Automated Numerical Evaluation of Multi Loop Amplitudes

Stephan Jahn[1]

[1]Max-Planck-Institut für Physik, München

March 29, 2017

[1]sjahn@mpp.mpg.de

Motivation

LHC signals look pretty much like the standard model

- upcoming era of precision physics
- require precision calculations
  - null-test the standard model
  - standard-model background in new-physics searches

automate higher order calculations

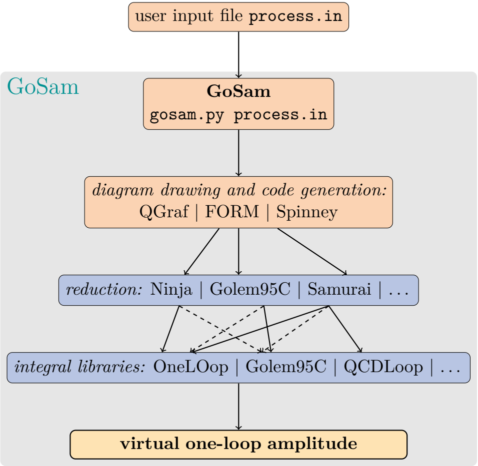
GoSam-1loop

The GoSam collaboration

- Nicolas Greiner
- Gudrun Heinrich
- Stephan Jahn
- Stephen Jones
- Matthias Kerner
- Gionata Luisoni
- Pierpaolo Mastrolia
- Giovanni Ossola
- Tiziano Peraro
- Johannes Schlenk
- Ludovic Scyboz
- Francesco Tramontano

former members

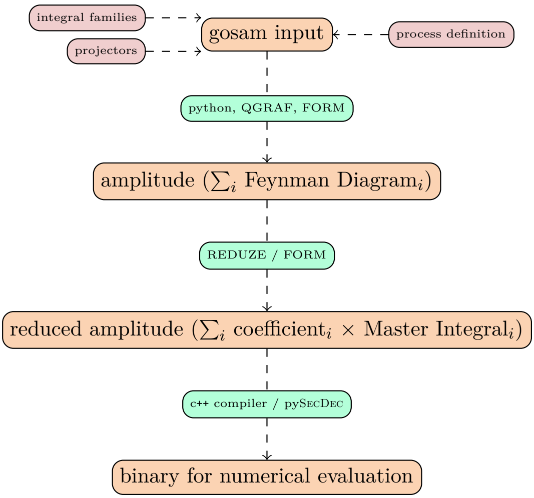
- Gavin Cullen
- Hans van Deurzen
- Edoardo Mirabella
- Joscha Reichel
- Thomas Reiter
- Johann Felix von Soden-Fraunhofen



available at
`http://gosam.hepforge.org/`

The GoSam-2loop collaboration

Nicolas Greiner
Gudrun Heinrich
Stephan Jahn
Stephen Jones
Matthias Kerner
et al.



coming soon

pySecDec

The SecDec collaboration

Sophia Borowka

Gudrun Heinrich

Stephan Jahn

Stephen Jones

Matthias Kerner

Johannes Schlenk

former members

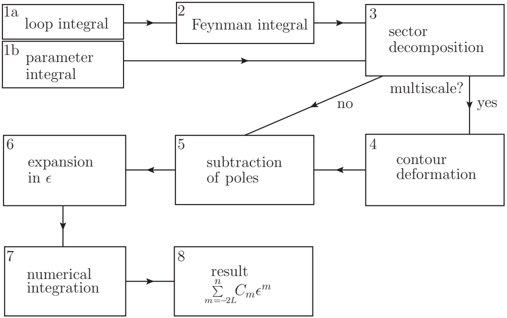
Thomas Binoth

Jonathon Carter

Tom Zirke

Other Implementations

- *C. Bogner, S. Weinzierl*
  Sector decoposition
  [0709.4092]
- *A.V. Smirnov*
  FIESTA 4
  [1511.03614]



available at
http://secdec.hepforge.org/

Master Integral - Momentum Representation

$$\mathcal{I} = \int d^D k_1 \cdot \ldots \cdot d^D k_L \frac{1}{P_1^{\nu_1} \cdot \ldots \cdot P_N^{\nu_N}}$$

D: dimensionality

L: number of loops

N: number of propagators

P_i: propagators ($< momentum >^2 \left[- < mass >^2\right] + i\delta$)

