}_e)$, wpwm.tar.gz (716K)
- $pp/p\bar{p} \to W^+(\to \mu^+ + \nu_\mu) + W^+(\to e^+ + \nu_e) +\ 2\,jets$, wpwp2jets.tar.gz (3.76M)

DEPENDENCIES

To run the process packages you need the following:

- Sherpa-1.4.0
- GoSam patch for Sherpa-1.4.0: linux, mac
- gosam-contrib-1.0, we recommend to set the installation path using the option `--prefix`.

- Interface with Sherpa 1.4.0 (March 2012) via BLHA-interface (--enable-lhole) with a little additional patch.
- Installation details on the webpage

- Only 3 steps for NLO:
  - download
  - un-tar package
  - run 'makecode' script
- Script for plots is also attached
- Example of interface with Rivet
- Soon possibility to shower

GoSam+Sherpa Process Packages

http://gosam.hepforge.org/proc/

Process List:

- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) + \ jet$, wm1jet.tar.gz (437K)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) + \ jet$, wp1jet.tar.gz (431K)
- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) + \ b\bar{b}$, wmbb.tar.gz (772K)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) + \ b\bar{b}$, wpbb.tar.gz (771K)
- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) + \ 2\,jets$, wm2jets.tar.gz (3.49M)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) + \ 2\,jets$, wp2jets.tar.gz (3.46M)
- $pp/p\bar{p} \to W^+(\to \mu^+ + \nu_\mu) + W^-(\to e^- + \bar{\nu}_e)$, wpwm.tar.gz (716K)
- $pp/p\bar{p} \to W^+(\to \mu^+ + \nu_\mu) + W^+(\to e^+ + \nu_e) + \ 2\,jets$, wpwp2jets.tar.gz (3.76M)

DEPENDENCIES

To run the process packages you need the following:

- Sherpa-1.4.0

wmbb : bash

File Edit View Scrollback Bookmarks Settings Help

```
luisonig@D22:wmbb$ ls
gosam_process_wmbb-1.0.tar.gz  makecode  makeplots  OLE_order.lh  OLE_order.olc  README  Run_LO.dat  Run_NLO.dat  Sherpa_References.tex
luisonig@D22:wmbb$
```

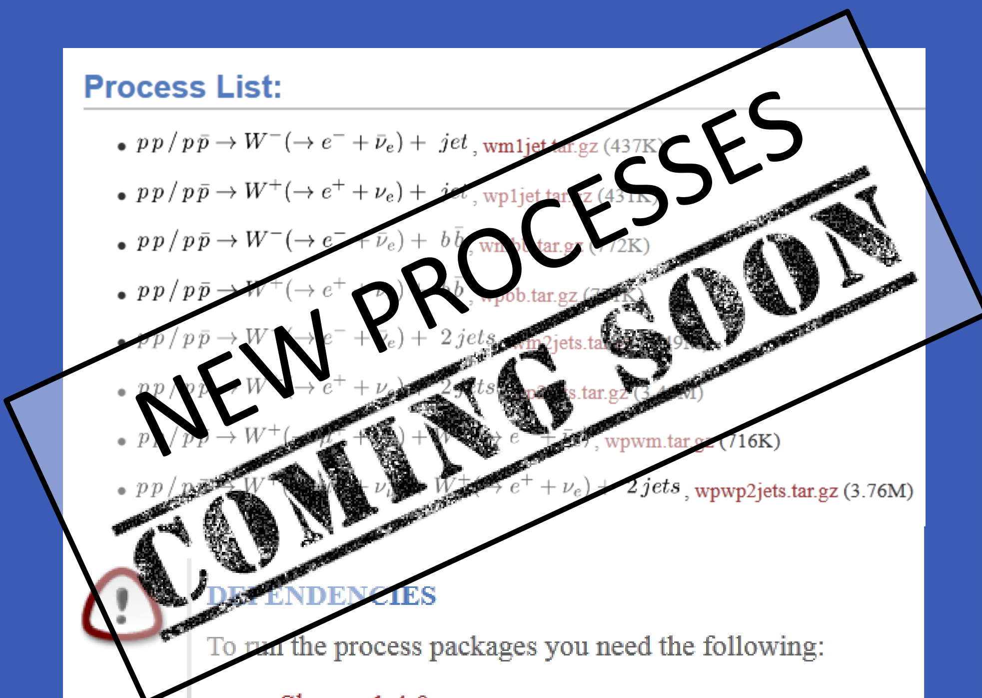
- Interface with Sherpa 1.4.0 (March 2012) via BLHA-interface (--enable-lhole) with a little additional patch.
- Installation details on the webpage

- Only 3 steps for NLO:
  - download
  - un-tar package
  - run ‘makecode‘ script
- Script for plots is also attached
- Example of interface with Rivet
- Soon possibility to shower

