Numerical evaluation of integrals for NNLO calculations with SecDec

Gudrun Heinrich

in collaboration with

Sophia Borowka

Max-Planck-Institute for Physics, Munich

TH/LPCC Institute, October 3, 2012

MAX-PLANCK-GESELLSCHAFT

Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

Motivation

Dimensionally regulated parameter integrals are ubiquitous when calculating higher order corrections

- multi-loop integrals after Feynman parametrisation:

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

Motivation

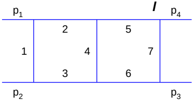
Dimensionally regulated parameter integrals are ubiquitous when calculating higher order corrections

- multi-loop integrals after Feynman parametrisation:

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

example planar double box with $p_1^2 = p_2^2 = p_3^2 = 0,\, p_4^2 \neq 0\;:\; N=7,\, L=2,\, D = 4-2\epsilon$

$$\begin{aligned}\mathcal{F} &= -s\,(x_2x_3x_{4567} + x_5x_6x_{1234} + x_2x_4x_6 + x_3x_4x_5)\\ &\quad -t\,x_1x_4x_7 - p_4^2\,x_7(x_2x_4 + x_5x_{1234}) + \mathcal{U}\sum_i x_i\, m_i^2 - i\,\delta\\ \mathcal{U} &= x_{123}x_{567} + x_4x_{123567}\\ x_{ijk\ldots} &= x_i + x_j + x_k + \ldots\end{aligned}$$



Motivation

Dimensionally regulated parameter integrals are ubiquitous when calculating higher order corrections

- multi-loop integrals after Feynman parametrisation:

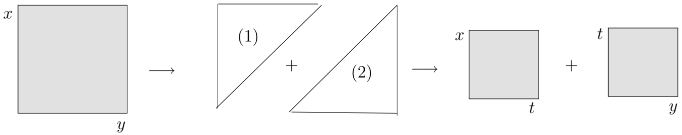
$$G = \frac{(-1)^N}{\prod_{j=1}^N \Gamma(\nu_j)} \int_0^\infty \prod_{j=1}^N dx_j \; x_j^{\nu_j - 1} \, \delta(1 - \sum_{l=1}^N x_l) \; \frac{\mathcal{U}(x)^{N-(L+1)D/2}}{\mathcal{F}(x)^{N-LD/2}}$$

- other multi-dimensional parameter integrals (e.g. phase space integrals, real radiation subtraction terms)

$$\int \mathrm{d}\Phi^{(D)} \, |\mathrm{ME}|^2 \sim \int \mathrm{d}s_{13} \, \mathrm{d}s_{23} \; s_{13}^{-1-\epsilon} \; \frac{\mathcal{F}(s_{13}, s_{23})}{s_{13} + s_{23}}$$

$$\sim \int_0^1 dx \, dy \, x^{-1-\epsilon} \, \frac{\mathcal{F}(x,y)}{x+y}$$

Factorisation of endpoint singularities



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

subst. (1) $y = x\,z$ (2) $x = y\,z$ to remap to unit cube

$$I = \int_0^1 dx\, x^{-1-\epsilon} \int_0^1 dz\, (1+z)^{-1} + \int_0^1 dy\, y^{-1-\epsilon} \int_0^1 dz\, z^{-1-\epsilon} (1+z)^{-1}$$

singularities are disentangled, number of integrals doubled

Sector Decomposition

- allows to extract UV and IR singularities from (dimensionally regulated) parameter integrals in an automated way
- produces a Laurent series in ϵ
- coefficients are finite parameter integrals
  $\Rightarrow$ integrate numerically
- can be applied in various contexts
  (e.g. multi-loop integrals, phase space integrals)

Sector Decomposition

- allows to extract UV and IR singularities from (dimensionally regulated) parameter integrals in an automated way
- produces a Laurent series in ϵ
- coefficients are finite parameter integrals
  $\Rightarrow$ integrate numerically
- can be applied in various contexts
  (e.g. multi-loop integrals, phase space integrals)

history:

- originally devised by K. Hepp 1966

(proof of Bogolyubov-Parasiuk theorem on renormalization)

also used by Denner, Roth 1996

- Binoth, GH 2000 : construction of a general algorithm to isolate infared divergences from multi-loop integrals
- meanwhile applied succesfully in various contexts, in particular NNLO real radiation

[Anastasiou et al, Binoth et al, Boughezal, Czakon, Denner/Pozzorini et al, Passarino et al, Melnikov, Petriello, Smirnov et al, Somogyi, Trocsanyi, Weinzierl, . . .]