ν_i: propagator powers

Master Integral - Feynman Representation

$$\mathcal{I} = (-1)^{N_\nu} \frac{\Gamma(N_\nu - LD/2)}{\prod_{j=1}^{N} \Gamma(\nu_j)} \int_0^1 \prod_{j=1}^{N} dx_j x_j^{\nu_j - 1} \delta\left(1 - \sum_{n=1}^{N} x_n\right) \frac{\mathcal{U}^{N_\nu - (L+1)D/2}}{\mathcal{F}^{N_\nu - LD/2}}$$

D: dimensionality

L: number of loops

N: number of propagators

ν_i: propagator powers (possibly negative)

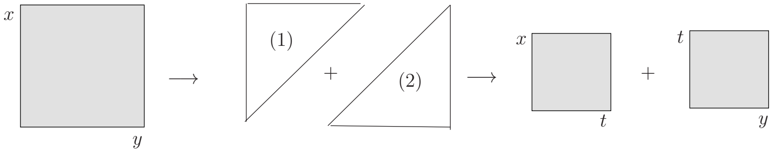
$N_\nu = \sum_{i=1}^{N} \nu_i$

$\mathcal{U}$: 1st Symanzik polynomial

$\mathcal{F}$: 2nd Symanzik polynomial

Sector Decomposition

or: Resolution of Overlapping Singularities



$$\int_0^1 dx \int_0^1 dy\ (x+y)^{a+b\epsilon} f(x,y)$$

$$= \int_0^1 dx \int_0^1 dy\ (x+y)^{a+b\epsilon} f(x,y)[\underbrace{\Theta(x-y)}_{(1)} + \underbrace{\Theta(y-x)}_{(2)}]$$

$$= \int_0^1 dx \int_0^1 dt\, x\, x^{a+b\epsilon} (1+t)^{a+b\epsilon} f(x,xt) + \int_0^1 dt \int_0^1 dy\, y\, y^{a+b\epsilon} (t+1)^{a+b\epsilon} f(yt,y)$$

Subtraction of Poles

$$
\begin{aligned}
&\int_0^1 dt\, t^{-1+b\epsilon} g(t) \\
&= \int_0^1 dt\, t^{-1+b\epsilon} \left(g(0) + g(t) - g(0) \right) \\
&= \underbrace{\int_0^1 dt\, t^{-1+b\epsilon} g(0)}_{=\frac{1}{b\epsilon} g(0)} + \underbrace{\int_0^1 dt\, t^{-1+b\epsilon} \left(g(t) - g(0) \right)}_{\text{finite for } \epsilon \to 0 \text{, expand integrand in } \epsilon}
\end{aligned}
$$

Basic Usage

$$\int_0^1 dx \int_0^1 dy (x+y)^{-2+\epsilon} = \frac{1}{\epsilon} + (1 - \log(2)) + O(\epsilon) \approx \frac{1}{\epsilon} + 0.306853 + O(\epsilon)$$

Step 1: write input files

generate_easy.py

```python
from pySecDec import make_package

make_package(

name = 'easy',
integration_variables = ['x','y'],
regulators = ['eps'],

requested_orders = [0],
polynomials_to_decompose = ['(x+y)^(-2+eps)'],

)
```

integrate_easy.py

```python
from pySecDec.integral_interface \
    import IntegralLibrary

# load c++ library
easy_integral = \
    IntegralLibrary('easy/easy_pylink.so')

# integrate
_, _, result = easy_integral()

# print result
print('Numerical Result:')
print(result)
```

Step 2: run pySecDec

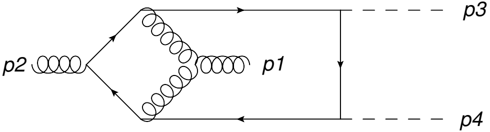
```
$ python generate_easy.py && make -C easy && python integrate_easy.py
<skipped some output>
Numerical Result:
 + (1.00015897181235158e+00 +/- 4.03392522752491021e-03)*eps^-1 + (3.06903035514056399e-01 +/-
    2.82319349818329918e-03) + O(eps)
```

New Features in pySecDec (SecDec-4)

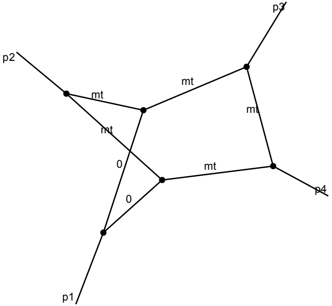
- fully relying on open-source software
- output of a C++ library suitable for amplitude calculations
- support more general tensor numerators
- can have for multiple regulators

