

pySECDEC: a toolbox for the numerical evaluation of multi-scale integrals

Stephan Jahn¹

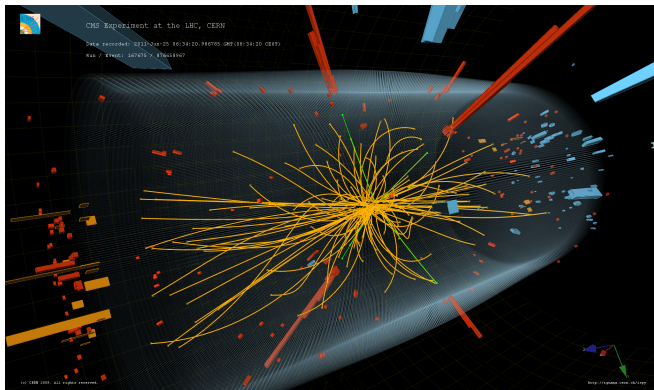
¹Max-Planck-Institut für Physik, München

Particle Physics School Colloquium

July 12, 2018

Introduction

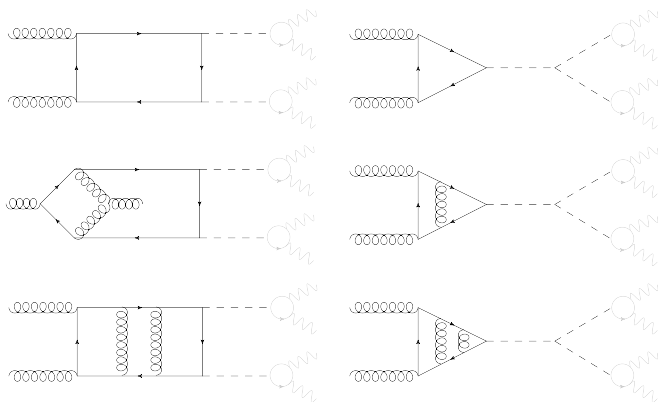
precision tests of the Standard Model at the LHC
requires computation of multi-loop Feynman integrals



<http://cds.cern.ch/journal/CERNBulletin/2011/41/News%20Articles/1387902>

Introduction

precision tests of the Standard Model at the LHC
requires computation of multi-loop Feynman integrals



pySECDEC

successor of SECDEC-3

S. Borowka, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk, T. Zirke [1502.06595]

Numerically computes regulated parameter integrals of the form

$$\mathcal{I} \equiv \int_0^1 dx_1 \dots \int_0^1 dx_N \prod_{i=1}^m f_i(\vec{x}, \vec{a})^{b_i + \sum_k c_{ik} \epsilon_k}$$

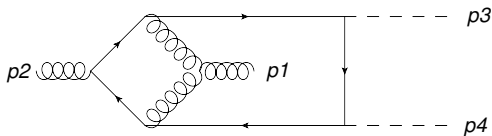
where the f_i are polynomials. Typically: $\mathcal{I}|_{\epsilon_k=0} = \infty$.

Example:

Loop integrals

after Feynman parametrization

in dimensional regularization



The SECDEC collaboration

Sophia Borowka
Gudrun Heinrich
Stephan Jahn
Stephen Jones
Matthias Kerner
Johannes Schlenk

former members

Thomas Binoth
Jonathon Carter
Tom Zirke

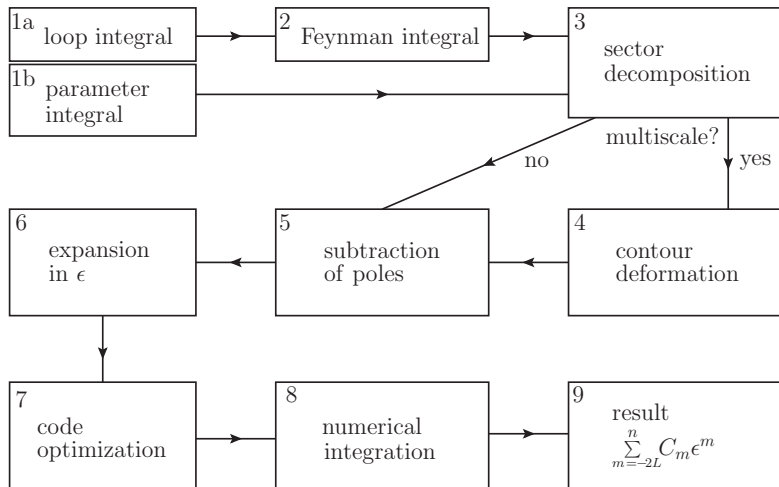
Paper

CPC 222 (2018) 313
[1703.09692]

