Numerical multi-loop calculations with SecDec

Gudrun Heinrich

in collaboration with

Sophia Borowka

Max-Planck-Institute for Physics, Munich

ACAT Beijing May 17, 2013

MAX-PLANCK-GESELLSCHAFT

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

Motivation

- 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 are limited

Motivation

- 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 are limited
- numerical methods are in general easier to automate, problems mainly are
  1. extraction of IR and UV singularities
  2. numerical convergence in the presence of integrable singularities (e.g. thresholds)
  3. speed/accuracy

Motivation

- 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 are limited
- numerical methods are in general easier to automate, problems mainly are
  1. extraction of IR and UV singularities
  2. numerical convergence in the presence of integrable singularities (e.g. thresholds)
  3. speed/accuracy

- SecDec 1.0 offers a solution to the first problem

Motivation

- 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 are limited
- numerical methods are in general easier to automate, problems mainly are
  1. extraction of IR and UV singularities
  2. numerical convergence in the presence of integrable singularities (e.g. thresholds)
  3. speed/accuracy

- SecDec 1.0 offers a solution to the first problem
- SecDec 2.0 offers a solution to the second problem

Motivation

- 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 are limited
- numerical methods are in general easier to automate, problems mainly are
  1. extraction of IR and UV singularities
  2. numerical convergence in the presence of integrable singularities (e.g. thresholds)
  3. speed/accuracy

- SecDec 1.0 offers a solution to the first problem
- SecDec 2.0 offers a solution to the second problem
- SecDec 2.1 improves the third problem (+new features)

The program SecDec

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

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, NNLO phase space integrals)

Sector Decomposition

history:

- originally devised by K. Hepp 1966

  (proof of Bogolyubov-Parasiuk theorem on renormalization)

  also used by Denner, Roth 1996
- construction of a general algorithm to isolate infared divergences from multi-loop integrals: Binoth, GH 2000
- meanwhile applied succesfully in various contexts, in particular NNLO real radiation

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

Sector Decomposition

public programs:

- 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. Smirnov, V. Smirnov, M. Tentyukov '08, '09
- SecDec (uses Mathematica, Fortran/C++)

  J. Carter, GH '10; S. Borowka, J. Carter, GH '12; S. Borowka, GH '13

  http://secdec.hepforge.org

Parametric integrals

parameter integrals are ubiquitous when calculating higher order corrections

- (multi-) loop Feynman integrals
- subtraction terms for IR singular real radiation at NNLO
- Wilson loop polygons
- ...

Multi-loop integrals

general form of scalar L-loop integral with N propagators (to powers ν_j) after Feynman parametrisation:

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

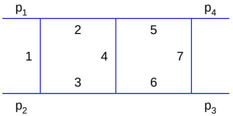
Multi-loop integrals

general form of scalar L-loop integral with N propagators (to powers ν_j) after Feynman parametrisation:

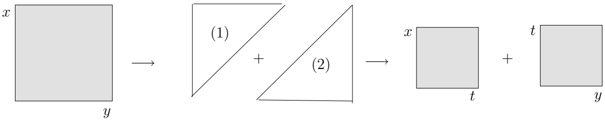
$$G = \frac{(-1)^N}{\prod_{j=1}^{N}\Gamma(\nu_j)} \int_0^\infty \prod_{j=1}^{N} dx_j \; x_j^{\nu_j-1}\, \delta(1-\sum_{l=1}^{N} x_l) \; \frac{\mathcal{U}(x)^{N-(L+1)D/2}}{\mathcal{F}(x)^{N-LD/2}}$$

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

$$\begin{aligned}
\mathcal{F} &= -s\,(x_2 x_3 x_{4567} + x_5 x_6 x_{1234} + x_2 x_4 x_6 + x_3 x_4 x_5) \\
&\quad -t\, x_1 x_4 x_7 - p_4^2\, x_7 (x_2 x_4 + x_5 x_{1234}) + \mathcal{U} \sum_i x_i\, m_i^2 - i\,\delta \\
\mathcal{U} &= x_{123} x_{567} + x_4 x_{123567} \\
x_{ijk\ldots} &= x_i + x_j + x_k + \ldots
\end{aligned}$$



