Numerical Evaluation of Multi-loop Integrals with SecDec 2.1

Sophia Borowka

MPI for Physics, Munich

In collaboration with G. Heinrich

Based on 1303.1157, 1209.6345, 1204.4152

LoopFest XII, Tallahassee, May 14th, 2013

http://secdec.hepforge.org/

Theoretical Predictions in the LHC Era

- A lot of progress has been achieved towards the goal of describing hadron collider processes consistently at NLO

Theoretical Predictions in the LHC Era

- A lot of progress has been achieved towards the goal of describing hadron collider processes consistently at NLO
- Calculations beyond NLO are also progressing well, but automation is difficult, and analytic methods to calculate e.g. two-loop integrals involving massive particles reach their limit

Theoretical Predictions in the LHC Era

- A lot of progress has been achieved towards the goal of describing hadron collider processes consistently at NLO
- Calculations beyond NLO are also progressing well, but automation is difficult, and analytic methods to calculate e.g. two-loop integrals involving massive particles reach their limit
- Numerical methods are in general easier to automate, problems mainly are
  - Extraction of IR and UV singularities
  - Numerical convergence in the presence of integrable singularities (e.g. thresholds)
  - Speed/accuracy

Theoretical Predictions in the LHC Era

- A lot of progress has been achieved towards the goal of describing hadron collider processes consistently at NLO
- Calculations beyond NLO are also progressing well, but automation is difficult, and analytic methods to calculate e.g. two-loop integrals involving massive particles reach their limit
- Numerical methods are in general easier to automate, problems mainly are
  - Extraction of IR and UV singularities (solved with SecDec 1.0)
  - Numerical convergence in the presence of integrable singularities (e.g. thresholds)
  - Speed/accuracy

Theoretical Predictions in the LHC Era

- A lot of progress has been achieved towards the goal of describing hadron collider processes consistently at NLO
- Calculations beyond NLO are also progressing well, but automation is difficult, and analytic methods to calculate e.g. two-loop integrals involving massive particles reach their limit
- Numerical methods are in general easier to automate, problems mainly are
  - Extraction of IR and UV singularities (solved with SecDec 1.0)
  - Numerical convergence in the presence of integrable singularities (e.g. thresholds) (solved with SecDec 2.0)
  - Speed/accuracy

Theoretical Predictions in the LHC Era

- A lot of progress has been achieved towards the goal of describing hadron collider processes consistently at NLO
- Calculations beyond NLO are also progressing well, but automation is difficult, and analytic methods to calculate e.g. two-loop integrals involving massive particles reach their limit
- Numerical methods are in general easier to automate, problems mainly are
  - Extraction of IR and UV singularities (solved with SecDec 1.0)
  - Numerical convergence in the presence of integrable singularities (e.g. thresholds) (solved with SecDec 2.0)
  - Speed/accuracy (improved with SecDec 2.1)

Public codes using the sector decomposition method

Idea and method of sector decomposition introduced by Hepp '66, Denner & Roth '96, Binoth & Heinrich '00

Public codes:

- sector_decomposition (uses GiNaC) C. Bogner & S. 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, M. Tentyukov '08 '09
- SecDec (uses Mathematica, Fortran/C++) J. Carter & G. Heinrich '10; SB, J. Carter, G. Heinrich '12; SB & G. Heinrich '13

Many people are/have been working on PURELY numerical methods, e.g. Anastasiou et al., Becker/Reuschle/Weinzierl et al., Binoth/Heinrich et al., Boughezal/Melnikov/Petriello et al., Czakon/Mitov et al., Freitas et al., Kurihara et al., Nagy/Soper et al., Passarino et al., ...

SecDec 2.1 can tackle ...

SecDec is a tool to numerically compute various sorts of integrals contributing to higher-order computations.