Other public Implementations

- *C. Bogner, S. Weinzierl*
Sector decomposition
[0709.4092]
supplemented by:
J. Gluza, K. Kajda, T. Riemann, V. Yundin
CSectors
[1010.1667]
- *A.V. Smirnov*
FIESTA 4
[1511.03614]

Flowchart



Loop Integral - Momentum Representation

$$\mathcal{I} = \int \frac{d^D k_1}{i\pi^{\frac{D}{2}}} \cdots \frac{d^D k_L}{i\pi^{\frac{D}{2}}} \frac{1}{P_1^{\nu_1} \cdots P_N^{\nu_N}}$$

D : dimensionality

L : number of loops

N : number of propagators

P_i : propagators ($\langle momentum \rangle^2 [-\langle mass \rangle^2] + i\delta$)

ν_i : propagator powers

Loop 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

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

$\mathcal{U} = \mathcal{U}(\vec{x})$: 1st Symanzik polynomial

$\mathcal{F} = \mathcal{F}(\vec{x}, p_i \cdot p_j, m_i^2)$: 2nd Symanzik polynomial

Feynman Parametrization with pySECDEC

```

1  >>> from pySecDec.loop_integral import
    ↪ LoopIntegralFromPropagators
2
3  >>> one_loop_bubble =
    ↪ LoopIntegralFromPropagators(
4  ...     propagators=['k^2-m^2', '(k-p)^2'],
5  ...     loop_momenta=['k']
6  ... )
7
8  >>> one_loop_bubble.exponentiated_U
9  ( + (1)*x0 + (1)*x1)**(2*eps - 2)
10
11 >>> one_loop_bubble.exponentiated_F
12 ( + (m**2 - p**2)*x0*x1 +
    ↪ (m**2)*x0**2)**(-eps)
13
14 >>> one_loop_bubble.Gamma_factor
15 gamma(eps)

```

```

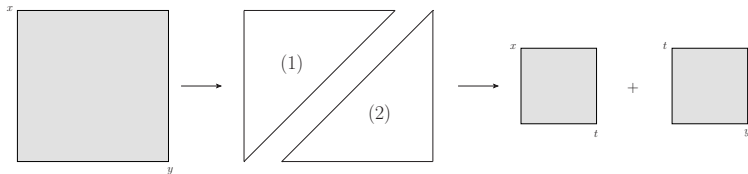
1  >>> from pySecDec.loop_integral import
    ↪ LoopIntegralFromGraph, plot_diagram
2
3  >>> one_loop_bubble = LoopIntegralFromGraph(
4  ...     internal_lines=[('m',(1,2)], [0,(1,2)]],
5  ...     external_lines=[('p',1), ('q',2)],
6  ...     replacement_rules=[('q','p')]
7  ... )
8
9  >>> one_loop_bubble.exponentiated_U
10 ( + (1)*x0 + (1)*x1)**(2*eps - 2)
11
12 >>> one_loop_bubble.exponentiated_F
13 ( + (m**2 - p**2)*x0*x1 +
    ↪ (m**2)*x0**2)**(-eps)
14
15 >>> one_loop_bubble.Gamma_factor
16 gamma(eps)
17
18 >>> plot_diagram(
19 ...     one_loop_bubble.internal_lines,
20 ...     one_loop_bubble.external_lines,
21 ...     filename='bubble1L',
22 ...     Gstart=986089
23 ... )

```



Sector Decomposition

or: Resolution of Overlapping Singularities



$$\begin{aligned}
 & \int_0^1 dx \int_0^1 dy (x+y)^{a+b\epsilon} f(x, y) \\
 &= \int_0^1 dx \int_0^1 dy (x+y)^{a+b\epsilon} f(x, y) [\underbrace{\Theta(x-y)}_{(1)} + \underbrace{\Theta(y-x)}_{(2)}] \\
 &= \int_0^1 dx \int_0^1 dt x x^{a+b\epsilon} (1+t)^{a+b\epsilon} f(x, xt) + \int_0^1 dt \int_0^1 dy y y^{a+b\epsilon} (t+1)^{a+b\epsilon} f(yt, y)
 \end{aligned}$$

