Feynman Integral Evaluation by a Sector decomposiTion Approach (FIESTA)

Mikhail Tentyukov

TTP Karlsruhe

ACAT2010, 22-27 Feb., Jaipur, India

FIESTA

Feynman **I**ntegral **E**valuation by a **S**ector decomposi**T**ion **A**pproach

A.V. Smirnov, M.N. Tentyukov, [arXiv:0807.4129]

A.V. Smirnov, V.A. Smirnov and M. Tentyukov, [arXiv:0912.0158]

http://www-ttp.particle.uni-karlsruhe.de/∼asmirnov/FIESTA.htm

is a software system for automatic numerical evaluation of coefficients of the ε-expansion of a scalar dimensionally regularized Feynman Integral with arbitrary indices:

$$F(a_1,\ldots,a_n) \;=\; \int\cdots\int \frac{\mathrm{d}^d k_1 \ldots \mathrm{d}^d k_l}{E_1^{a_1}\ldots E_n^{a_n}}\,,$$

$d = 4 - 2\varepsilon$

a_n – indices (not necessary positive)

l – the number of loops

$1/E_n$ – propagators: $1/(-p^2+m^2-i0)$

Usage

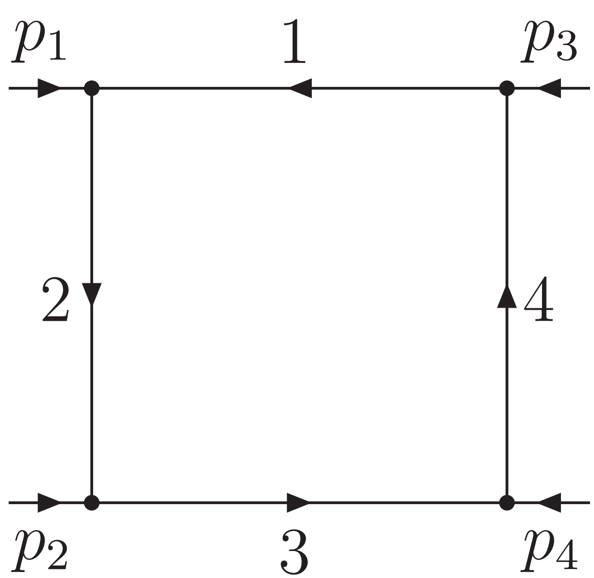
Interface in Mathematica:

`SDEvaluate[UF[loop_momenta,propagators,subst],indices,order]`

- `loop_momenta` – a set of loop momenta
- `propagators` – set of propagators
- `subst` – set of substitutions for external momenta, masses and other values
- `indices` – set of indices
- `order` – required order of the ε-expansion

Usage

Massless on-shell box diagram with $S = 3$ and $T = 1$:



1. $-k^2$
2. $-(k + p_1)^2$
3. $-(k + p_1 + p_2)^2$
4. $-(k + p_1 + p_2 + p_4)^2$

with the following substitutions:

$p_1^2 \to 0, \quad p_2^2 \to 0, \quad p_4^2 \to 0, \quad p_1 p_2 \to -S/2, \quad p_2 p_4 \to -T/2, \quad p_1 p_4 \to (S+T)/2, \quad S \to 3, \quad T \to 1$

Example

```
SDEvaluate[UF[loop_momenta,propagators,subst], indices,order]
```

```
In[1]:=  << FIESTA_2.0.0.m
FIESTA, version 2.0.0
In[2]:= SDEvaluate[UF[{k},{-k^2,-(k+p1)^2,-(k+p1+p2)^2,
        -(k+p1+p2+p4)^2},{p1^2->0,p2^2->0,p4^2->0,
        p1 p2->-S/2,p2 p4->-T/2,p1 p4->(S+T)/2,S->3,
        T->1}],{1,1,1,1},0]
. . .
Total time used: 1.52991 seconds.
                                            -6
                   1.33333   -0.73241 + 5. 10   pm46
Out[3]= -4.3865 + ------- + ----------------------- + 0.000013 pm47
                       2                ep
                     ep
```

From the analytical answer:

$$-4.3864908 + 1.3333333\frac{1}{\varepsilon^2} - 0.7324081\frac{1}{\varepsilon}$$

That's all!

The software structure

- MATHEMATICA user interface
- MATHEMATICA part: treating negative indices, sector decomposition, resolution of singularities and preparing expressions to be integrated
- Memory problems resolution: using the hash - based database TokyoCabinet with help of the C-program QLink
- Numerical integration via C-program CIntegrate by means of the Thomas Hahn CUBA library.

Parallelization

By default the system works in the sequential mode.