It can tackle:

- General Feynman integrals and more general parametric functions for arbitrary kinematics

Feynman graph

or

parametric function

General Feynman Integral

- **Generic Feynman integrals** in D dimensions at L loops with N propagators to power ν_j of rank R with $N_\nu = \sum_{j=1}^{N} \nu_j$, e.g. scalar multi-loop integral in Feynman parametrization

$$G = \frac{(-1)^{N_\nu}}{\prod_{j=1}^{N} \Gamma(\nu_j)} \Gamma(N_\nu - LD/2) \int_0^\infty \prod_{j=1}^{N} \mathrm{d}x_j \; x_j^{\nu_j - 1} \, \delta(1 - \sum_{l=1}^{N} x_l) \frac{\mathcal{U}^{N_\nu - (L+1)D/2}(\vec{x})}{\mathcal{F}^{N_\nu - LD/2}(\vec{x})}$$

- Extension to physical kinematics including mass thresholds since SecDec 2.0: Limitation of multi-scale integrals to the Euclidean region lifted! SB, Carter, Heinrich '12

NEW in SecDec 2.1

- Computation of contracted **tensor** integrals at in principle arbitrary rank possible SB & Heinrich '13

$$\mathcal{I}_{\mathrm{Rank3}} = \iint \mathrm{d^D}k_1 \, \mathrm{d^D}k_2 \; \frac{p_{1\mu} k_1^\mu k_{1\nu} k_2^\nu}{D_1 D_2 D_3 D_4 D_5}$$

Parametric Functions

A general parametric function can be

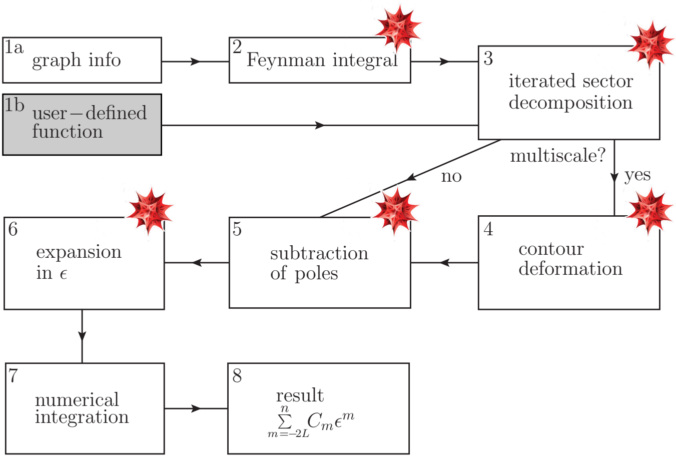
- a phase space integral where IR divergences are regulated dimensionally
- functions similar to hypergeometric functions, e.g.

$$
{}_3F_2(a_1,\dots,a_3;b_1,b_2;\beta) \propto \iint_0^1 \mathrm{d}x\mathrm{d}y\; x^{a_1-1}(1-x)^{b_1-a_1-1}y^{a_2-1}(1-y)^{b_2-a_2-1}(1-\beta xy)^{-a_3}
$$

NEW in SecDec 2.1

- Computation of more general **user-defined polynomial integrals** matching the Feynman loop integral structure SB & Heinrich '12 '13

Operational Sequence of the SecDec 2.1 Program



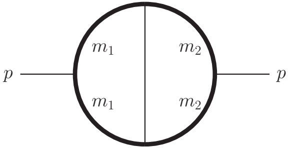
Numerical integration:

CUBA library Hahn et al. '04 '11 or BASES Kawabata '95

New features of the program SecDec Version 2.1

- Computation of contracted **tensor integrals** at in principle arbitrary rank possible
- **User-defined** functions amenable to contour deformation can be inserted and decomposed directly
- **User-friendliness** and **efficiency** improved (e.g. convergence behavior written to result files)

Example I: 2-loop bubble with 2 mass scales



- Scalar:

$$\mathcal{I}_{\mathrm{Scalar}} = \iint \mathrm{d}^{\mathrm{D}}k_1 \, \mathrm{d}^{\mathrm{D}}k_2 \, \frac{1}{D_1 D_2 D_3 D_4 D_5}$$

- Rank 3:

$$\mathcal{I}_{\mathrm{Rank3}} = \iint \mathrm{d}^{\mathrm{D}}k_1 \, \mathrm{d}^{\mathrm{D}}k_2 \, \frac{p_{1\mu} k_1^{\mu} k_{1\nu} k_2^{\nu}}{D_1 D_2 D_3 D_4 D_5}$$

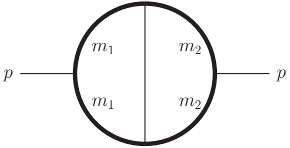
$D_i = (k_i + p_i)^2 - m_i^2 + \mathrm{i}\delta$, k_i - loop momenta,