Subtraction of Poles

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

Contour Deformation - A Simple Example

consider the massive one loop bubble



$$\int d^D k \frac{1}{(k^2 - m^2 + i\delta) ((k-p)^2 - m^2 + i\delta)}$$

taking $\delta(1 - x_1 - x_2)$ into account:

$$\begin{aligned} \mathcal{F}(\vec{x}, p_i \cdot p_j, m_i^2) &= [m^2 (x_1 + x_2)^2 - p^2 x_1 x_2]_{x_2=1-x_1} \\ &= m^2 - p^2 x_1 (1 - x_1) \end{aligned}$$

- ▶ physical threshold $p^2 \geq 4m^2$
- ▶ can have $\mathcal{F} = 0$ although $x_1 \neq 0$

Contour Deformation

or: Satisfying the Feynman Prescription

D. E. Soper [hep-ph/9910292], T. Binoth et al. [hep-ph/0504267], C. Anastasiou et al. [hep-ph/0703282], S. Borowka et al. [1204.4152]

$$\begin{aligned} \mathcal{I} &= \int \frac{d^D k_1}{i\pi^{\frac{D}{2}}} \cdot \dots \cdot \frac{d^D k_L}{i\pi^{\frac{D}{2}}} \frac{1}{P_1^{\nu_1} \cdot \dots \cdot P_N^{\nu_N}} \\ &= (-1)^{N_\nu} \frac{\Gamma(N_\nu - LD/2)}{\prod_{j=1}^N \Gamma(\nu_j)} \int_0^1 \prod_{j=1}^N dx_j x_j^{\nu_j-1} \delta\left(1 - \sum_{n=1}^N x_n\right) \frac{\mathcal{U}^{N_\nu - (L+1)D/2}}{\mathcal{F}^{N_\nu - LD/2}} \end{aligned}$$

P_i : propagators ($\langle \text{momentum} \rangle^2 [-\langle \text{mass} \rangle^2] + \boxed{i\delta}$)

$\mathcal{F} = \mathcal{F}(\vec{x}, p_i \cdot p_j, m_i^2) - \boxed{i\delta}$: 2nd Symanzik polynomial

Contour Deformation

or: Satisfying the Feynman Prescription

D. E. Soper [hep-ph/9910292], T. Binoth et al. [hep-ph/0504267], C. Anastasiou et al. [hep-ph/0703282], S. Borowka et al. [1204.4152]

$$\mathcal{F}(\vec{x}, p_i \cdot p_j, m_i^2) - \boxed{i\delta}$$

move integration contour to the complex plane

$$\vec{z}: [0, 1]^n \longrightarrow \mathbb{C}^n$$

$$x_k \rightarrow z_k(\vec{x})$$

$$\equiv x_k - i\lambda_k x_k (1 - x_k) \frac{\partial \operatorname{Re}[\mathcal{F}(\vec{x})]}{\partial x_k}$$

such that

$$\operatorname{Im}[\mathcal{F}(\vec{z}(\vec{x}))] \leq \operatorname{Im}[\mathcal{F}(\vec{x})] \quad \forall \vec{x} \in [0, 1]^n$$

Contour Deformation

or: Satisfying the Feynman Prescription

D. E. Soper [hep-ph/9910292], T. Binoth et al. [hep-ph/0504267], C. Anastasiou et al. [hep-ph/0703282], S. Borowka et al. [1204.4152]

$$\begin{aligned}
 \mathcal{F}(\vec{z}(\vec{x})) &= + \operatorname{Re} \mathcal{F}(\vec{x}) + i \operatorname{Im} \mathcal{F}(\vec{x}) \\
 &+ \sum_k \lambda_k \left[-i \left(\frac{\partial \operatorname{Re} \mathcal{F}(\vec{x})}{\partial x_k} \right)^2 + \frac{\partial \operatorname{Re} \mathcal{F}(\vec{x})}{\partial x_k} \frac{\partial \operatorname{Im} \mathcal{F}(\vec{x})}{\partial x_k} \right] x_k (1 - x_k) \\
 &+ \sum_{k,l} \frac{\lambda_k \lambda_l}{2} \left[-\frac{\partial^2 \operatorname{Re} \mathcal{F}(\vec{x})}{\partial x_k \partial x_l} \quad -i \frac{\partial^2 \operatorname{Im} \mathcal{F}(\vec{x})}{\partial x_k \partial x_l} \right] \prod_{i=k,l} \frac{\partial \operatorname{Re} \mathcal{F}(\vec{x})}{\partial x_i} x_i (1 - x_i) \\
 &+ \mathcal{O}(\lambda^3)
 \end{aligned}$$

