Numerical evaluation of multi-loop integrals with pySecDec

Sophia Borowka

CERN

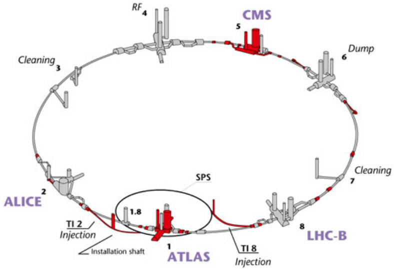
Collaborators: G. Heinrich, S. Jahn, S.P. Jones, M. Kerner, J. Schlenk, T. Zirke

Based on: 1703.09692 [hep-ph]

http://secdec.hepforge.org

LoopFest XVI, Argonne National Laboratory, June 1st 2017

Higher order predictions



- we need accurate predictions at NLO and beyond
- we want differential predictions
- we want to take exact mass dependence of t, H, W, Z, (b, c) into account
- phenomenological predictions involve complicated multi-loop multi-scale integrals

Techniques for multi-loop multi-scale integrals

- Diverse approaches on the market to compute these
  - Feynman parametrization, Mellin-Barnes representation, differential equations, difference equations, dispersion integrals, integrals in coordinate space, gluing, experimental mathematics, combination of several methods

  Many people involved: Anastasiou, Bailey, Bauberger, Berends, Bogner, Böhm, Broadhurst, Brown, Davydychev, Duhr, Ferguson, Fleischer, Gangl, Gehrmann, Glover, Halliday, Henn, t'Hooft, Jegerlehner, Kalmykov, Kniehl, Kotikov, Kreimer, Lee, Panzer, Remiddi, Rhodes, Ricotta, Smirnov, Tancredi, Tarasov, Tausk, Veltman, Veretin, Weinzierl, ...
- contribution lookup table: **loop** encyclop**edia** (to appear)

Loopedia

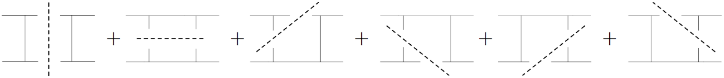
Feynman parametrized loop integrals

- Feynman parametrization is highly automatable
- Scalar multi-loop integral in Feynman parametrization

$$G = \frac{(-1)^{N_\nu}}{\prod_{j=1}^{N}\Gamma(\nu_j)}\Gamma(N_\nu - LD/2)\int\limits_0^\infty \prod_{j=1}^{N} \mathrm{d}x_j \; x_j^{\nu_j-1}\,\delta(1-\sum_{l=1}^{N} x_l)\frac{\mathcal{U}(\vec{x})^{N_\nu-(L+1)D/2}}{\mathcal{F}(\vec{x},s_{ij})^{N_\nu-LD/2}}$$

D : dimension, N : #propagators, L : #loops,
ν_j : propagator powers, $N_\nu = \sum_{j=1}^{N}\nu_j$

- $\mathcal{U}$ and $\mathcal{F}$ can be constructed via **topological cuts** or by the propagators in momentum space



$$\mathcal{F} = -s_{12}\,x_1x_3 - s_{23}\,x_2x_4 - p_1^2\,x_1x_2 - p_2^2\,x_2x_3 - p_3^2\,x_3x_4 - p_4^2\,x_4x_1\;.$$

Analytic vs. numerical approach

Analytical:

+ in general fast evaluation of result
+ dependence on kinematic variables visible
- adding more scales is a challenge
- automation difficult

Numerical:

+ highly automatable
+ scales are not a problem
- precision vs. speed
- intuitive understanding of result harder

Numerical evaluation of Feynman integrals

Main problems beyond the one-loop level:

- Extraction of IR and UV singularities
- Numerical convergence in the presence of integrable singularities (e.g. thresholds)
- Speed / accuracy

Many people are/have been working on numerical methods for phenomenological predictions, e.g. Anastasiou/Beerli/Kunszt et al., Bogner/Reuschle/Weinzierl et al., Binoth/Heinrich et al., Boughezal/Melnikov/Petriello et al., Czakon et al., Freitas et al., Gluza et al., Kurihara et al., Nagy/Soper et al., Passarino et al.,...