Sector Decomposition

public programs:

- sector_decomposition (uses Ginac) Bogner, Weinzierl '07
- FIESTA (uses Mathematica, C) A. Smirnov, V. Smirnov, M. Tentyukov '08, '09
- SecDec (uses Mathematica, perl, Fortran/C++)

  J. Carter, GH '10; S. Borowka, GH '12

  http://secdec.hepforge.org

limitation until recently:
multi-scale integrals limited to Euclidean region
(e.g. no thresholds)

NOW:

Sector Decomposition

public programs:

- sector_decomposition (uses Ginac) Bogner, Weinzierl '07
- FIESTA (uses Mathematica, C) A. Smirnov, V. Smirnov, M. Tentyukov '08, '09
- SecDec (uses Mathematica, perl, Fortran/C++)

  J. Carter, GH '10; S. Borowka, GH '12

  http://secdec.hepforge.org

limitation until recently:
multi-scale integrals limited to Euclidean region
(e.g. no thresholds)

NOW:

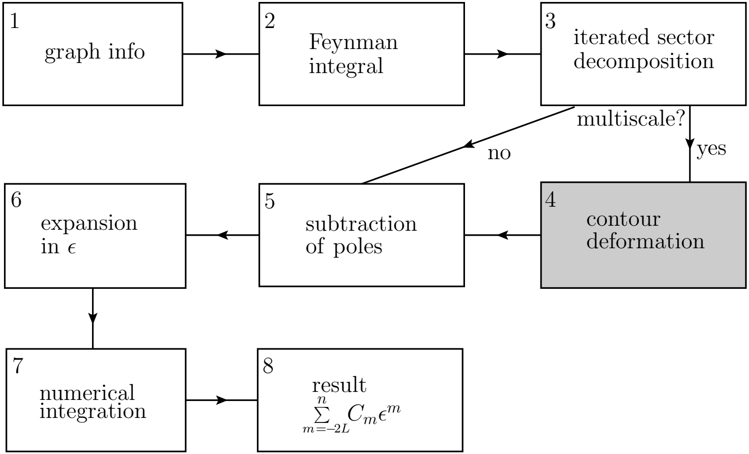
extension of SecDec to general kinematics

method: deformation of integration contour into complex plane

Soper '99, Nagy, Binoth; Kurihara et al, Anastasiou et al, Weinzierl et al.

The program SecDec

basic building blocks (example loop integrals)



Installation and Usage

http://secdec.hepforge.org

SecDec is hosted by Hepforge, IPPP D

- Home
- Subversion
- Tracker
- Wiki

SecDec

Sophia Borowka, Jonathon Carter, Gudrun Heinrich

A program to evaluate dimensionally regulated parameter integrals numerically

Download Program   FAQ   ChangeLog

NEW: Version 2.1 of the program can be downloaded as SecDec-2.1.tar.gz.

Version 2.0 of the program can be downloaded as SecDec-2.0.tar.gz.

To install the program:

- tar xzvf SecDec-2.1.tar.gz
- cd SecDec-2.1
- ./install

Prerequisites: Mathematica (version 6 or higher), Perl, Fortran/C++ compiler

Installation and Usage

- installation:

```
tar xzvf SecDec-2.1.tar.gz
cd SecDec-2.1
./install
```

  - prerequisites:
    Mathematica (version 6 or above), perl, a Fortran/C++ compiler

Installation and Usage

- installation:
  ```
  tar xzvf SecDec-2.1.tar.gz
  cd SecDec-2.1
  ./install
  ```
  - prerequisites:
    Mathematica (version 6 or above), perl, a Fortran/C++ compiler
- user input: two files:
  - parameter.input: parameters for the integrand specification and numerical integration (text file)
  - template_graph.m: definition of the integrand (Mathematica syntax)

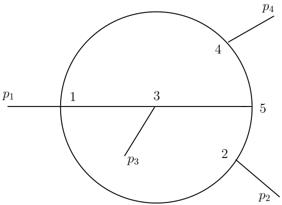
Usage

parameter.input:

```
------------
# graphname (can contain underscores, numbers, but should not contain commas)
graph=Bnp6m1256
#-----------------------------------------------------------------------------------
# number of propagators:
propagators=6
#-----------------------------------------------------------------------------------
# number of external legs:
legs=4
#-----------------------------------------------------------------------------------
# number of loops:
loops=2
#-----------------------------------------------------------------------------------
# construct F,U from cuts
cutconstruct=1
##########################################################
# parameters for subtractions and epsilon expansion
##########################################################
# epsord: level up to which expansion in eps is desired
# (default is epsord=0: Laurent series is cut after finite part eps^0)
# series will be calculated from eps^(-maxpole) to eps^epsord
# note that epsord is negative if only some pole coeffs are required
epsord=0
```