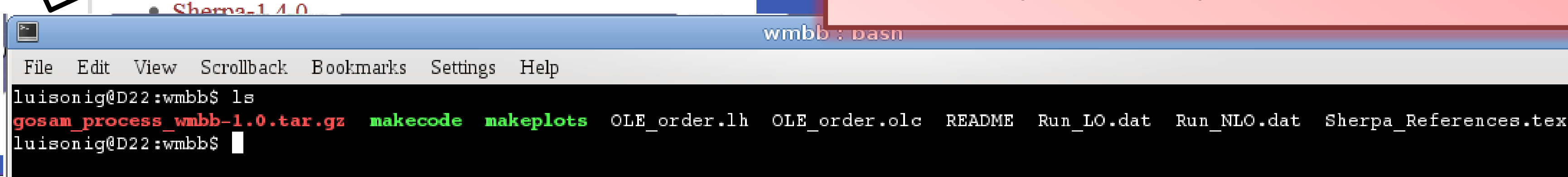
GoSam+Sherpa Process Packages

http://gosam.hepforge.org/proc/

- Interface with Sherpa 1.4.0 (March 2012) via BLHA-interface (--enable-lhole) with a little additional patch.
- Installation details on the webpage



- Only 3 steps for NLO:
  - download
  - un-tar package
  - run ‘makecode‘ script
- Script for plots is also attached
- Example of interface with Rivet
- Soon possibility to shower

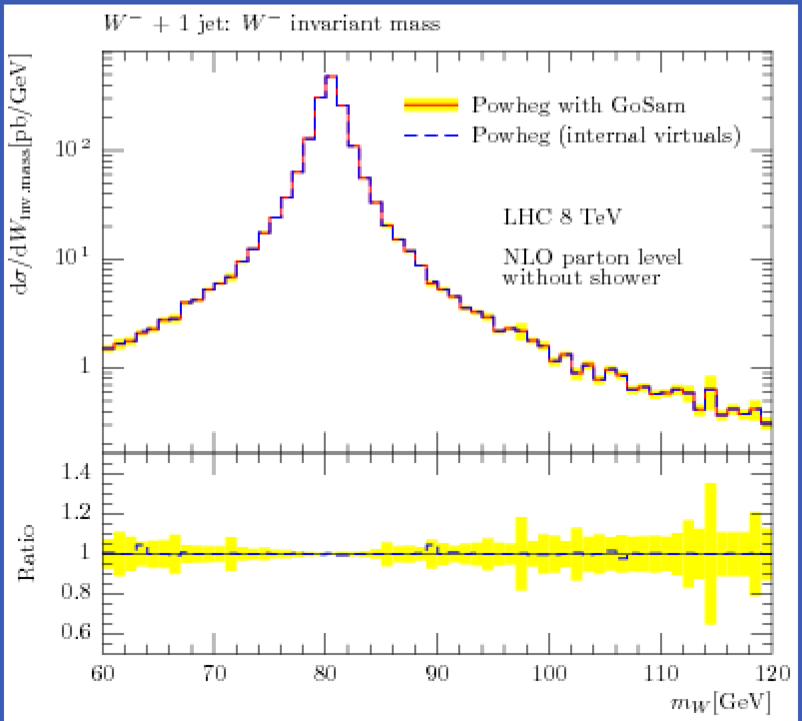


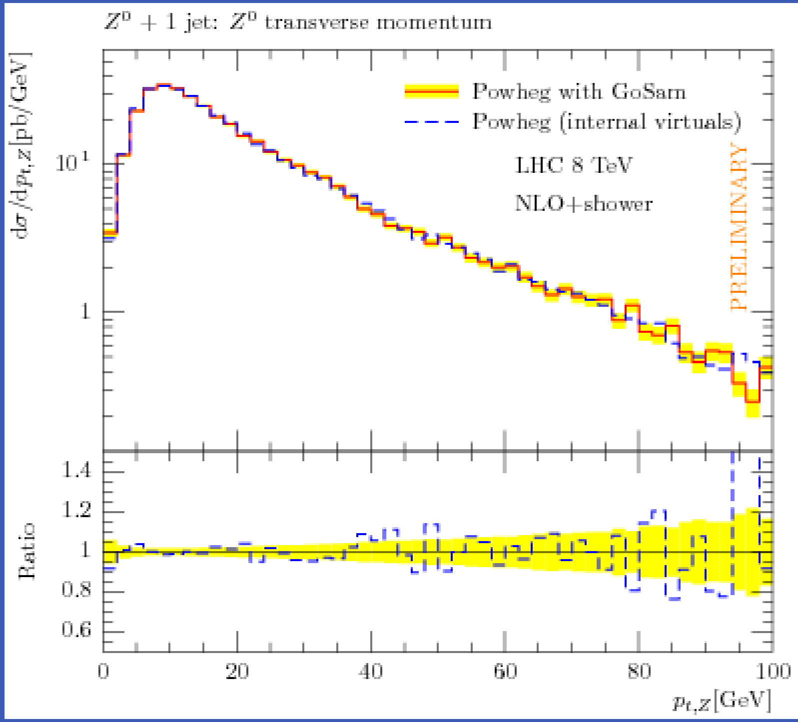
GoSam+Powheg Box

- Powheg Box - GoSam interface developed recently

[In collaboration with C.Oleari and P.Nason]