Specify of the number of copies of `CIntegrate` to be launched: NumberOfLinks=8; to perform numerical evaluation in parallel. Parallelization on cluster: CIntegrate -slave FILENAME.

The `Mathematica` part distributes the integration tasks between those copies and collects the result, preparing the expression of the next order at the same time.

Specify of the number of Mathematica subkernels: NumberOfSubkernels=8; to perform `Mathematica` part in parallel.

Sector decomposition I

- Initially sector decomposition was used for proving theorems on Feynman integrals.

Sector decomposition I

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2000. Then there were many attempts to use sector decomposition for numerical evaluation.

Sector decomposition I

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2000. Then there were many attempts to use sector decomposition for numerical evaluation.
- First public code by Bogner, Weinzierl — 2007

Sector decomposition I

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2000. Then there were many attempts to use sector decomposition for numerical evaluation.
- First public code by Bogner, Weinzierl — 2007
- Public code FIESTA by Smirnov, Tentyukov — 2008

Sector decomposition I

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2000. Then there were many attempts to use sector decomposition for numerical evaluation.
- First public code by Bogner, Weinzierl — 2007
- Public code FIESTA by Smirnov, Tentyukov — 2008
- FIESTA 2 by A. Smirnov, V. Smirnov, Tentyukov — 2009

Sector decomposition II

α - representation and some usual tricks, see e.g. A.V. Smirnov, M.N. Tentyukov, [arXiv:0807.4129] :

$$F(a_1 ... a_n) = \int_{x_j=0}^{1} dx_1 \ldots dx_n \left(\prod_{j=1}^{n'} x_j^{a_j-1} \right) \frac{U^{A-(l+1)d/2}}{F^{A-ld/2}}$$

- n number of propagators
- a_i indices
- l number of loops
- n' number of positive indices minus one.
- $A = \sum_{i=1}^{n} a_i$
- U and F are constructively defined polynomials of x_i

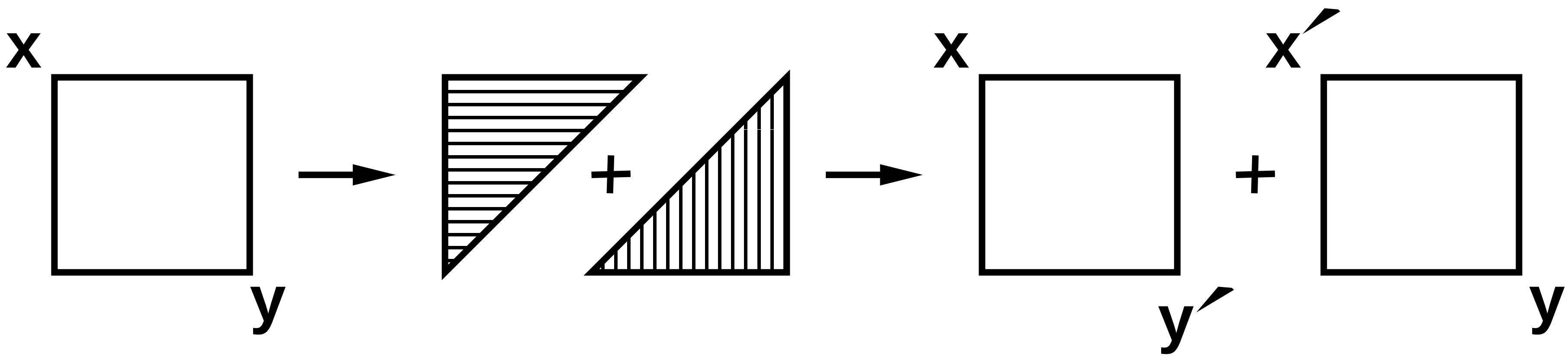
No zeroes in U F $\rightarrow$ singularities $x_j^{a_j-1}$ integrated analitically.

Sector decomposition III

Problems with overlapping singularities:

$$\int_0^1 \frac{dxdyf(x,y)}{(x+y)^{2-\varepsilon}}$$

with $\varepsilon \to 0$ problems in both $x = 0$ and $y = 0$ *simultaneously*.



$$\int_0^1 dx \int_0^x dy + \int_0^1 dy \int_0^y dx \quad \begin{pmatrix} y \to xy' \\ x \to x'y \end{pmatrix} = \int_0^1 \frac{dxdy' f(x, xy')}{x^{1-\varepsilon}(1+y')^{2-\varepsilon}} + \int_0^1 \frac{dx' dy f(x'y, y)}{y^{1-\varepsilon}(1+x')^{2-\varepsilon}}$$

Singularities are factorized, can be integrated analitycally

Sector decomposition strategies I

$$F(a_1 ... a_n) = \int_{x_j=0}^{1} dx_1 \ldots dx_n \left(\prod_{j=1}^{n'} x_j^{a_j-1} \right) \frac{U^{A-(l+1)d/2}}{F^{A-ld/2}}$$

No zeroes in $U\ F \rightarrow$ singularities $x_j^{a_j-1}$ integrated analitically.