- ▶ correct sign at $\mathcal{O}(\lambda)$
- ▶ indeterminate sign at $\mathcal{O}(\lambda^2)$

correct sign if λ_k “small enough”

Expansion in the Regulator(s) ϵ

- ▶ Taylor expansion and series multiplication
poles in ϵ explicitly factorize by construction
- ▶ cumbersome bookkeeping
individual factors must be expanded to different orders

Code Optimization

- ▶ faster numerics due to **Code Optimization in FORM**
J. Kuipers, T. Ueda, J.A.M. Vermaseren [1310.7007]

Numerical Integration

- ▶ CUBA and `gsl_integration_cquad` (GNU scientific library)
T. Hahn [hep-ph/0404043], *M. Galassi et al.* [GSL Reference Manual]
- ▶ can link any numerical integrator using the C++ interface

A Simple Example - Analytic Calculation

$$\begin{aligned}
 & \int_0^1 dx \int_0^1 dy (x+y)^{-2+\epsilon} \\
 &= 2 \int_0^1 dx x^{-1+\epsilon} \int_0^1 dt (1+t)^{-2+\epsilon} \\
 &= \frac{2}{\epsilon} \left[\int_0^1 dt (1+t)^{-2} + \epsilon \int_0^1 dt (1+t)^{-2} \log(1+t) + O(\epsilon^2) \right] \\
 &= \frac{2}{\epsilon} \left[\frac{1}{2} + \epsilon \frac{1}{2} (1 - \log(2)) + O(\epsilon^2) \right] = \frac{1}{\epsilon} + (1 - \log(2)) + O(\epsilon)
 \end{aligned}$$

A Simple Example - Basic pySECDEC 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

```

1 from pySecDec import make_package
2
3 make_package(
4
5     name = 'easy',
6     integration_variables = ['x','y'],
7     regulators = ['eps'],
8
9     requested_orders = [0],
10    polynomials_to_decompose = ['(x+y)^(-2+eps)'],
11
12 )

```

integrate_easy.py

```

1 from pySecDec.integral_interface \
2     import IntegralLibrary
3
4 # load c++ library
5 easy_integral = \
6     IntegralLibrary('easy/easy_pylink.so')
7
8 # integrate
9 _, _, result = easy_integral()
10
11 # print result
12 print('Numerical Result:')
13 print(result)

```

Step 2: run pySECDEC

```

1 $ python generate_easy.py && make -C easy && python integrate_easy.py
2 <skipped some output>
3 Numerical Result:
4 + (1.00015897181235158e+00 +/- 4.03392522752491021e-03)*eps^-1 + (3.06903035514056399e-01 +/-
  ↪ 2.82319349818329918e-03) + 0(eps)

```

A Simple Example - pySECDEC Documentation

The screenshot shows a web browser displaying the documentation for a simple example. The page title is "2.1. A Simple Example". The left sidebar contains a navigation menu with items: "1. Installation", "2. Getting Started" (expanded), "2.1. A Simple Example" (selected), "2.2. Evaluating a Loop Integral", "2.3. List of Examples", "3. Overview", "4. SecDecUtil", "5. Reference Guide", and "6. References". The main content area starts with the heading "2.1. A Simple Example" and the text "We first show how to compute a simple dimensionally regulated integral:". Below this is a mathematical equation:
$$\int_0^1 dx \int_0^1 dy (x+y)^{-2\epsilon}$$
. The text continues: "To run the example change to the `easy` directory and run the commands:" followed by a code block:

```
$ python generate_easy.py
$ make -C easy
$ python integrate_easy.py
```

 The next text says "This will evaluate and print the result of the integral:" followed by a code block showing numerical and analytic results:

```
Numerical Result: + (1.00015697181235158e+00 +/- 4.03192522752491821e-03)*eps^-1 + (3.06903035514056399)
Analytic Result: + (1.0000000)*eps^-1 + (0.306853) + O(eps)
```