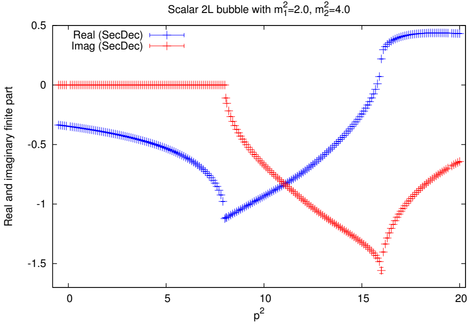
$s_{ij} = (p_i + p_j)^2$, p_i - external momenta

analytical result for scalar 2L Bubble by Bauberger & Böhm '94

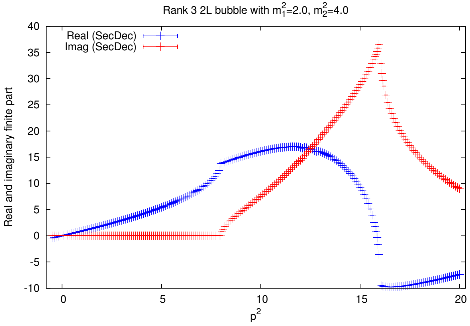
2-loop bubble with 2 mass scales - Results

thresholds at $4 \cdot m_1^2 = 8$ and $4 \cdot m_2^2 = 16$



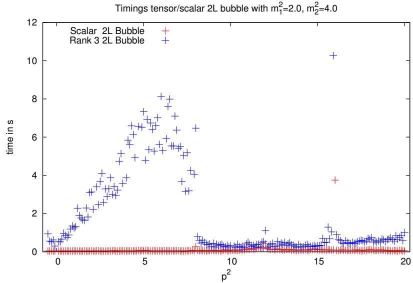


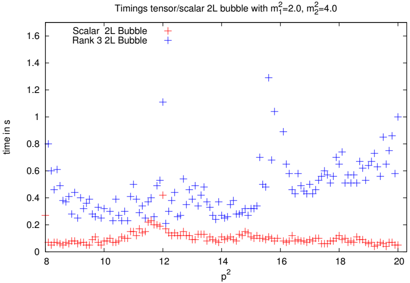
Scalar integral



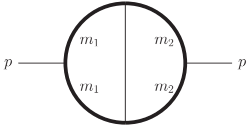
Rank 3

2-loop bubble with 2 mass scales - Timings

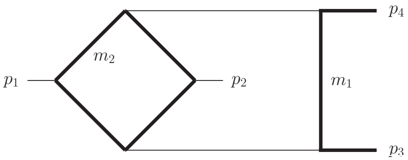




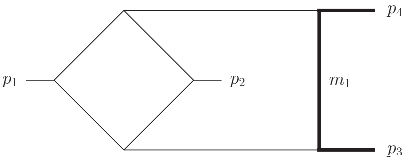
- relative & absolute accuracy 0.1%
- Scalar integral is finite, rank 3 integral has $\mathcal{O}(\epsilon^{-2})$ poles
- Intel Core i7 Processor



Example II:
Massive non-planar 2-loop diagrams for $t\bar{t}$@NNLO



(a) ggtt1



(b) ggtt2

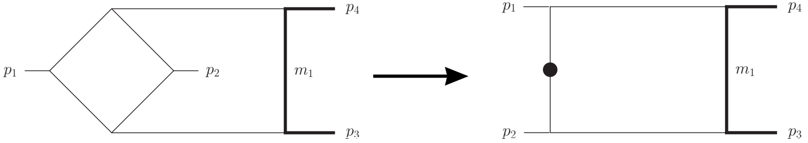
- Diagram *ggtt*1 entering heavy fermionic corrections:
  finite, **no** analytical results available
  $\rightarrow$ easily computable with SecDec
- Diagram *ggtt*2 entering light fermionic corrections:
  leading pole $\mathcal{O}(\epsilon^{-4})$, spurious divergence, analytic result by
  Manteuffel & Studerus '12
  $\rightarrow$ many functions to integrate, cancellations
  $\Rightarrow$ analytic preparation

Analytical manipulations beforehand

Goals for better numerical convergence:

1) decrease number of numerical integration parameters
2) turn linear divergences $x^{-2-\epsilon}$ into logarithmic ones
3) decrease number of functions

Achieving goal 1: Integrate out one loop first



$\rightarrow$ with clever transformations analytical integration of one Feynman parameter is possible

Convert linear divergences into logarithmic ones

Achieving goal 2: Distribute divergences more evenly among Feynman parameters by rearranging them.
How can this be put into practice?
Idea: SB & Heinrich '13