Application: Higgs Boson Pair Production

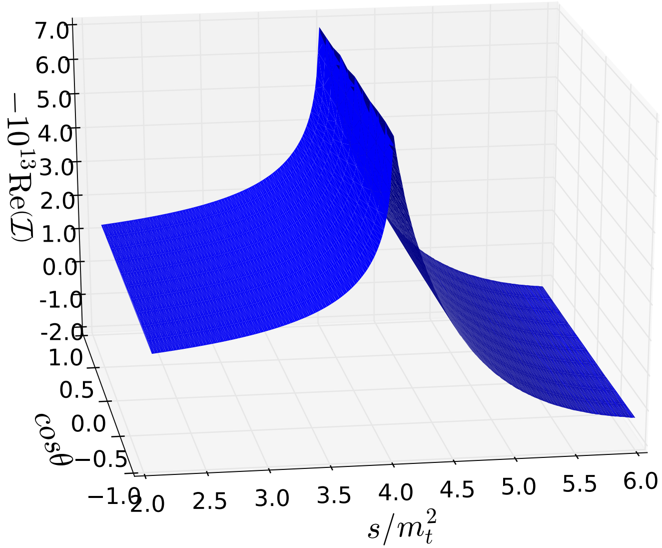
S. Borowka, N. Greiner, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk, T. Zirke [1608.04798]

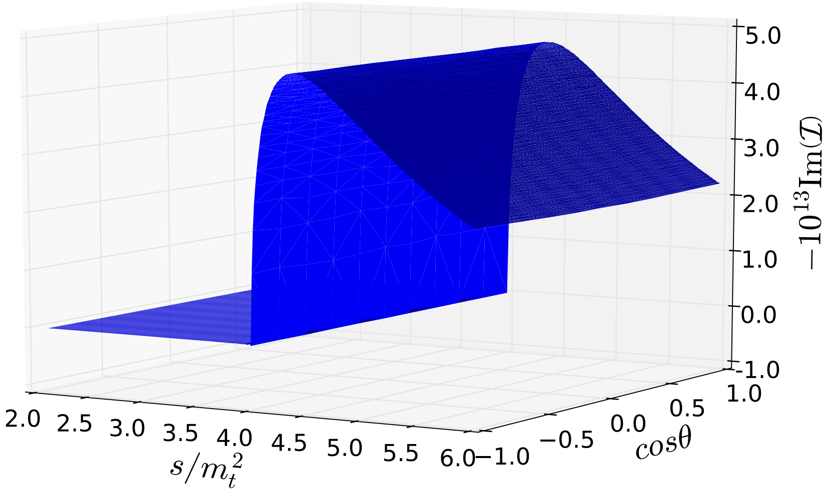


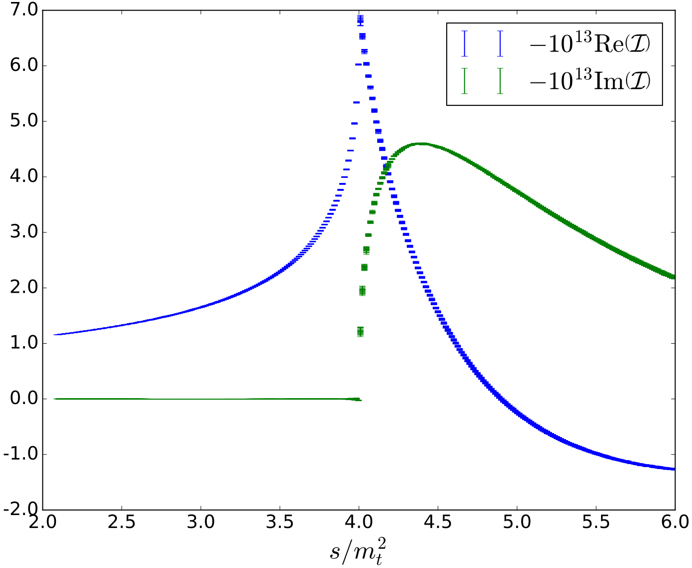
$\mu^{-6-4\epsilon}\,\mathcal{I} \equiv$