  - Test examples against existing processes in the Powheg Box:





BSM physics with GoSam

- New models can be added via

  FeynRules (UFO)[Christensen, Duhr]　　LanHEP [Semenov]

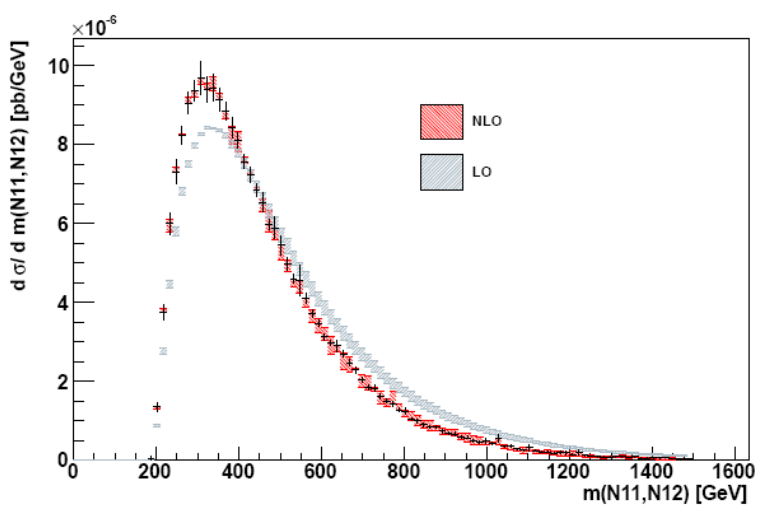
- Allows to compute one-loop corrections also for BSM phenomenology

  - Example: $pp \longrightarrow \chi_0^1 \chi_0^1$ in MSSM

Veto:

pt_jet > 20 GeV
eta_jet < 4.5

Scale　Mz



[Figure by G.Cullen and N.Greiner]

Conclusions and Outlook

- GoSam is a code for the computation of one-loop multi-leg amplitudes
  - Based on Feynman diagrams
  - Uses D-dimensional reduction tecniques
  - Flexible and broadly applicable tool
  - Public
  - Easy to interface with MC event generator to perform full NLO calculations:
    - so far interfaced with:

http://gosam.hepforge.org/

SHERPA

POWHEG BOX

- Possibilities for precision studies using NLO parton-level matched with parton-shower and with hadronization effects just around the corner
  - Possible to steer everything by just editing a single input card

- We look forward to interfacing with other tools and performing NLO analyses for the LHC

Backup slides…

Reduction methods: Samurai [default]

[Mastrolia, Ossola, Reiter, Tramontano 10]

- OPP reduction algorithm [Ossola, Papadopoulos, Pittau 07]
- D-dimensional extension [Ellis, Giele, Kunszt, Melnikov 08]
- Coefficient of polynomials via DFT [Mastrolia et al. 08]
- Computation of the full rational term in one go [Internal GoSam algebraic handling]

For any one-loop amplitude:

$$\mathcal{A}_n = \int d^d\bar{q} \frac{\mathcal{N}(\bar{q},\epsilon)}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{n-1}} \quad ; \quad \mathcal{N}(\bar{q},\epsilon) = N_0(\bar{q}) + \epsilon N_1(\bar{q}) + \epsilon^2 N_2(\bar{q})$$

$$\bar{D}_i = (\bar{q}+p_i)^2 - m_i^2 = (q+p_i)^2 - m_i^2 - \mu^2 \quad ; \quad \bar{\not{q}} = \not{q} + \not{\mu} \; ; \quad \bar{q}^2 = q^2 - \mu^2$$

Result of integration can be expressed as linear combination of scalar integrals: boxes, triangles, bubbles, tadpoles and rational terms

Integrals with μ^2 in the numerator

$$\mathcal{A}_n = \sum_{i_0<i_1<i_2<i_3}^{m-1} d(i_0i_1i_2i_3)D_0(i_0i_1i_2i_3) + \sum_{i_0<i_1<i_2}^{m-1} c(i_0i_1i_2)C_0(i_0i_1i_2) + \sum_{i_0<i_1}^{m-1} b(i_0i_1)B_0(i_0i_1) + \sum_{i_0}^{m-1} a(i_0)A_0(i_0) + \mathcal{R}$$

Reduction methods: Tensorial Reconstr.