1) **blow down** set of Feynman parameters
2) apply method of **sector decomposition backwards**

Reminder: the method of sector decomposition:

$$\int_0^1 dx_1 \int_0^1 dx_2 \, \frac{1}{(x_1 + x_2)^{2+\epsilon}} \left(\theta(x_1 - x_2) + \theta(x_2 - x_1)\right)$$

$$= \int_0^1 dx_1 \int_0^{x_1} dx_2 \, \frac{1}{(x_1 + x_2)^{2+\epsilon}} + \int_0^1 dx_2 \int_0^{x_2} dx_1 \, \frac{1}{(x_1 + x_2)^{2+\epsilon}}$$

$$= \int_0^1 dx_1 \int_0^1 dt \, \frac{x_1}{(x_1 + x_1 t)^{2+\epsilon}} + \int_0^1 dx_2 \int_0^1 d\tilde{t} \, \frac{1}{x_2^{1+\epsilon}(\tilde{t} + 1)^{2+\epsilon}}$$

Convert linear divergences into logarithmic ones

Achieving goal 2: Distribute divergences more evenly among Feynman parameters by rearranging them.
How can this be put into practice?
Idea: SB & Heinrich '13

1) **blow down** set of Feynman parameters
2) apply method of **sector decomposition backwards**

Reminder: the method of sector decomposition:

$$\int_0^1 dx_1 \int_0^1 dx_2 \; \frac{1}{(x_1+x_2)^{2+\epsilon}} \left(\theta(x_1-x_2)+\theta(x_2-x_1)\right)$$

$$= \int_0^1 dx_1 \int_0^{x_1} dx_2 \; \frac{1}{(x_1+x_2)^{2+\epsilon}} + \int_0^1 dx_2 \int_0^{x_2} dx_1 \; \frac{1}{(x_1+x_2)^{2+\epsilon}}$$

$$= \int_0^1 dx_1 \int_0^1 dt \; \frac{x_1}{(x_1+x_1 t)^{2+\epsilon}} + \int_0^1 dx_2 \int_0^1 d\tilde{t} \; \frac{1}{x_2^{1+\epsilon}(\tilde{t}+1)^{2+\epsilon}} \qquad (1)$$

Convert linear divergences into logarithmic ones

Achieving goal 2: Distribute divergences more evenly among Feynman parameters by rearranging them.
How can this be put into practice?
Idea: SB & Heinrich '13

1) **blow down** set of Feynman parameters
2) apply method of **sector decomposition backwards**

Reminder: the method of sector decomposition:

$$\int_0^1 dx_1 \int_0^1 dx_2 \, \frac{1}{(x_1+x_2)^{2+\epsilon}} \left(\theta(x_1-x_2)+\theta(x_2-x_1)\right)$$

$$= \int_0^1 dx_1 \int_0^{x_1} dx_2 \, \frac{1}{(x_1+x_2)^{2+\epsilon}} + \int_0^1 dx_2 \int_0^{x_2} dx_1 \, \frac{1}{(x_1+x_2)^{2+\epsilon}}$$

$$= \int_0^1 dx_1 \int_0^1 dt \, \frac{x_1}{(x_1+x_1 t)^{2+\epsilon}} + \int_0^1 dx_2 \int_0^1 d\tilde{t} \, \frac{1}{x_2^{1+\epsilon}(\tilde{t}+1)^{2+\epsilon}}$$

Convert linear divergences into logarithmic ones

Achieving goal 2: Distribute divergences more evenly among Feynman parameters by rearranging them.
How can this be put into practice?
Idea: SB & Heinrich '13

1) **blow down** set of Feynman parameters
2) apply method of **sector decomposition backwards**

Sector decomposition backwards to find expression for Eq. (1):

$$\int_0^1 dx_1 \int_0^1 dx_2 \, \frac{\theta(x_1 - x_2) + \theta(x_2 - x_1)}{(x_1 + x_2)^{2+\epsilon}} - \int_0^1 dx_2 \int_0^{x_2} dx_1 \, \frac{1}{(x_1 + x_2)^{2+\epsilon}}$$
$$= \int_0^1 dx_1 \int_0^{x_1} dx_2 \, \frac{1}{(x_1 + x_2)^{2+\epsilon}}$$