 The text then explains: "The file `generate_easy.py` defines the integral and calls `pySecDec` to perform the sector decomposition. When run it produces the directory `easy` which contains the code required to numerically evaluate the integral. The `make` command builds this code and produces a library. The file `integrate_easy.py` loads the integral library and evaluates the integral. The user is encouraged to copy and adapt these files to evaluate their own integrals." Below this is a "Note" box:

Note

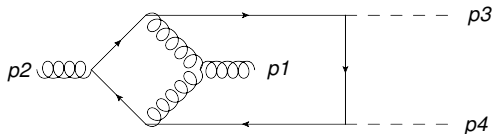
If the user is interested in evaluating a loop integral there are many convenience functions that make this much easier. Please see [Evaluating a Loop Integral](#) for more details.

 At the bottom of the page, the URL `http://secdec.readthedocs.io/` is displayed in a large font.

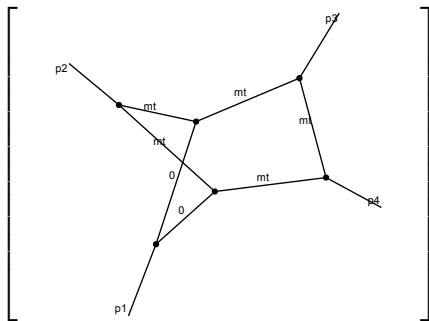
`http://secdec.readthedocs.io/`

Higgs Boson Pair Production

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



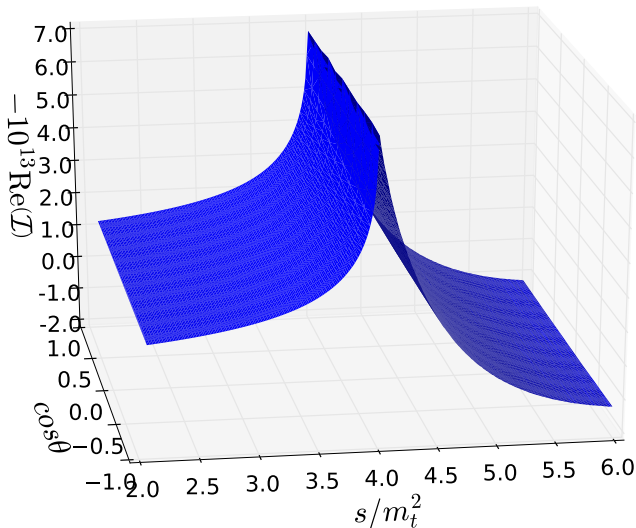
$\mu^{-6-4\epsilon} \mathcal{I} \equiv \text{finite}$