[Heinrich, Ossola, Reiter, Tramontano 10]

- Tensorial reconstruction convoluted with tensor integrals:

Rewrite numerator function as linear combination of tensors

$$\mathcal{N}(q) = \sum_{r=0}^{R} C_{\mu_1 \ldots \mu_r} q_{\mu_1} \cdots q_{\mu_r}$$

$$C_{\mu_1 \ldots \mu_r} q_{\mu_1} \cdots q_{\mu_r} = \sum_{(i_1,i_2,i_3,i_4) \vdash r} \hat{C}^{(r)}_{i_1\, i_2\, i_3\, i_4} \cdot (q_1)^{i_1} (q_2)^{i_2} (q_3)^{i_3} (q_4)^{i_4}$$

Determine the coefficients by sampling in q_μ in a bottom-up approach

if $q_\mu = (x, y, z, w)$ then $\mathcal{N}(q) = \mathcal{N}(x, y, z, w)$

Level-0 $q = (0,0,0,0)$; $\mathcal{N}(0,0,0,0) \equiv \mathcal{N}^{(0)} = C_0$

Level-1 4 systems, each sampling a monomial depending on one component of q_μ only

$$\mathcal{N}^{(1)}(q) \equiv \mathcal{N}(q) - \mathcal{N}^{(0)}$$

$$q = (x,0,0,0) \Rightarrow \mathcal{N}^{(1)}(x,0,0,0) \equiv x\, C_1 + x^2\, C_{11} + \ldots + x^R\, C_{\underbrace{11\ldots1}_{R \text{ times}}}$$

$$q = (0,y,0,0) \Rightarrow \mathcal{N}^{(1)}(0,y,0,0) \equiv y\, C_2 + y^2\, C_{22} + \ldots + y^R\, C_{\underbrace{22\ldots2}_{R \text{ times}}}$$

Allows to avoid numerical instabilities due to vanishing Gram determinants

Derive & Numpolvec

- The latest version of GoSam also implements two new features to improve speed and precision:
  - **`derive`**: computes the numerator by expanding in a Taylor series

$$\mathcal{N}(\hat{q}) = \mathcal{N}(0) + \hat{q}^{\mu}\frac{\partial}{\partial \hat{q}_{\mu}}\mathcal{N}(\hat{q})|_{q=0} + \frac{1}{2!}\hat{q}^{\mu}\hat{q}^{\nu}\frac{\partial}{\partial \hat{q}_{\mu}}\frac{\partial}{\partial \hat{q}_{\nu}}\mathcal{N}(\hat{q})|_{q=0} + \dots$$

    one-to-one correspondence between derivatives at $\hat{q} = 0$ and the coefficients of the tensor integrals

  - **`numpolvec`**: uses numerical polarization vectors for external massless gauge bosons
    - This allows to reduce the code by generating only few helicities

OPP

[F.Tramontano 11]

OPP integrand decomposition: 4-dim

- At integrand level the structure is enriched by polynomial terms that integrate to zero (I multiplied with all the propagators)

$$N(q) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d(i_0 i_1 i_2 i_3) + \tilde{d}(q; i_0 i_1 i_2 i_3)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i + \sum_{i_0<i_1<i_2}^{m-1} \left[c(i_0 i_1 i_2) + \tilde{c}(q; i_0 i_1 i_2)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$
$$+ \sum_{i_0<i_1}^{m-1} \left[b(i_0 i_1) + \tilde{b}(q; i_0 i_1)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i + \sum_{i_0}^{m-1} \left[a(i_0) + \tilde{a}(q; i_0)\right] \prod_{i\neq i_0}^{m-1} D_i$$

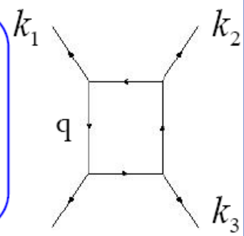
- A choice of q fulfilling 4-ple cut condition: $D_{i0} = D_{i_1} = D_{i_2} = D_{i_3} = 0$ will single out just one polynomial

$$\Delta_{i_0 i_1 i_2 i_3} = \left[d(i_0 i_1 i_2 i_3) + \tilde{d}(q; i_0 i_1 i_2 i_3)\right]$$

$\tilde{d}$ can **only** be of the type $q.p$

where $p = \varepsilon_{\alpha\beta\gamma} k_1^\alpha k_2^\beta k_3^\gamma$

[proof in OPP 2007]



- Once fitted such polynomial we can subtract it from both sides and repeat the game with another multiple cut condition -> recursive solution

- For each phase space point the only requirement for the reduction is the knowledge of the numerical value of the numerator function N for a small set of values of the loop momentum variable, solutions of the multiple cut conditions

OPP

[F.Tramontano 11]