Convert linear divergences into logarithmic ones II

$\vec{x}_{jk}$ denotes all Feynman parameters excluding x_j and x_k

Assume $\alpha > 1$ and functions P, Q, R such that a linear divergence appears in x_j in Eq. (2) after sector decomposition

$$\prod_{i=1}^{N} \left\{ \int_0^1 \mathrm{d}x_i \right\} \left[x_j \left(P(\vec{x}_{jk}) + x_k Q(\vec{x}_{jk}) \right) + R(\vec{x}_{jk}) \right]^{-\alpha} \qquad (2)$$

Convert linear divergences into logarithmic ones II

$\vec{x}_{jk}$ denotes all Feynman parameters excluding x_j and x_k

Assume $\alpha > 1$ and functions P, Q, R such that a linear divergence appears in x_j in Eq. (2) after sector decomposition

$$\prod_{i=1}^{N}\left\{\int_0^1 \mathrm{d}x_i\right\}\left[x_j\left(P(\vec{x}_{jk}) + x_k Q(\vec{x}_{jk})\right) + R(\vec{x}_{jk})\right]^{-\alpha} \tag{2}$$

$$= \prod_{i=1}^{N}\left\{\int_0^1 \mathrm{d}x_i\right\}\frac{1}{x_j}\left[x_j P(\vec{x}_{jk}) + x_k Q(\vec{x}_{jk}) + R(\vec{x}_{jk})\right]^{-\alpha}$$

$$- \prod_{i=1}^{N}\left\{\int_0^1 \mathrm{d}x_i\right\}\frac{1}{x_j}\left[x_k\left(x_j P(\vec{x}_{jk}) + Q(\vec{x}_{jk})\right) + R(\vec{x}_{jk})\right]^{-\alpha}$$

Convert linear divergences into logarithmic ones II

$\vec{x}_{jk}$ denotes all Feynman parameters excluding x_j and x_k

Assume $\alpha > 1$ and functions P, Q, R such that a linear divergence appears in x_j in Eq. (2) after sector decomposition

$$\prod_{i=1}^{N} \left\{ \int_0^1 \mathrm{d}x_i \right\} \left[x_j \left(P(\vec{x}_{jk}) + x_k Q(\vec{x}_{jk}) \right) + R(\vec{x}_{jk}) \right]^{-\alpha} \tag{2}$$

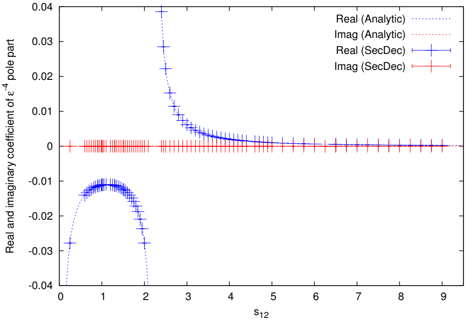
$$= \prod_{i=1}^{N} \left\{ \int_0^1 \mathrm{d}x_i \right\} \frac{1}{x_j} \left[x_j P(\vec{x}_{jk}) + x_k Q(\vec{x}_{jk}) + R(\vec{x}_{jk}) \right]^{-\alpha}$$

$$- \prod_{i=1}^{N} \left\{ \int_0^1 \mathrm{d}x_i \right\} \frac{1}{x_j} \left[x_k \left(x_j P(\vec{x}_{jk}) + Q(\vec{x}_{jk}) \right) + R(\vec{x}_{jk}) \right]^{-\alpha}$$

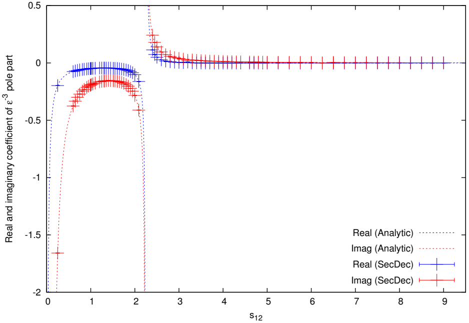
We rest with a logarithmic divergence in x_j.