Extraction of IR and UV singularities

IR and UV singularity extraction beyond 1-loop

Diverse methods have been worked out

- R*-operation Chetyrkin, Tkachov, V.A.Smirnov '70s, '80s
- Polynomial exponent raising Tkachov '96, Passarino '00
- Sector decomposition Binoth & Heinrich '00
- Computation of residues within Mellin-Barnes representation
  Anastasiou, Daleo '06; Czakon '06
- Subtraction terms Freitas '12; Becker, Weinzierl '12
- Quasi-finite basis Panzer '14; Manteuffel, Schabinger, Panzer '14

+ important other works on UV renormalization Bogoliubov, Parasiuk, Hepp, Zimmermann, Broadhurst, Kreimer, Connes,...

Numerical evaluation using sector decomposition

Public codes:

- sector_decomposition (uses GiNaC) Bogner & Weinzierl '07
  supplemented with `CSectors` Gluza, Kajda, Riemann, Yundin '10
  for construction of integrand in terms of Feynman parameters
- FIESTA* (uses Mathematica, C) A.V. Smirnov, V.A. Smirnov, Tentyukov '08 '09, A.V. Smirnov '13, A.V. Smirnov '15
- (py)SecDec* (uses python, C++)
  Carter & Heinrich '10; SB, Carter, Heinrich '12; SB & Heinrich '13; SB, Heinrich, Jones, Kerner, Schlenk, Zirke '15; SB, Heinrich, Jahn, Jones, Kerner, Schlenk, Zirke '17

*Multi-scale integrals not limited to the Euclidean region
SB, J. Carter & G. Heinrich '12; A.V. Smirnov '13

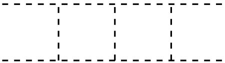
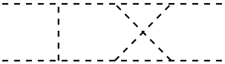
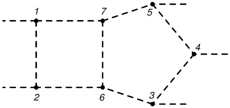
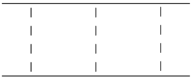
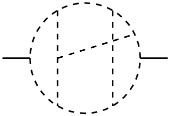
Sector decomposition algorithms

- Iteration of the sector decomposition leads to extraction of IR and UV divergences
- Heuristic algorithm `'iterative'` Binoth & Heinrich '00
  - Infinite recursion may appear
- Other strategies avoid infinite recursion
  Bogner, Weinzierl '07 '08, A. Smirnov, Tentyukov '08, Kaneko, Ueda '09 '10
  - For complicated examples: can lead to more decomposed sectors or functions of higher complexity

Since SecDec 3:

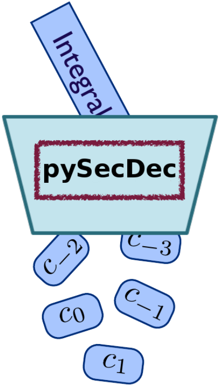
- Deterministic strategy using algebraic geometry `'geometric_ku'` Kaneko and Ueda '09 '10
- `G1` strategy combined with Cheng-Wu theorem `'geometric'`
  SB, Heinrich, Jones, Kerner, Schlenk, Zirke '15

Comparison of decomposition strategies

Diagram	`'iterative'`	`'geometric_ku'`	`'geometric'`
	282 sectors 1 s	266 sectors 8 s	166 sectors 4 s
	368 sectors 1 s	360 sectors 9 s	235 sectors 5 s
	548 sectors 3 s	506 sectors 15 s	304 sectors 4 s
	infinite recursion	72 sectors 5 s	76 sectors 1 s
	27336 sectrs 5510 s	32063 sectrs 11856 s	27137 sectrs 443 s

The program pySecDec

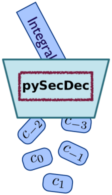
pySecDec use cases



Feynman integral

parametric integral

pySecDec use cases



Feynman integral

parametric integral

- compute single
  - **multi-loop** multi-scale integrals
  - general **parametric** functions
- generate integral library of integrals
- use algebra package for symbolic manipulations on integrals

pySecDec toolbox SB, Heinrich, Jahn, Jones, Kerner, Schlenk, Zirke '17

Primary objective:

- facilitate generation of integral libraries

Improved structure:

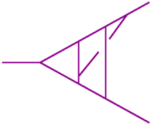
- dependences on open-source libraries only
  NUMPY www.numpy.org, SYMPY www.sympy.org,
  FORM Vermaseren et al. '00 '13, NAUTY McKay, Piperno '13

More power:

- multiple regulators allowed
- iterated remapping of singularities on integration boundary
- decreased number of iterated sectors by finding graph isomorphisms
- no distinction between loop & parametric functions
- evaluation of amplitudes

Download pySecDec

http://secdec.hepforge.org



SecDec

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

A program to evaluate dimensionally regulated parameter integrals numerically

home download program user manual faq changelog

NEW! Download the latest version of pySecDec as pySecDec-1.1.1.tar.gz. The manual is available here.

Download version 1.1 of pySecDec as pySecDec-1.1.tar.gz. The manual is available here.

The first release version of pySecDec can be downloaded as pySecDec-1.0.tar.gz. The manual is available here.
See also the corresponding paper arXiv:1703.09692.

Install pySecDec

- **Install**:
  tar xzvf pySecDec-1.1.1.tar.gz
  cd pySecDec-1.1.1
  make

- **Prerequisites**:
  - python 2.7 or 3
  - python libraries numpy & sympy
  - C++ compiler

- **Optional prerequisites**:
  - geometric decomposition strategies:
    NORMALIZ Bruns, Ichim, Roemer, Soeger '12
  - NEATO http://www.graphviz.org

Input for pySecDec

- no input files needed, direct python interface

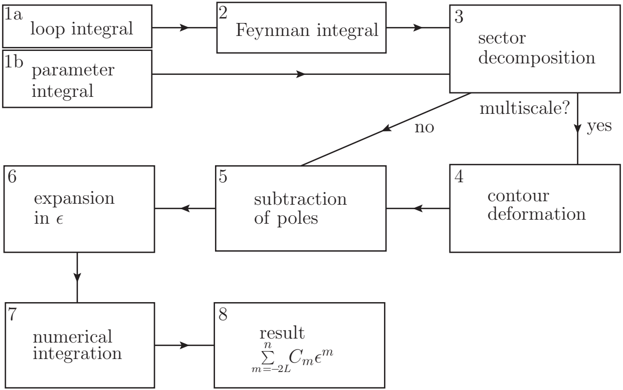
```
#! /usr/bin/env python
from pySecDec.loop_integral import loop_package
import pySecDec as psd

li = psd.loop_integral.LoopIntegralFromGraph(
internal_lines = [['m',[3,4]],['m',[4,5]],['m',[3,5]],[0,[1,2]],[0,[4,1]],[0,[2,5]]],
external_lines = [['p1',1],['p2',2],['p3',3]],

replacement_rules = [
                        ('p1*p1', 0),
                        ('p2*p2', 0),
                        ('p3*p3', 's'),
                        ('p4*p4', 0),
                        ('p1*p2', 's/2'),
                        ('p2*p3', '-s/2'),
                        ('p1*p3', '-s/2'),
                        ('m**2', 'msq')
                    ]
)


Mandelstam_symbols = ['s']
mass_symbols = ['msq']
```

Outline of the program pySecDec



numerical integration: CUBA library Hahn '04, QMC (Coming soon)

Timings

Comparison of pySecDec to SecDec 3

Example	pySecDec timings (s) (algebraic, numerical)	SecDec 3 timings (s) (algebraic, numerical)
P_{126}	(78.5, 27.5)	(78, 57.5)
formfactor3L	(685, 0.3)	(127, 1.5)
f^A_{66}, Euclidean point	(32, 0.01)	(8, 0.3)
f^A_{66}, physical point	(97, 7.6)	(47, 9.8)
box2L with inverse propagator	(1364, 5.4)	(189, 6.5)

4-core Intel(R) Core(TM) i7-3770 CPU @3.40GHz

- algebraic part usually takes longer, but is only done once
- numerical evaluation faster

Numerical integration of two-loop amplitude

So far SecDec has been used for...