Extension to D-dim

- fix a parametric form for the loop momentum in terms of a linear combination of four known 4-vectors e_i suitably chosen

$$\not{\bar{q}} = \not{q} + \not{\mu} \qquad \bar{q}^2 = q^2 - \mu^2 \qquad q = x_1 e_1 + x_2 e_2 + x_3 e_3 + x_4 e_4$$

the vanishing term (spurious term in the OPP terminology) are then polynomials of x_i and μ^2

- The problem is to fit the coefficients in the polynomials Δ

$$N(\bar{q}) = \sum_{i<<m}^{n-1} \Delta_{ijk\ell m}(\bar{q}) \prod_{h\neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i<<\ell}^{n-1} \Delta_{ijk\ell}(\bar{q}) \prod_{h\neq i,j,k,\ell}^{n-1} \bar{D}_h + \\ + \sum_{i<<k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h\neq i,j,k}^{n-1} \bar{D}_h + \sum_{i<j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h\neq i,j}^{n-1} \bar{D}_h + \sum_{i}^{n-1} \Delta_{i}(\bar{q}) \prod_{h\neq i}^{n-1} \bar{D}_h$$

✓ Example: 3-ple cut residue

$$\Delta_{ijk}(\bar{q}) = c_{3,0}^{(ijk)} \boxed{+ c_{3,7}^{(ijk)}\mu^2} - \left((c_{3,1}^{(ijk)} \boxed{+ c_{3,8}^{(ijk)}\mu^2}) x_4 + (c_{3,4}^{(ijk)} \boxed{+ c_{3,9}^{(ijk)}\mu^2}) x_3 \right) (e_1 \cdot e_2) + \\ + \left(c_{3,2}^{(ijk)} x_4^2 + c_{3,5}^{(ijk)} x_3^2 \right) (e_1 \cdot e_2)^2 - \left(c_{3,3}^{(ijk)} x_4^3 + c_{3,6}^{(ijk)} x_3^3 \right) (e_1 \cdot e_2)^3 \,.$$

✓ with the 3 cut conditions: $D_i = D_j = D_k = 0$ one fixes x_1, x_2 and the product $x_3 x_4$

OPP

[F.Tramontano 11]

Amplitudes & Master Integrals

$$\mathcal{A}_n = \sum_{i<j<k<\ell}^{n-1} \left\{ c_{4,0}^{(ijk\ell)} I_{ijk\ell}^{(d)} + \frac{(d-2)(d-4)}{4} c_{4,4}^{(ijk\ell)} I_{ijk\ell}^{(d+4)} \right\}$$
$$+ \sum_{i<j<k}^{n-1} \left\{ c_{3,0}^{(ijk)} I_{ijk}^{(d)} - \frac{(d-4)}{2} c_{3,7}^{(ijk)} I_{ijk}^{(d+2)} \right\}$$
$$+ \sum_{i<j}^{n-1} \left\{ c_{2,0}^{(ij)} I_{ij}^{(d)} + c_{2,1}^{(ij)} J_{ij}^{(d)} + c_{2,2}^{(ij)} K_{ij}^{(d)} - \frac{(d-4)}{2} c_{2,9}^{(ij)} I_{ij}^{(d+2)} \right\}$$
$$+ \sum_{i}^{n-1} c_{1,0}^{(i)} I_i^{(d)}$$

$$\int d^d\bar{q} \frac{\bar{q}\cdot e_2}{\bar{D}_i \bar{D}_j} = J_{ij}^{(d)}$$
$$\int d^d\bar{q} \frac{(\bar{q}\cdot e_2)^2}{\bar{D}_i \bar{D}_j} = K_{ij}^{(d)}$$

$$d = 4 - 2\varepsilon$$

The sources of rational terms are the integrals with μ^2 powers in the numerator

They are generated by the reduction algorithm(R1), but could also be present ab initio in the numerator function as a consequence of the d-dimensional algebraic manipulations (R2)

$$\int d^d\bar{q} \frac{\mu^2}{\bar{D}_i \bar{D}_j} = -\frac{(d-4)}{2} I_{ij}^{(d+2)}$$
$$\int d^d\bar{q} \frac{\mu^4}{\bar{D}_i \bar{D}_j \bar{D}_k \bar{D}_\ell} = \frac{(d-2)(d-4)}{4} I_{ijk\ell}^{(d+4)}$$
$$\int d^d\bar{q} \frac{\mu^2}{\bar{D}_i \bar{D}_j \bar{D}_k} = -\frac{(d-4)}{2} I_{ijk}^{(d+2)}$$

Rational term

[F.Tramontano 11]

More on the rational terms:

- Treatment strictly related the way the numerator function is furnished
  - Diagramatic approach allows for the classification in two categories:

$$R = R1 + R2$$

- **R1** develops automatically performing the D-dimensional reduction of the tensors spanning the 4-dimensional part of the loop momentum
- **R2** are present in the UV diagrams: bubbles, rank 2 and 3 triangles and rank4 boxes.
- At least two possibilities for R2 automatic computation:
  - for any fixed gauge theory calculate once and for all the contribution from all the diagrams that can generate R2 terms and define a set of tree level Feynman rules that give the R2 contribution for any process: **MadLoop approach**
  - Alternatively: construct the numerator function by implementing (few and universal) algebraic rules to get the R2 term on a diagram by diagram basis: **GoSam approach**