- Linear divergences can be turned into logarithmic ones
- In the case of *ggtt*2 this leads to a total reduction of number of functions by 2/3 (**goal 3 achieved**)

Result for the non-planar massive two loop diagram ggtt2



Leading pole

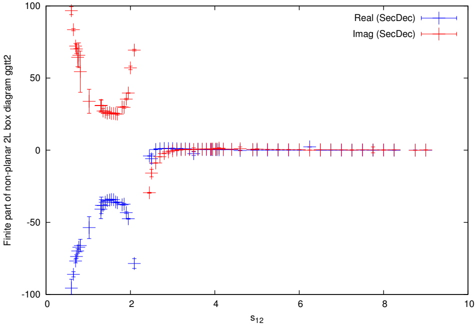


Sub-leading pole

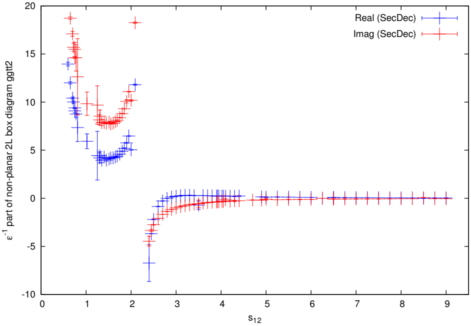
$m_1^2 = 1,\ p_1^2 = p_2^2 = 0,\ p_3^2 = p_4^2 = m_1^2,\ s_{23} = -1.25$

analytic results: Manteuffel & Studerus '12

Result for the non-planar massive two loop diagram ggtt2

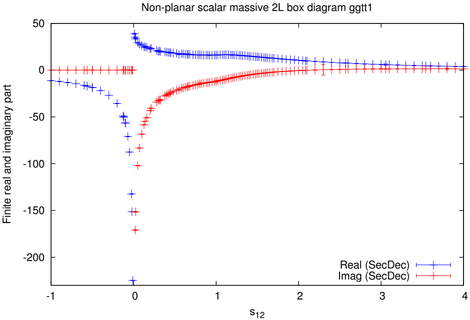


Finite part

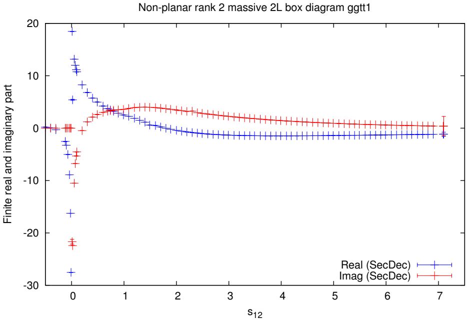


$1/\epsilon$ pole

Results for the non-planar massive two loop diagram ggtt1



Scalar integral



Rank 2

$$m_1^2 = m_2^2 = 1,\ p_1^2 = p_2^2 = 0,\ p_3^2 = p_4^2 = m_1^2,\ s_{23} = -1.25$$

Install SecDec 2.1

- **Download**:
  http://secdec.hepforge.org
- **Install**:
  tar xzvf SecDec.tar.gz
  cd SecDec-2.1
  ./install
- **Prerequisites**:
  Mathematica (version 6 or above), Perl, Fortran and/or C++ compiler

Summary & Outlook

Summary

- SecDec 2.1 is a useful tool to compute various sorts of integrals: general loop integrals, contracted tensor integrals and user-defined polynomial integrals
- We computed non-planar 2-loop 4-point master integrals entering $t\bar{t}$@NNLO computations
- We found a new transformation which allows for the reduction of divergences and the number of functions to integrate

Outlook

- Phenomenological application to two-loop problems
- Combination with new unitarity inspired reduction of 2-loop amplitudes

Backup

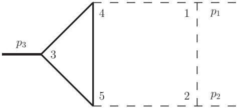
User Input I

- param.input: parameters for integrand specification and numerical integration

```
################ input parameters for sector decomposition ################
#---------------------------------------------------------------------------------------------------
# subdirectory for the mathematica output files (will be created if non-existent) :
# if not specified, a directory with the name of the graph given below will be created by default
subdir=2loop
#---------------------------------------------------------------------------------------------------
# if outputdir is not specified: default directory for
# the output will have integral name (given below) appended to directory above,
# otherwise specify full path for Mathematica output files here
outputdir=
#---------------------------------------------------------------------------------------------------
# graphname (can contain underscores, numbers, but should not contain commas)
graph=P126
#---------------------------------------------------------------------------------------------------
# number of propagators:
propagators=6
#---------------------------------------------------------------------------------------------------
# number of external legs:
legs=3
#---------------------------------------------------------------------------------------------------
# number of loops:
loops=2
#---------------------------------------------------------------------------------------------------
# construct integrand (F and U) via topological cuts (only possible for scalar integrals)
# default is 0 (no cut construction used)
cutconstruct=1
#########################################################
# parameters for subtractions and epsilon expansion
#########################################################
```