Problem 1: Factorisation of endpoint singularities



UV and IR singularities will show up as endpoint singularities of type

$$I = \int_0^1 dx \int_0^1 dy\, x^{-1-\epsilon} (a_1 x + a_2 y)^{-1} [\underbrace{\Theta(x-y)}_{(1)} + \underbrace{\Theta(y-x)}_{(2)}]$$

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

$$I = \int_0^1 dx\, x^{-1-\epsilon} \int_0^1 dz\, (a_1 + a_2 z)^{-1} + \int_0^1 dy\, y^{-1-\epsilon} \int_0^1 dz\, z^{-1-\epsilon} (a_1 z + a_2)^{-1}$$

singularities are factorized, number of integrals doubled

Problem 2: Dealing with integrable singularities

now consider an integral of type

$$I = \int_0^1 dx \int_0^1 dy \, x^{-1-\epsilon} \, (a_1 \, x - a_2 \, y)^{-1}$$

(e.g. loop integral containing Lorentz invariants with different sign)

- limitation of SecDec 1.0:
  integrand should not change sign
  $\Rightarrow$ multi-scale integrals limited to Euclidean region where integrand is positive definite
- NEW (version $\geq$ 2.0):

Problem 2: Dealing with integrable singularities

now consider an integral of type

$$I = \int_0^1 dx \int_0^1 dy\, x^{-1-\epsilon} (a_1 x - a_2 y)^{-1}$$

(e.g. loop integral containing Lorentz invariants with different sign)

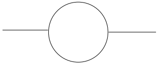
- limitation of SecDec 1.0:
  integrand should not change sign
  $\Rightarrow$ multi-scale integrals limited to Euclidean region where integrand is positive definite
- NEW (version $\geq$ 2.0):

extension of SecDec to general kinematics

method: deformation of integration contour into complex plane
Soper '99, Nagy, Binoth; Kurihara/Kaneko et al, Anastasiou et al, Weinzierl et al.

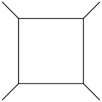
Integrable singularities: examples

$\mathcal{F}_{\rm bubble} = -p^2\, x\,(1-x) + m^2\; - i\,\delta$



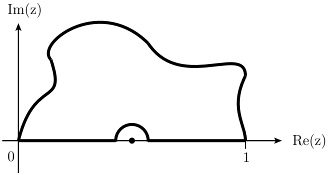
$\mathcal{F}_{\rm bubble}$ vanishing at $x = 1/2, p^2 = 4\, m^2 \Rightarrow$ threshold

$\mathcal{F}_{\rm box} = -s_{12}\, x_1 x_3 - s_{23} x_2 x_4 - i\,\delta$



$\mathcal{F}_{\rm box}$ can vanish inside integration region if s_{12} and s_{23} have different sign
("Euclidean region" if both s_{12} and s_{23} are negative)

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

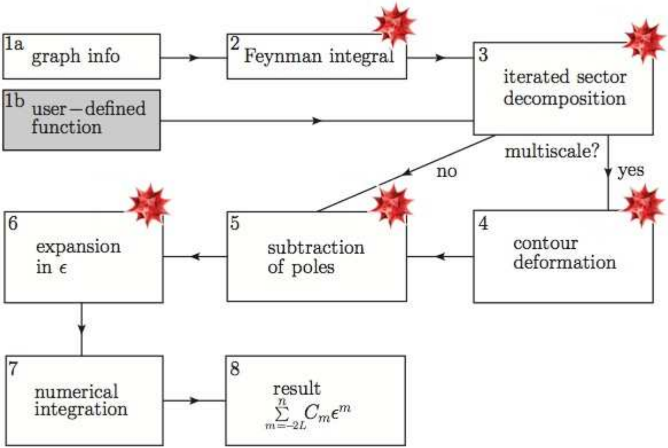
complexify:

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

For small λ correct sign of Im part is guaranteed:

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

The program SecDec



numerical integration: Cuba [T. Hahn et al], Bases [S. Kawabata]

Installation and Usage

- installation:
  ```
  tar xzvf SecDec-2.1.tar.gz
  cd SecDec-2.1
  ./install
  ```
  - prerequisites:
    Mathematica (version 6 or above), perl, Fortran/C++

Installation and Usage

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

Usage: parameter.input

Bubble2Lrank3.input - /Users/gudrun/SecDec-2.1/loop/demos/

File Edit Search Preferences Shell Macro Windows Help

```
################ input parameters for sector decomposition ################
#
# ############### all lines beginning with # are comments ##################
#------------------------------------------------------------------------------------
# subdirectory for the mathematica output files (will be created if non-existent):
subdir=2loop
#------------------------------------------------------------------------------------
# graphname (can contain underscores, numbers, but should not contain commas)
graph=Bubble2Lrank3
#------------------------------------------------------------------------------------
# number of propagators:
propagators=5
#------------------------------------------------------------------------------------
# number of external legs:
legs=2
#------------------------------------------------------------------------------------
# number of loops:
loops=2
#------------------------------------------------------------------------------------
# construct integrand (F and U) via topological cuts (only for scalar integrals)
# default is 0 (no cut construction used)
cutconstruct=0
#########################################################
# 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
#----------------------------------------------------------------------------------
#  flag for prefactor:
```

Usage: graph.m

Bubble2Lrank3.m - /Users/gudrun/SecDec-2.1/loop/demos/

File Edit Search Preferences Shell Macro Windows Help

```
(* USER INPUT: *)

(* give -list of loop momenta (momlist)
          -list of propagators (proplist):
          -numerator: list of scalar products of loop momenta contracted with
           external vectors or loop momenta;
          -list of propagator powers (powerlist), default is 1:
           powers of propagators as listed in proplist         *)

(* example is 2-loop 2-point integral with k1.k2 k1.p1 in the numerator *)

momlist={k1,k2};

proplist={k1^2-ms[1],(k1+p1)^2-ms[1],(k1-k2)^2,(k2+p1)^2-ms[2],k2^2-ms[2]};

numerator={k1*k2,k1*p1};

(* optional: give propagator powers if different from one *)

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

(* optional: give on-shell conditions *)
(* note that in constructing F, (pi+pj)^2 will automatically be called sp[i,j];
    pi^2 will be called ssp[i];
    masses m_i^2 must be called ms[i];
    for the numerator, only the replacements given explicitly in onshell will be made  *)

onshell={};

(* Dim can be changed, but symbol for epsilon must be the same *)
Dim=4-2*eps;
```

Usage

to launch one run:

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

New features of SecDec 2

- **loop** integrals:
  - no restriction on the kinematics!
  - tensors of (in principle) arbitrary rank
  - several options for the user to tune the numerical integration
  - can be parallelized easily (also thanks to `CUBA-3.x` [T. Hahn])
  - extension to non-standard loop integrals
    (useful e.g. if some parameter(s) have been integrated out analytically already)

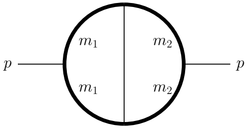
New features of SecDec 2

- **loop** integrals:
  - no restriction on the kinematics!
  - tensors of (in principle) arbitrary rank
  - several options for the user to tune the numerical integration
  - can be parallelized easily (also thanks to `CUBA-3.x` [T. Hahn])
  - extension to non-standard loop integrals
    (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 2

- **loop** integrals:
  - no restriction on the kinematics!
  - tensors of (in principle) arbitrary rank
  - several options for the user to tune the numerical integration
  - can be parallelized easily (also thanks to `CUBA-3.x` [T. Hahn])
  - extension to non-standard loop integrals
    (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