- checks of analytically calculated integrals
- prediction of master integrals SB, Heinrich '13; SB '14
- fast evaluation of 34 massive bubble topologies (4 scales) for $\mathcal{O}(\alpha_s\alpha_t)$ MSSM Higgs-mass contributions
  SB, Hahn, Heinemeyer, Heinrich, Hollik '14
- as a library for 513 massive bubble integrals for $\mathcal{O}(\alpha_s\alpha_x)$ MSSM Higgs-mass contributions SB, Passehr, Weiglein, 1706.xxxxx
- as a library for 327 integrals needed in Higgs-pair production at NLO with full top-quark mass dependence SB, N. Greiner, G. Heinrich, S.P. Jones, M. Kerner, J. Schlenk, U. Schubert, T. Zirke 16

$\longrightarrow$ pySecDec is ready for many more applications! $\longleftarrow$

Summary and Outlook

Summary:

- successor of SecDec presented
- pySecDec completely written in python
- dependences on open source software only
- easy to install, flexible modular usage
- decreased timings thanks to efficient code generation

Outlook:

- publish implementation of Quasi Monte Carlo integration
- many phenomenological applications possible
  $(gg \to Hj, ZZ, \gamma\gamma ...)$

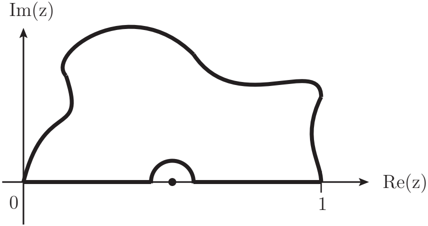
Backup

Extension to physical kinematics

- For kinematics in the physical region, $\mathcal{F}$ can still vanish after sector decomposition

$$\mathcal{F}_{Bubble} = -s\, t_1(1 - t_1) + m^2 - i\delta$$

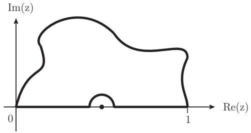
but a deformation of the integration contour



and Cauchy's theorem can help

$$\oint_c f(t)\mathrm{d}t = \int_0^1 f(t)\mathrm{d}t + \int_1^0 \frac{\partial z(t)}{\partial t} f(z(t))\mathrm{d}t = 0$$

Deformation of the integration contour to integrate mass thresholds



- Integrand is analytically continued into the complex plane

$$\mathcal{F}(\vec{t}) \to \mathcal{F}(\vec{t} + \mathrm{i}\vec{y}(\vec{t})) = \mathcal{F}(\vec{t}) + \mathrm{i}\sum_j y_j(\vec{t})\frac{\partial \mathcal{F}(\vec{t})}{\partial t_j} + \mathcal{O}(y(\vec{t})^2)$$

- The integration contour is deformed by

$$\vec{t} \to \vec{z} = \vec{t} + \mathrm{i}\vec{y}\ ,$$

$$y_j(\vec{t}) = -\lambda_j t_j (1 - t_j)\frac{\partial \mathcal{F}(\vec{t})}{\partial t_j}$$

Soper '99, Beerli '08

Soper, Nagy; Binoth; Anastasiou/Beerli/Kunszt et al., Kurihara et al., Freitas et al., Becker/Reuschle/Weinzierl et al.

Optimization of deformation parameter λ_j

SB, Carter, Heinrich '12

1) Normalize $\lambda_j \lesssim 1$ by

$$\lambda_j = \frac{\lambda}{\max[\frac{\partial \mathcal{F}(\vec{t})}{\partial t_j}]_{\text{pre}}}$$

2) If λ^3 terms exist, minimize the ratio of $\mathcal{O}(\lambda^3)/\mathcal{O}(\lambda)$

3) Maximize the modulus of the denominator

$$\text{Re}[\mathcal{F}(\lambda_i, \vec{t})]^2 + \text{Im}[\mathcal{F}(\lambda_i, \vec{t})]^2$$

by presampling with 10 different $\lambda_{ij}, i \in \{1, ..., 10\}$

4) Optional: minimize complex argument of $\mathcal{F}$

5) Check if the sign of the imaginary part is still negative in all presampled cases