User Input II

- template.m: definition of the integrand (Mathematica syntax)




```
(******* USER INPUT for construction of integrand ***************************)

(******* Use with cutconstruct=1 *********************************************)

proplist={{ms[1],{3,4}},{ms[1],{4,5}},{ms[1],{5,3}},
          {0,{1,2}},{0,{1,4}},{0,{2,5}}};

(******* Use with cutconstruct=0 *********************************************)
(*
momlist={k1,k2};
proplist={k1^2-ms[1],(k1+p3)^2-ms[1],(k1-k2)^2-ms[1],
         (k2+p3)^2,(k2+p1+p3)^2,k2^2};
numerator={1};
*)

(******* Propagator powers (optional) ****************************************)

powerlist=Table[1,{i,Length[proplist]}];

(******* On-shell conditions (optional) **************************************)
onshell={ssp[1]->0,ssp[2]->0,ssp[3]->sp[1,2],sp[1,3]->0,sp[2,3]->0};

(******* Set Dimension *******************************************************)

Dim=4-2*eps;
(***************************************************************************)
```

Program Test Run

- ./launch -p param.input -t template.m

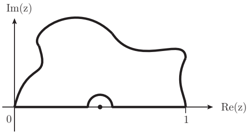
```
********** This is SecDec version 2.0 **********
Authors: Sophia Borowka, Jonathon Carter, Gudrun Heinrich
************************************************
graph = P126
primary sectors 1,2,3,4,5,6, will be calculated
calculating F and U . . .
done
written to /home/pcl335a/sborowka/Work/SecDecBeta/loop/2loop/P126/FUN.m

results of the decomposition will be written to
/home/pcl335a/sborowka/Work/SecDecBeta/loop/2loop/P126
doing sector decomposition . . .
done

working on pole structure: 2 logarithmic poles, 0 linear poles, 0 higher poles
C++ functions created for pole structure 2l0h0
compiling 2l0h0/epstothe0 ...
doing numerical integrations in P126/2l0h0/epstothe0
compiling 2l0h0/epstothe-1 ...
doing numerical integrations in P126/2l0h0/epstothe-1
compiling 2l0h0/epstothe-2 ...
doing numerical integrations in P126/2l0h0/epstothe-2
working on pole structure: 1 logarithmic pole, 0 linear poles, 0 higher poles
C++ functions created for pole structure 1l0h0
compiling 1l0h0/epstothe0 ...
doing numerical integrations in P126/1l0h0/epstothe0
compiling 1l0h0/epstothe-1 ...
doing numerical integrations in P126/1l0h0/epstothe-1
working on pole structure: 0 logarithmic poles, 0 linear poles, 0 higher poles
C++ functions created for pole structure 0l0h0
compiling 1l0h0/epstothe0 ...
doing numerical integrations in P126/0l0h0/epstothe0
Output written to /home/pcl335a/sborowka/Work/SecDecBeta/loop/2loop/P126/P126_pfull.res

To remove intermediate files, execute the command /home/pcl335a/sborowka/Work/SecDecBeta/loop/launchcleanP126
```

Get the Result

- resultfile P126_full.res

```
***********************************************
***OUTPUT: P126 p5 ***************
 point: 7.0
 ext. legs: 0.0 0.0 7.0
 prop. mass: 1.0 0. 0. 0. 0. 0.
Prefactor=-Exp[-2EulerGamma*eps]
***********************************************
****** eps^-2 coeff ******

result       =0.07563683
             +0.1003924148 I
error        =0.000493522517701388
             + 0.00139691015080074 I
CPUtime (all eps^-2 subfunctions)  =0.04
CPUtime (longest eps^-2 subfunction)  =0.01
.
.
.

****** eps^0 coeff ******

result       =0.906978296750816
             -0.908781551612644 I
error        =0.00754504726896407
             + 0.0442867373250588 I
CPUtime (all eps^0 subfunctions)  =2.44
CPUtime (longest eps^0 subfunction)  =0.51

***********************************************

Time taken for decomposition = 2.005725

Total time for subtraction and eps expansion = 41.5057 secs
Time taken for longest subtraction and eps expansion = 17.8613 secs
```

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

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

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

Soper '99

Soper, Nagy, Binoth; Kurihara et al., Anastasiou et al., Freitas et al., Becker et al.