In general: sector decomposition process.

Single step is not enough, iterative process

At each step the choice is not unique.

The way sector decomposition is performed is called a

sector decomposition strategy

Sector decomposition strategies II

Different strategies:

Sector decomposition strategies II

Different strategies:

- Strategy A (Spyvakovsky, 1983)

Sector decomposition strategies II

Different strategies:

- Strategy A (Spyvakovsky, 1983)
- Strategy B and C by Bogner and Weinzierl (2008)

Sector decomposition strategies II

Different strategies:

- Strategy A (Spyvakovsky, 1983)
- Strategy B and C by Bogner and Weinzierl (2008)
- Strategies by Binoth and Heinrich (X, ...), 2000-...

Sector decomposition strategies II

Different strategies:

- Strategy A (Spyvakovsky, 1983)
- Strategy B and C by Bogner and Weinzierl (2008)
- Strategies by Binoth and Heinrich (X, ...), 2000-...
- Strategy by Smirnov S in FIESTA (2008)

Sector decomposition strategies II

Different strategies:

- Strategy A (Spyvakovsky, 1983)
- Strategy B and C by Bogner and Weinzierl (2008)
- Strategies by Binoth and Heinrich (X, ...), 2000-...
- Strategy by Smirnov S in FIESTA (2008)
- Hepp sectors and Speer sectors (strategy SS)also can be represented as iterative strategies (A.Smirnov, V. Smirnov, 2009)

Sector decomposition strategies II

Different strategies:

- Strategy A (Spyvakovsky, 1983)
- Strategy B and C by Bogner and Weinzierl (2008)
- Strategies by Binoth and Heinrich (X, ...), 2000-...
- Strategy by Smirnov S in FIESTA (2008)
- Hepp sectors and Speer sectors (strategy SS)also can be represented as iterative strategies (A.Smirnov, V. Smirnov, 2009)
- Geometrical approach to sector decomposition Kaneko, Ueda (2009) – not implemented in FIESTA yet.

Sector decomposition strategies III

The default strategy S was developed by A. Smirnov, it is guaranteed to terminate.

Diagram	A	B	C	S	X
Box	12	12	12	12	12
Double box	755	586	586	362	293
Triple box	M	114256	114256	22657	10155
D420	8898	564	564	180	F

“F”: sector decomposition fails in an infinite loop.
“M”: memory overflow on a 8Gb machine.
Experimenting with `STRATEGY=STRATEGY_X` (not guaranteed to terminate).
Speer sectors strategy SS leads to the same sectors as S but much faster.

Singularities and numerical instability

$$\int_0^1 x^{-1+\varepsilon} g(x) = \frac{g(0)}{\varepsilon} + \int_0^1 x^{-1+\varepsilon} O(x)$$

Near 0, $x^{-1+\varepsilon}$ is large, the reminder $x^{-1+\varepsilon}O(x)$ is small but might be a difference of rather huge numbers.

Trivial example:

$$\frac{x+1}{x} - \frac{1-x+x^2+2x^3}{x(1+x^2)}$$

`Together[%]=` $\frac{2-x^2}{1+x^2}$

Singularities and numerical instability

$$\int_0^1 x^{-1+\varepsilon} g(x) = \frac{g(0)}{\varepsilon} + \int_0^1 x^{-1+\varepsilon} O(x)$$

Near 0, $x^{-1+\varepsilon}$ is large, the reminder $x^{-1+\varepsilon}O(x)$ is small but might be a difference of rather huge numbers.

Trivial example:

$$\frac{\log(1+x)+1}{x} - \frac{1-x+x^2+2x^3}{x(1+x^2)}$$

`Together[%]=` $\frac{x-2x^3+\log(1+x)+x^2\log(1+x)}{x(1+x^2)}$

Singularities and numerical instability

$$\int_0^1 x^{-1+\varepsilon} g(x) = \frac{g(0)}{\varepsilon} + \int_0^1 x^{-1+\varepsilon} O(x)$$

Near 0, $x^{-1+\varepsilon}$ is large, the reminder $x^{-1+\varepsilon}O(x)$ is small but might be a difference of rather huge numbers.