Usage

template_graph.m:



Bnp6massi

```
(* USER INPUT: *)

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

numerator={1};

(* give on-shell conditions *)

onshell=Table[ssp[i]->0,{i,4}];

(* Dimension can be changed, but symbol for epsilon must remain the same *)

Dim=4-2*eps;

```

Usage

to launch one run:

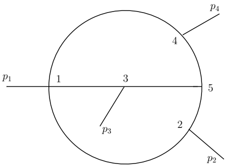
```
./launch -p parameter.input -t template_graph.m
```

to scan over a set of parameter values:

- do decomposition once (exeflag=1 in parameter.input)
- define parameter values in multiparam.input

```
perl multinumerics.pl -p multiparam.input
```

Non-planar four-point functions



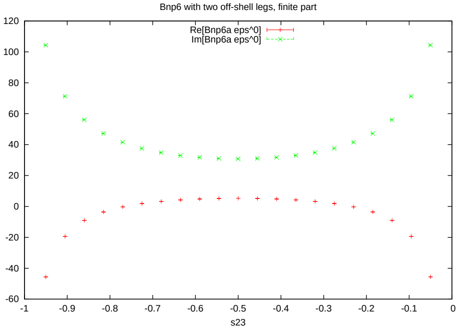
$s_{12} = 3,\ p_1^2, p_2^2 = 1$

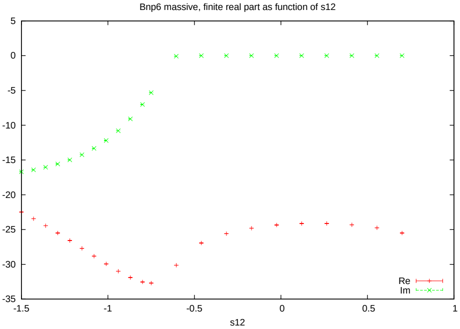
typical timings per PS point: (acc. 1%)

14 *s* far from threshold

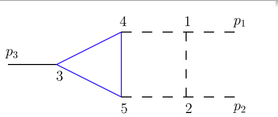
250 *s* close to threshold

$s_{23} = -0.4,\ m_1, m_2, m_5, m_6 = 0.25$





Other examples



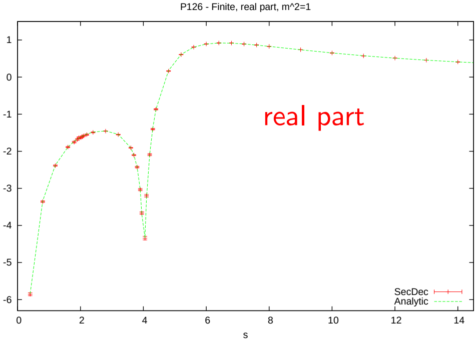
a two-loop triangle

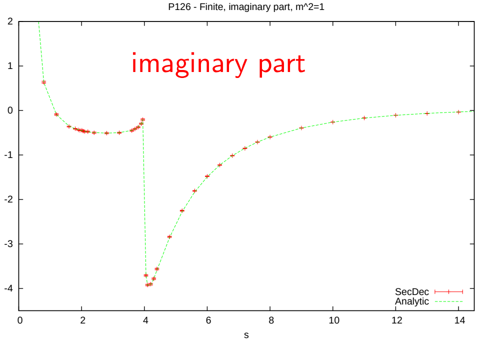
Kotikov '97, Kalmykov '04, Passarino et al '04, Remiddi et al '04

timings:

far from threshold: $\sim 12\,s$

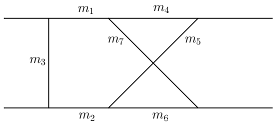
close to threshold: $\sim 14\,s$





s/m^2

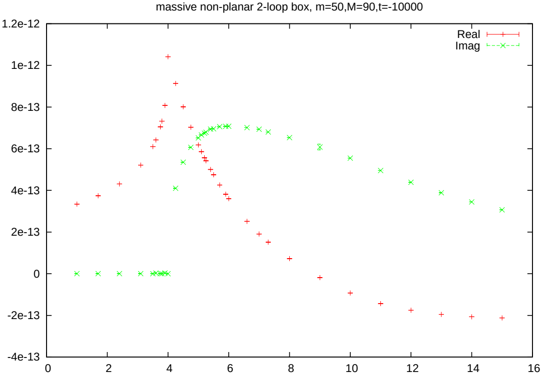
Non-planar four-point functions



compared with numerical result from Yuasa et al, arXiv:1112.0637

$m_1 = m_2 = m_5 = m_6 = m = 50,\; m_3 = m_4 = m_7 = M = 90,\; p_1^2 = p_2^2 = p_3^2 = p_4^2 = m^2,\; s_{23} = -10^4$

timings: far from threshold: $\sim 20\,s$, close to threshold: $\sim 500\,s$



Beyond loop integrals

SecDec turns parameter integrals into a Laurent series

- input:

$$I = \int_0^1 dx_1 \ldots \int_0^1 dx_N \prod_{i=1}^{m} P_i(\vec{x}, \{\alpha\})^{\nu_i}$$