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

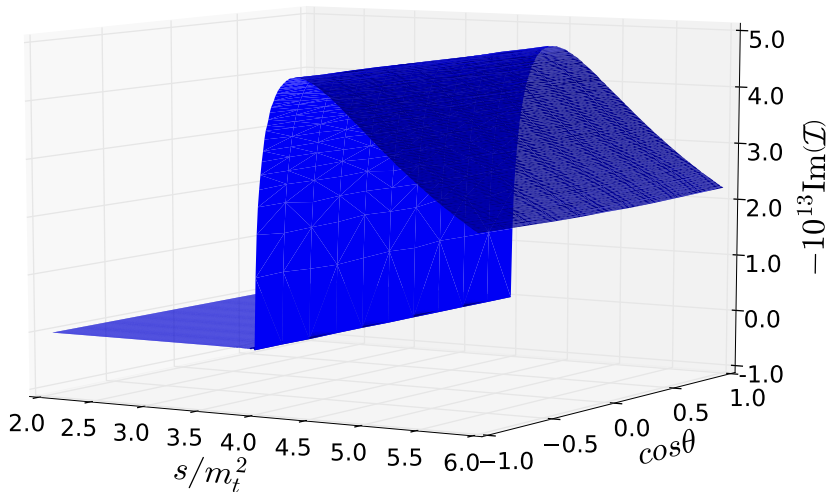
Higgs Boson Pair Production

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



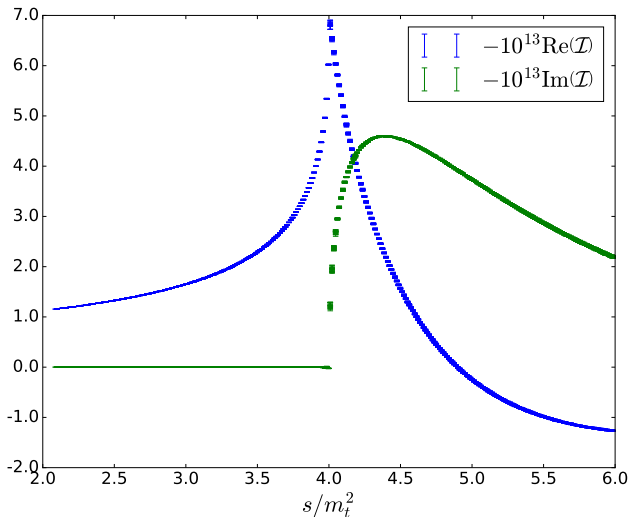
Higgs Boson Pair Production

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



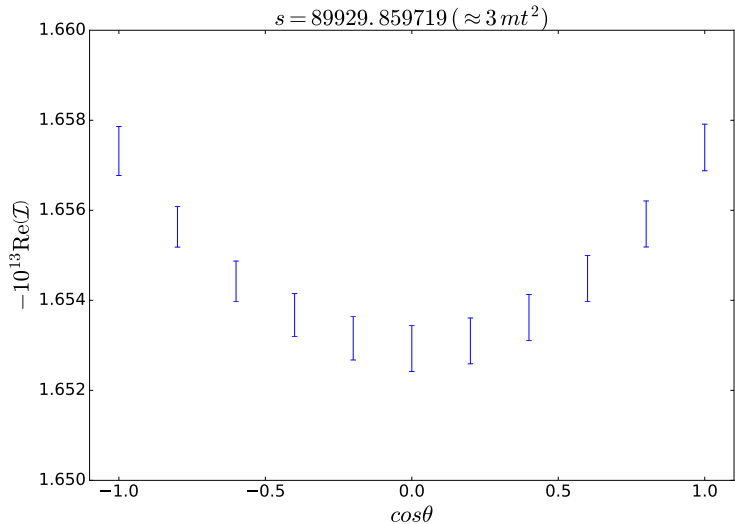
Higgs Boson Pair Production

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



Higgs Boson Pair Production

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



Conclusion

- ▶ pySECDEC: automated numerical evaluation of loop integrals
- ▶ can be used to compute amplitudes and physical cross-sections
- ▶ further reading:
 - ▶ homepage: <http://secdec.hepforge.org/>
 - ▶ source code: <https://github.com/mppmu/secdec/>
 - ▶ documentation: <http://secdec.readthedocs.io/>

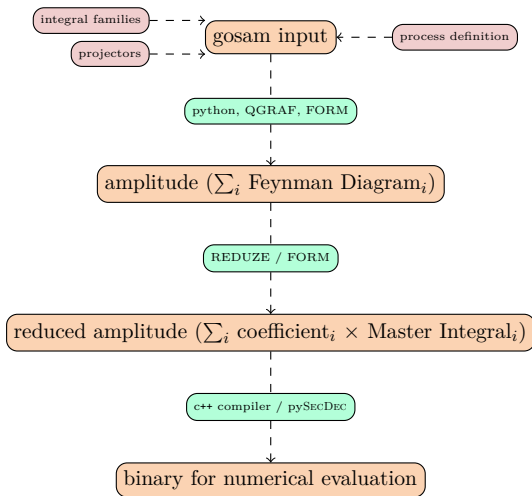
BACKUP

Timings

Table 5
Comparison of timings (algebraic, numerical) using pySECDEC, SECDEC 3 and FIESTA 4.1.

	pySECDEC time (s)	SECDEC 3 time (s)	FIESTA 4.1 time (s)
triangle2L	(40.5, 9.6)	(56.9, 28.5)	(211.4, 10.8)
triangle3L	(110.1, 0.5)	(131.6, 1.5)	(48.9, 2.5)
elliptic2L_euclidean	(8.2, 0.2)	(4.2, 0.1)	(4.9, 0.04)
elliptic2L_physical	(21.5, 1.8)	(26.9, 4.5)	(115.3, 4.4)
box2L_invprop	(345.7, 2.8)	(150.4, 6.3)	(21.5, 8.8)

GoSAM-Xloop



The GoSAM-Xloop collaboration

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