Rational term

[F.Tramontano 11]

GoSam offers different options for the computation of the R2 terms

Thanks to the fact that we generate analytic expressions for the d-dimensional numerator function $\bar{N}(\bar{q})$

- **implicit**: $\mathbf{R}_2$ terms are kept in the numerator and reduced at runtime using the d-dimensional decomposition of the numerator
- **explicit**: $\mathbf{R}_2$ terms are calculated analytically (without entering in the numerical decomposition)
- **only**: only the $\mathbf{R}_2$ term is kept in the final result (this option does not require any additional libraries)
- **off**: all $\mathbf{R}_2$ terms are set to zero

R2 is a gauge dependent quantity

Precision tests

Use the decomposition of the numerator function $N(\bar{q})$ after determining all coefficients

$$N(\bar{q}) = \sum_{i<<m}^{n-1} \Delta_{ijk\ell m}(\bar{q}) \prod_{h\neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i<<\ell}^{n-1} \Delta_{ijk\ell}(\bar{q}) \prod_{h\neq i,j,k,\ell}^{n-1} \bar{D}_h + $$
$$+ \sum_{i<<k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h\neq i,j,k}^{n-1} \bar{D}_h + \sum_{i<j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h\neq i,j}^{n-1} \bar{D}_h + \sum_{i}^{n-1} \Delta_{i}(\bar{q}) \prod_{h\neq i}^{n-1} \bar{D}_h$$

1. **Global** $(N = N)$**-test**
2. **Local** $(N = N)$**-test**
3. **Power-test**

Are those methods **reliable** in detecting **unstable phase space points**?

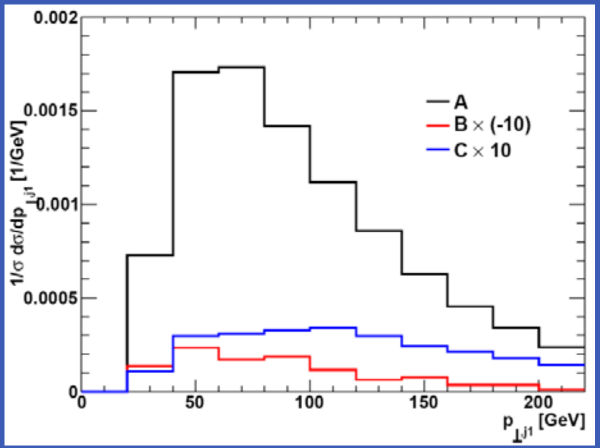
[G.Ossola EPS2001]

W^+W^- + 2 jets @ NLO with GoSam

[Greiner, Heinrich, Mastrolia, Ossola, Reiter, Tramontano 12]

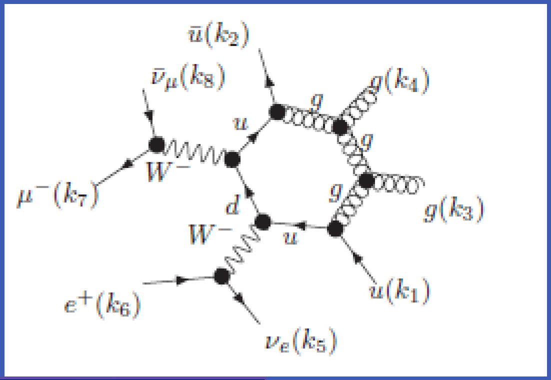
- Part A: no 3rd gen. quarks in fermion loops and VB attached to closed fermion loops, [Melia, Melnikov, Rontsch, Zanderighi 11]
- Part B: VB attached to closed fermion loops,
- Part C: 3rd gen. quarks in the loops.

→ previously unknown (Parts B and C)

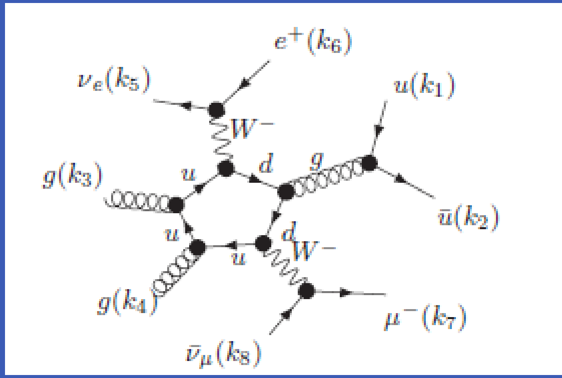


- No b quarks in both initial and final state

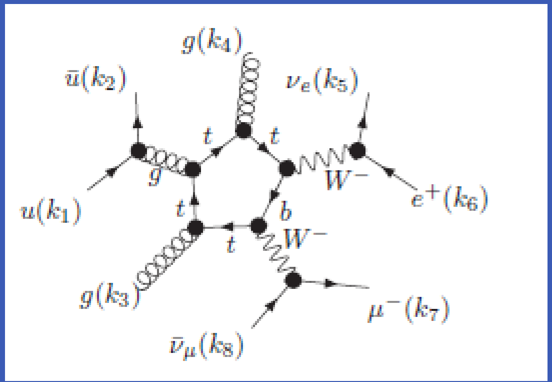
Part A:



Part B:



Part C:



W+W- + 2 jets @ NLO with GoSam

[Greiner, Heinrich, Mastrolia, Ossola, Reiter, Tramontano 12]

Parameters	
$M_W = 80.399$ GeV	$\Gamma_W = 2.085$ GeV
$M_Z = 91.188$ GeV	$\Gamma_Z = 2.4952$ GeV
$M_t = 171.2$ GeV	$\Gamma_t = 0.$ GeV
$M_b = 4.7$ GeV	$\Gamma_b = 0.$ GeV
$\alpha(M_Z) = 1/128.802$	$c_W^2 = M_W^2/M_Z^2$

$E_{T,miss} \geq 30$ GeV

$p_{T,l} \geq 20$ GeV, $|\eta_j| \leq 2.4$

anti-k_T $R = 0.4$

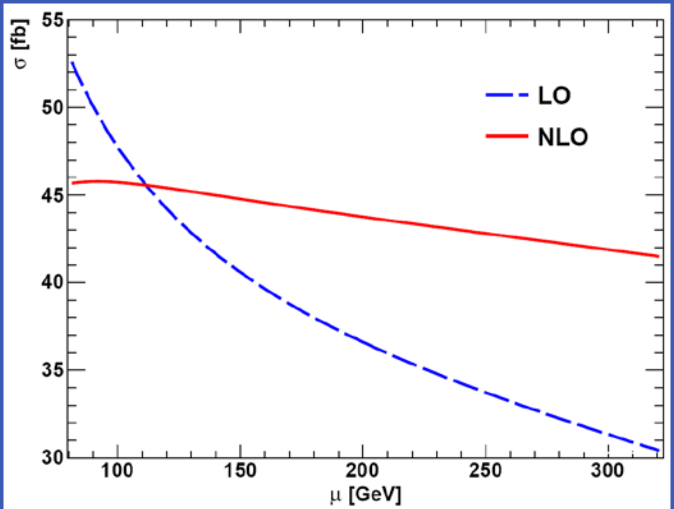
$p_{T,j} \geq 20$ GeV, $|\eta_j| \leq 3.2$, $\Delta R_{jj} \geq 0.4$