$P_i(\vec{x}, \{\alpha\})$ are polynomial functions of the parameters x_j

SecDec (subdirectory `general`) can deal with:

  - half integer powers ν_i
  - symbolic functions in decomposition stage, specified only at numerical integration stage

- output:

$$I = \sum_{k=-2L}^{n} C_k \, \epsilon^k$$

Example with half integer powers

Hypergeometric functions

$$ {}_pF_{p-1}(a_1,\ldots,a_p;b_1,\ldots,b_{p-1};\beta) = \frac{\Gamma(b_{p-1})}{\Gamma(a_p)\Gamma(b_{p-1}-a_p)} $$
$$ \int_0^1 dz\,(1-z)^{-1-a_p+b_{p-1}}\,z^{-1+a_p}\;{}_{p-1}F_{p-2}(a_1,\ldots,a_{p-1};b_1,\ldots,b_{p-2};\beta) $$

example
${}_4F_3(-4\epsilon,-1/2-\epsilon,-3/2-2\epsilon,1/2-3\epsilon;-1/2+2\epsilon,-1/2+4\epsilon,1/2+6\epsilon;\beta)$

for $\beta=0.5$

Example with half integer powers

```
(* example is Hypergeometric4F3 *)

a1=-4eps;
a2=-1/2-eps;
a3=-3/2-2eps;
a4=1/2-3eps;
b1=-1/2+2eps;
b2=-1/2+4eps;
b3=1/2+6eps;

factorlist={{10,8},{Gamma[a2], -1}, {Gamma[a3], -1}, {Gamma[a4], -1},
 {Gamma[b1], 1}, {Gamma[b2], 1}, {Gamma[b3], 1},
 {Gamma[-a2 + b1], -1}, {Gamma[-a3 + b2], -1},{Gamma[-a4 + b3], -1},
 {1 - z[1], -1 - a2 + b1}, {z[1], -1 + a2},
 {1 - z[2], -1 - a3 + b2}, {z[2], -1 + a3},
 {1 - z[3], -1 - a4 + b3}, {z[3], -1 + a4},
 {1 - beta*z[1]*z[2]*z[3], -a1}};

```

ϵ order	analytic result	numerical result	time taken (secs)
ϵ^0	1	$0.999997 \pm 1.7 \times 10^{-5}$	1.6
ϵ^1	-4.27969	-4.2810 ± 0.0055	54
ϵ^2	-26.6976	-26.625 ± 0.121	90

Summary and Outlook

- SecDec is a flexible tool to calculate multi-loop integrals (or more general parameter integrals) numerically

Summary and Outlook

- SecDec is a flexible tool to calculate multi-loop integrals (or more general parameter integrals) numerically
- publicly available at http://secdec.hepforge.org

Summary and Outlook

- SecDec is a flexible tool to calculate multi-loop integrals (or more general parameter integrals) numerically
- publicly available at http://secdec.hepforge.org
- NEW:
  - loop integrals NOT restricted to Euclidean kinematics anymore
  - graph definition made easy (topology based)
  - extension to wider class of integrals (e.g. Wilson loops)
  - extraction of endpoint singularities from general parametric functions with option to define implicit functions

Summary and Outlook

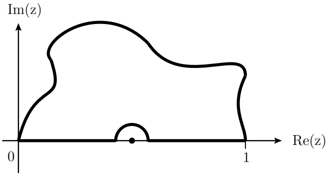
- SecDec is a flexible tool to calculate multi-loop integrals (or more general parameter integrals) numerically
- publicly available at http://secdec.hepforge.org
- NEW:
  - loop integrals NOT restricted to Euclidean kinematics anymore
  - graph definition made easy (topology based)
  - extension to wider class of integrals (e.g. Wilson loops)
  - extraction of endpoint singularities from general parametric functions with option to define implicit functions
- speed still needs to be improved for application to full two-loop amplitudes