Trivial example:

$$\frac{\log(1+x)+1}{x} - \frac{1-x+x^2+2x^3}{x(1+x^2)}$$

$$\frac{1-x^2}{1+x^2} + \frac{\log(1+x)}{x}$$

Singularities and numerical instability

$$\int_0^1 x^{-1+\varepsilon} g(x) = \frac{g(0)}{\varepsilon} + \int_0^1 x^{-1+\varepsilon} O(x)$$

Near 0, $x^{-1+\varepsilon}$ is large, the reminder $x^{-1+\varepsilon}O(x)$ is small but might be a difference of rather huge numbers.

Two methods:

- `Integration By Parts` until all negative x powers disappear – too complex expressions;
- `Multiple Precision Arithmetics` in those points where it is necessary (default).

Numerical integration I

Integration by the CUBA library.

- Up to Terabyte expressions! No chance to compile!
- Usual approach (MAPLE's `codegen`) does not help.
- Our own interpreter

Translation to a bytecode (one time): expression $\rightarrow$ sequence of triples:

`1+x[1]+x[1]*x[3]` $\longrightarrow$

```
1: '+', 1, x[1]
2: '*', x[1], x[3]
3: '+', ^1, ^2
```

Numerical integration II

Integration by the CUBA library.

- Up to Terabyte expressions! No chance to compile!
- Usual approach (MAPLE's `codegen`) does not help.
- Our own interpreter

Translation to a bytecode (one time): expression $\rightarrow$ sequence of triples, Prefix optimization:

```
(1+x[1]+x[2])                 1: '+',  1, x[1]
     *(1+x[1]+x[3])   -->     2: '+', ^1, x[2]
                              3: '+', ^1, x[3]
                              4: '*', ^2, ^3
```

(Plus some heuristics)

Interpretation of this bytecode (many times).

Other facilities

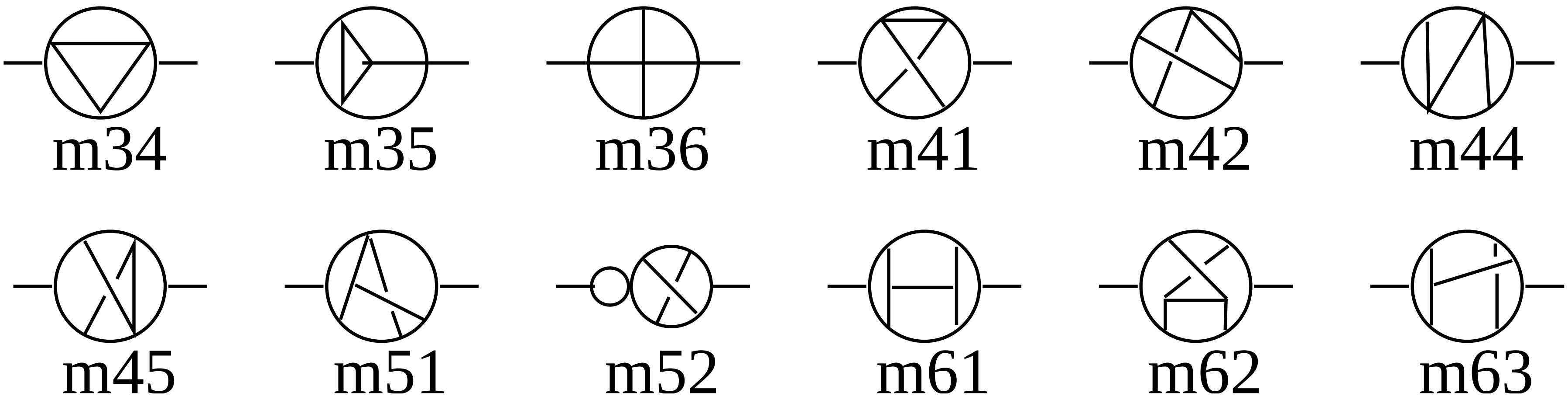
1. Evaluation of parametric integrals (for example, connected to Wilson loops, V. Del Duca, C. Duhr and V. A. Smirnov, arXiv:0911.5332):
   `SDEvaluateDirect[x,{P[1],P[2],...,P[n]}, {d[1],d[2],...,d[n]}, order]`
2. Asymptotic expansion:
   `SDExpand[UF[loop_momenta,propagators,replacements], indices,order,expand_var,expand_degree]`
3. Pole analysis (for which values of d does the integral have poles?):
   `SDAnalyze[{U,F,h},degrees,order,dmin,dmax]`