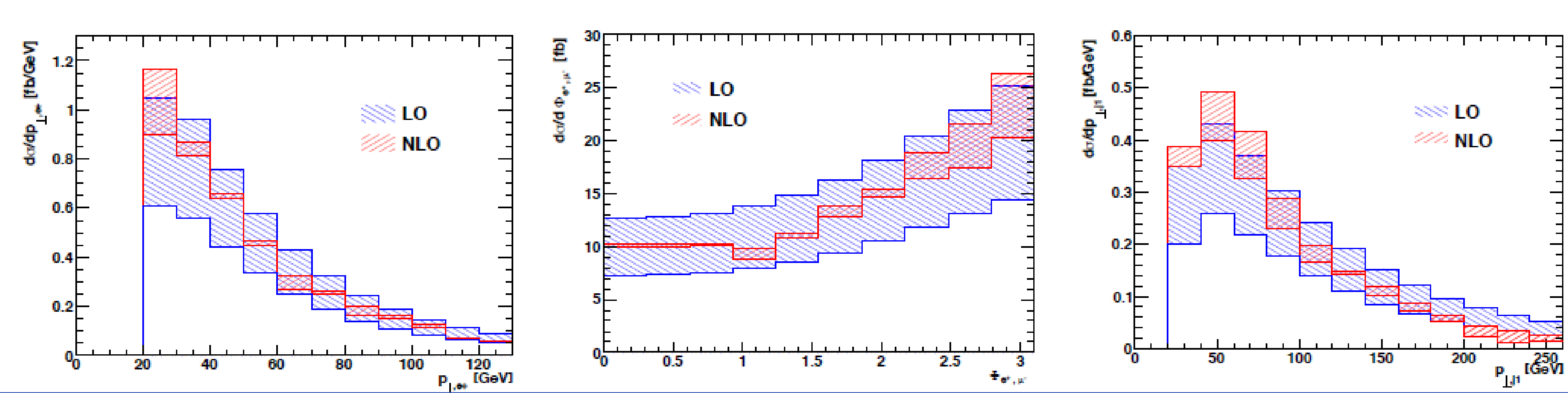
MSTW2008

$\alpha_{s,LO}(M_Z) = 0.13355$

$N_f = 4$

$\alpha_{s,NLO}(M_Z) = 0.1149$





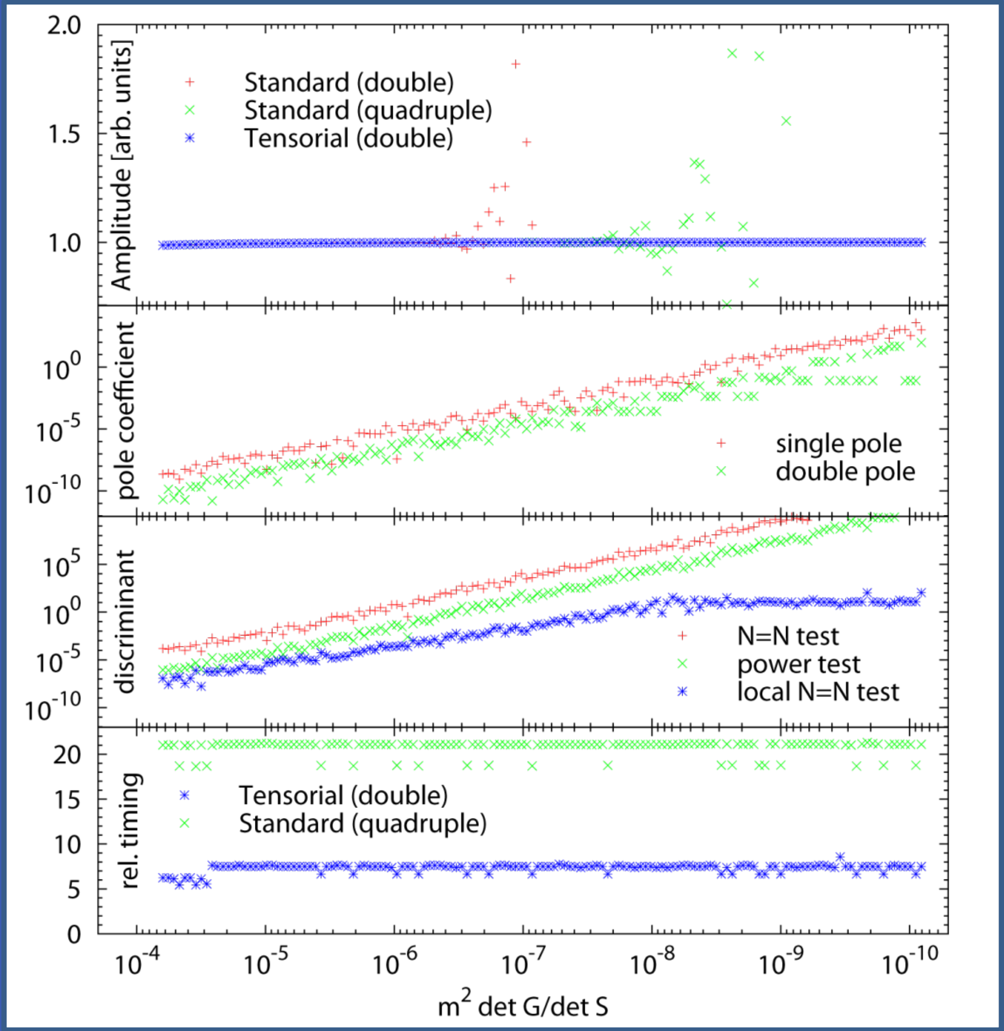
GoSam as standalone code

- When the full code is ready:

```
ttH_virtual : bash
File  Edit  View  Scrollback  Bookmarks  Settings  Help
luisonig@D22:ttH_virtual$ ls
codegen     diagrams-0.hh   diagrams-1.log  helicity1    helicity14  helicity3  helicity6  Makefile.conf    model.hh
common      diagrams-0.log  doc             helicity12   helicity15  helicity4  helicity7  Makefile.source
config.sh   diagrams-1.hh   helicity0       helicity13   helicity2   helicity5  Makefile   matrix
luisonig@D22:ttH_virtual$
```

- Contributions divided in directories by helicity
- Many configuration switches (renorm/scalar loop/reduction strategy) in **`common/config.f90`**
- QCD renormalization fully done
  - different parts can be steered from **`common/config.f90`**
  - different renormalization schemes implemented (DRED/tHV): can partially convert from one to another at runtime (DRED -> CDR) [DRED= dim. reduction, CDR= conv. Dim regulariz., tHV= tHooft-Veltman]
  - Yukawa coupling renormalization is missing!
- Model parameters in **`common/model.f90`**

Approching the Gram



[Heinrich, Ossola, Reiter, Tramontano 10]