Summary and Outlook

- SecDec is a flexible tool to calculate multi-loop integrals (or more general parameter integrals) numerically
- publicly available at http://secdec.hepforge.org
- NEW:
  - loop integrals NOT restricted to Euclidean kinematics anymore
  - graph definition made easy (topology based)
  - extension to wider class of integrals (e.g. Wilson loops)
  - extraction of endpoint singularities from general parametric functions with option to define implicit functions
- speed still needs to be improved for application to full two-loop amplitudes
- implement interface to reduction programs (e.g. Reduze, Fire, Air, ...) to have a widely applicable two-loop tool

Extra slides

Contour deformation



Cauchy: integral over closed contour is zero if no poles are enclosed

$$\int_0^1 \prod_{j=1}^{N} x_j \mathcal{I}(\vec{x}) = \int_0^1 \prod_{j=1}^{N} x_j \left| \frac{\partial z_k(\vec{x})}{\partial x_l} \right| \mathcal{I}(\vec{z}(\vec{x}))$$

$i\,\delta$ prescription for Feynman propagators $\Rightarrow$ $Im(\mathcal{F})$ should be < 0

complexify:

$$\vec{z}(\vec{x}) = \vec{x} - i\,\vec{\tau}(\vec{x})\;,\;\tau_k = \lambda\,x_k(1-x_k)\,\frac{\partial \mathcal{F}(\vec{x})}{\partial x_k}$$

For small λ correct sign of Im part is guaranteed:

$$\mathcal{F}(\vec{z}(\vec{x})) = \mathcal{F}(\vec{x}) - i\,\lambda \sum_j x_j(1-x_j)\left(\frac{\partial \mathcal{F}}{\partial x_j}\right)^2 + \mathcal{O}(\lambda^2)$$

Example for symbolic functions

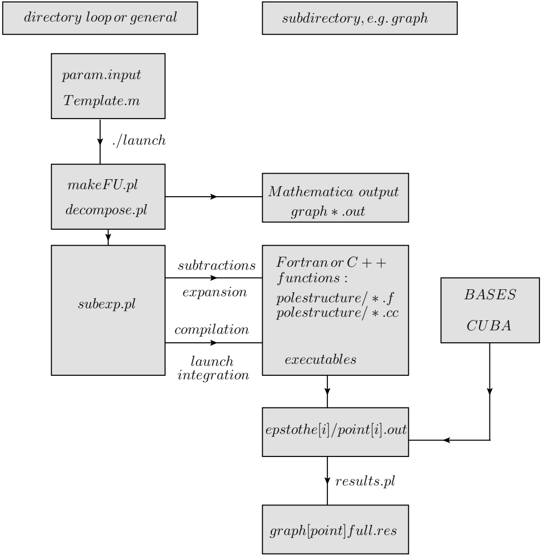
$$f(\vec{x}) = (x_1 + x_2)^{-2-2\epsilon} x_3^{-1-4\epsilon} dum_1(\vec{x}, \beta)^{1+\epsilon} dum_2(\vec{x})^{2-6\epsilon} cut(x_3)$$

$$cut(x_3) \equiv \Theta(x_3 - x_3^{\mathrm{cut}})$$

- x_3^{cut} can be specified by the user in the parameter input file
- the functions dum_i can be defined by the user in a separate file
- $f(\vec{x})$ and dummy functions and can contain symbolic constants
- $\Rightarrow$ flexibility to define e.g. measurement functions

Flowchart of the program SecDec

user input ⟶



New features of SecDec

- **loop** integrals:
  - no restriction on the kinematics!
  - graph definition made easy (topology based)
  - can be parallelized easily (also thanks to `CUBA-3.0`)
  - extension to non-standard loop integrals
    due to option to skip decomposition into N primary sectors

    useful e.g. if some parameter(s) have been integrated out analytically already

New features of SecDec

- **loop** integrals:
  - no restriction on the kinematics!
  - graph definition made easy (topology based)
  - can be parallelized easily (also thanks to `CUBA-3.0`)
  - extension to non-standard loop integrals
    due to option to skip decomposition into N primary sectors
    useful e.g. if some parameter(s) have been integrated out analytically already
- **general** parameter integrals: (extraction of endpoint singularities from general parametric functions:)
  - option to define implicit functions

New features of SecDec

- **loop** integrals:
  - no restriction on the kinematics!
  - graph definition made easy (topology based)
  - can be parallelized easily (also thanks to `CUBA-3.0`)
  - extension to non-standard loop integrals
    due to option to skip decomposition into N primary sectors
    useful e.g. if some parameter(s) have been integrated out analytically already
- **general** parameter integrals: (extraction of endpoint singularities from general parametric functions:)
  - option to define implicit functions
- **both** parts:
  - loops over ranges of numerical values automated