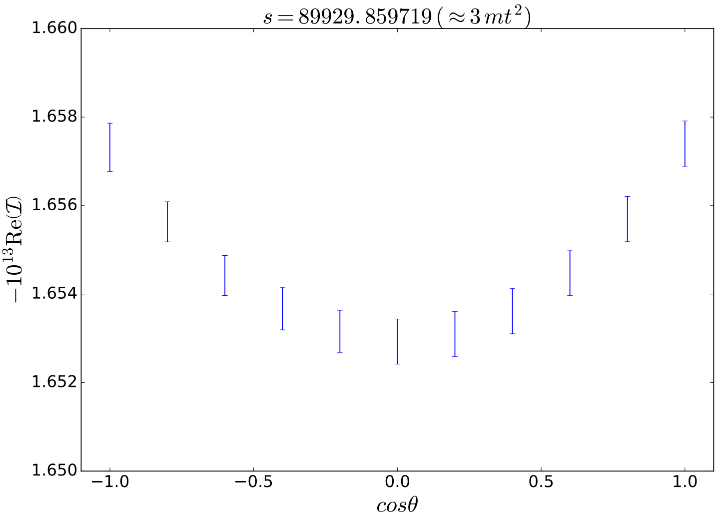


$, \ \mu = 1\mathrm{GeV}$


7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
$-10^{13}\mathrm{Re}(\mathcal{I})$
1.0
0.5
0.0
−0.5
−1.0
$cos\theta$
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
s/m_t^2


5.0
4.0
3.0
2.0
1.0
0.0
-1.0
$-10^{13}\mathrm{Im}(\mathcal{I})$
2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
s/m_t^2
−1.0 −0.5 0.0 0.5 1.0
$cos\theta$

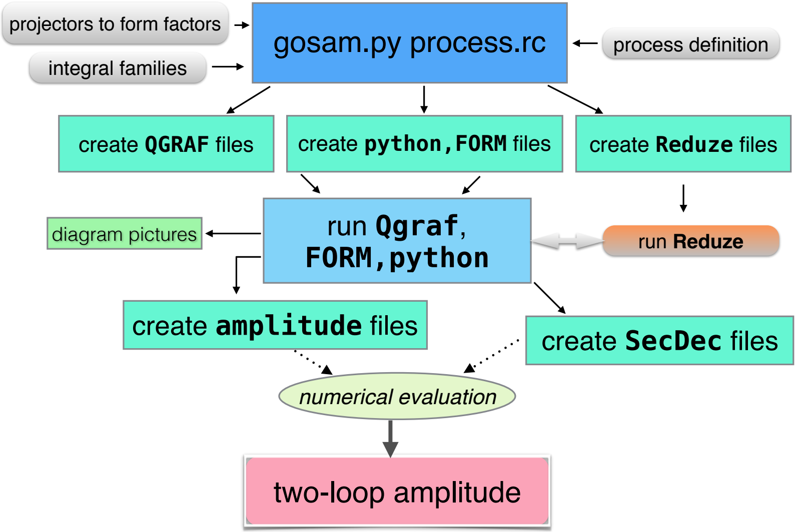

7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
$-10^{13}\mathrm{Re}(\mathcal{I})$
$-10^{13}\mathrm{Im}(\mathcal{I})$
s/m_t^2


$s = 89929.859719\,(\approx 3\,mt^2)$
1.660
1.658
1.656
1.654
1.652
1.650
$-10^{13}\mathrm{Re}(\mathcal{I})$
−1.0
−0.5
0.0
0.5
1.0
$cos\theta$

Summary

- motivate need of automated precision calculations
  - GoSam (http://gosam.hepforge.org/)
  - pySecDec (http://secdec.hepforge.org/)

- introduction to the Sector Decomposition approach
  - description of the method
  - simple example
  - application in gg $\rightarrow$ HH [1608.04798]


projectors to form factors
integral families
gosam.py process.rc
process definition
create QGRAF files
create python,FORM files
create Reduze files
diagram pictures
run Qgraf,
FORM,python
run Reduze
create amplitude files
create SecDec files
numerical evaluation
two-loop amplitude

Basic Usage - Analytical Calculation

$$\int_0^1 dx \int_0^1 dy (x+y)^{-2+\epsilon}$$

$$= 2 \int_0^1 dx\, x^{-1+\epsilon} \int_0^1 dt\ (1+t)^{-2+\epsilon}$$

$$= \frac{2}{\epsilon} \left[\int_0^1 dt\ (1+t)^{-2} + \epsilon \int_0^1 dt\ (1+t)^{-2} \log(1+t) + O(\epsilon^2) \right]$$

$$= \frac{2}{\epsilon} \left[\frac{1}{2} + \epsilon \frac{1}{2} (1 - \log(2)) + O(\epsilon^2) \right] = \frac{1}{\epsilon} + (1 - \log(2)) + O(\epsilon)$$