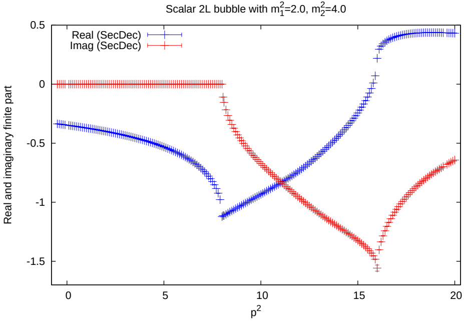
Results: tensor integrals

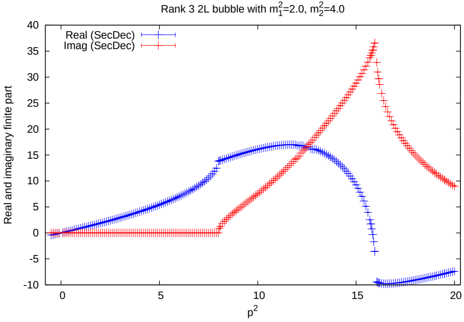


a two-mass two-loop bubble

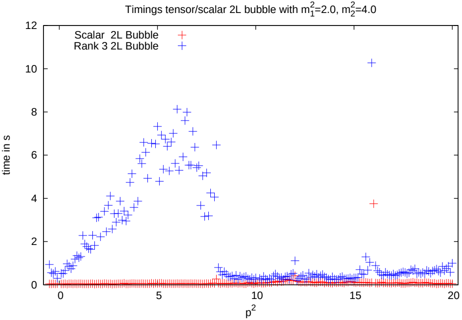
(analytical result:1-dim integral representation)

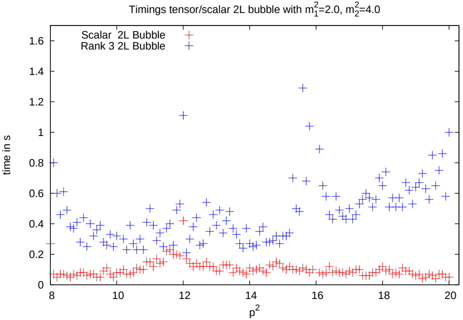
[Bauberger, Böhm '95]





Timings for tensor integrals





relative accuracy 0.1%

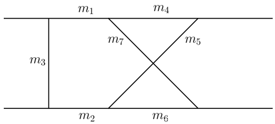
Intel core i7 processor

below 1st threshold at $p^2 = 8$: Imaginary part zero

$\Rightarrow$ artificially increases integration time

good timings for tensor integrals $\Rightarrow$ no need for reduction

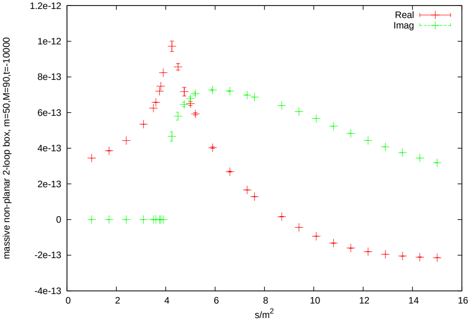
Non-planar four-point functions



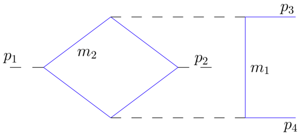
compared with numerical result from Yuasa et al, CPC 183 (2012)

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

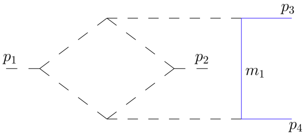
timings: (relative accuracy 10^{-3}): far from threshold: $\sim 20\,s$, close to threshold: $\sim 500\,s$



Non-planar four-point functions for $pp \to t\bar{t}$@NNLO



ggtt1: finite
analytic result unknown
(blue lines are massive)

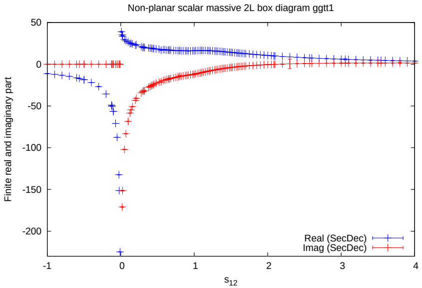


ggtt2: $1/\epsilon^4$ poles
analytical manipulations and new type of transformations to assist SecDec
$\rightarrow$ triggered development of code to allow non-standard input

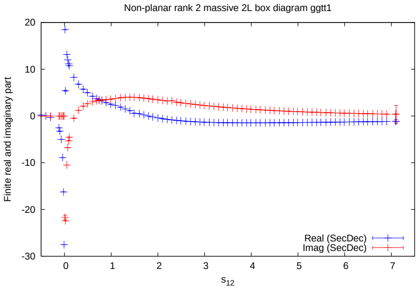
Non-planar four-point functions: ggtt1

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

analytical result unknown



scalar integral



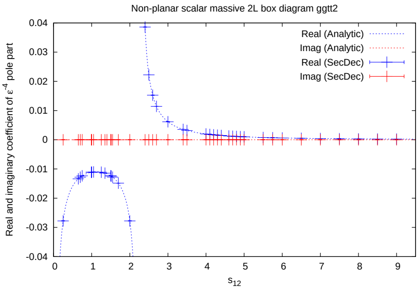
rank two tensor integral
$(\mathcal{N} = k_1 \cdot k_2)$

timings per phase space point: (relative accuracy 10^{-3})

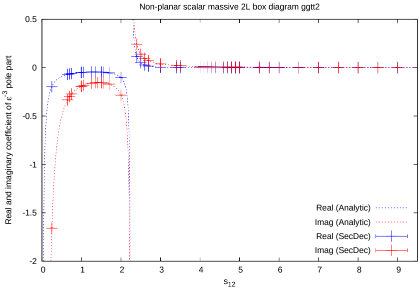
far from threshold: $\mathcal{O}(10\,s)$, very close to threshold: $\mathcal{O}(500\,s)$

Non-planar four-point functions: ggtt2

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



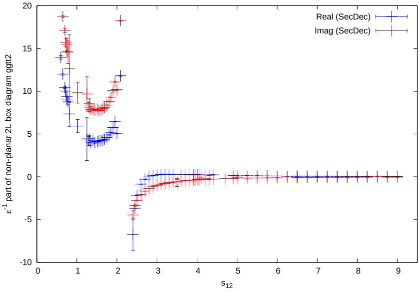
leading pole



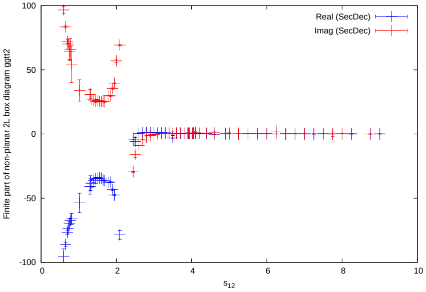
subleading pole

analytical result by Manteuffel, Studerus '12

Non-planar four-point functions: ggtt2



$1/\epsilon$ part



finite part

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
  - automated procedure to optimize the contour deformation

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
  - automated procedure to optimize the contour deformation
  - option to evaluate contracted tensor integrals

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
  - automated procedure to optimize the contour deformation
  - option to evaluate contracted tensor integrals
  - extension to wider class of integrals

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
  - automated procedure to optimize the contour deformation
  - option to evaluate contracted tensor integrals
  - extension to wider class of integrals

OUTLOOK:

- phenomenological application (NNLO)

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
  - automated procedure to optimize the contour deformation
  - option to evaluate contracted tensor integrals
  - extension to wider class of integrals

OUTLOOK:

- phenomenological application (NNLO)
- combination with unitarity-inspired reduction of two-loop amplitudes