Applications

- Systematic crosschecks in the calculation of the three-loop static quark potential (Smirnov, Smirnov, Steinhauser)
- Three-loop non-planar vertex integrals at the threshold (Marquard, Steinhauser)
- Evaluation of terms of an asymptotic expansion in a limit of momenta and masses
- Crosschecks for analytical results for the four-loop massless propagator integrals (Baikov, Chetyrkin) and the numerical results for the next terms in ε - expansion, see the next slide
- Evaluation and numerical checks of the master integrals for the three-loop formfactors (Baikov, Chetyrkin, Smirnov, Smirnov, Steinhauser; Huber, Heinrich, Kosower, V.Smirnov)
- Using `SDAnalyze[]` to analytically evaluate the three remaining master integrals for the three-loop formfactors (Lee, Smirnov, Smirnov)
- Parametric integrals contributing to Wilson loops by Del Duca, Duhr, Smirnov

Four-loop propagators

Most complicated 4-loop progataor masters (Baikov,Chetyrkin):

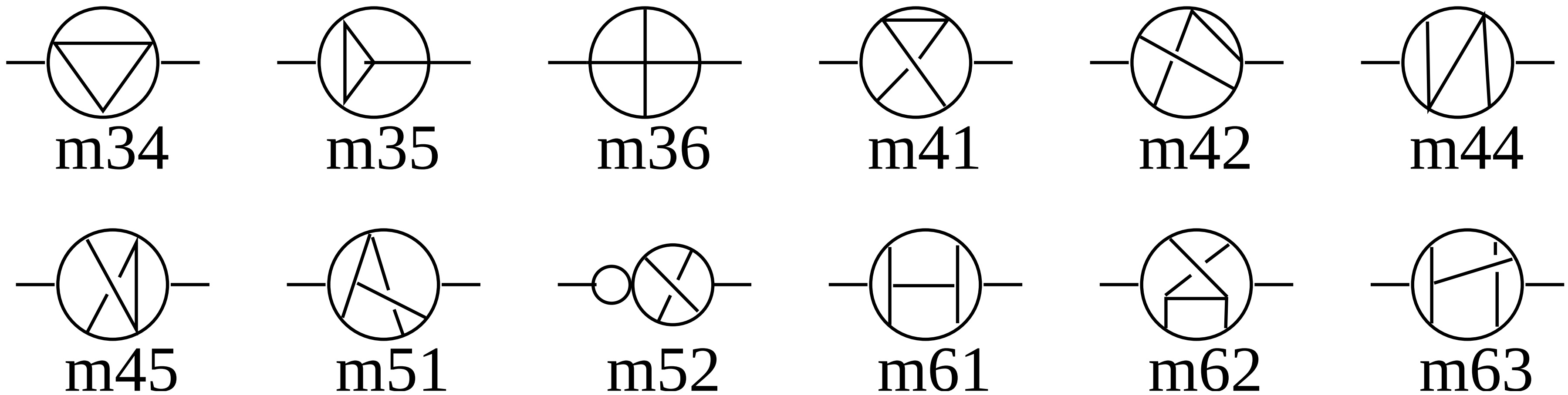


Int. id:	Degree of ε:	Exact Value:	Cuba Vegas 500 000 result:	Cuba Vegas 1 500 000 result:
m61	ε^{-1}	-10.3692776	-10.36941 $\pm$ 0.00011	-10.36931 $\pm$ 0.00006
	ε^{0}	-70.99081719	-70.989 $\pm$ 0.002	-70.990 $\pm$ 0.0011
	ε^{1}	-21.663005	-21.633 $\pm$ 0.023	-21.650 $\pm$ 0.013
	ε^{2}	—	2832.86 $\pm$ 0.17	2832.69 $\pm$ 0.096([a])

[a] Calculated with the FORTRAN VEGAS using 1 550 000 samples.

Four-loop propagators

Most complicated 4-loop progataor masters (Baikov,Chetyrkin):



Int. id:	Degree of ε:	Cuba Vegas 500 000 time:	Cuba Vegas 1 500 000 time:
m61	ε^{-1}	2262.86s	3495.28s
	ε^{0}	15242.10s	39673.48s
	ε^{1}	61481.36s	162453.52s
	ε^{2}	202018.31s	1794640.00s([a])
	Total time:	768727.00s	—

8-core E5472 3.0 GGh, 4.6TB,

1794640 sec = 21 days.

[a] FORTRAN VEGAS using 1 550 000 samples.

Conclusions

- FIESTA has many other tunable parameters, see our paper [arXiv:0912.0158]
- FIESTA is perfectly parallelized on multi-kernel computers. A triple box diagram up to the finite part was evaluated in 62.4 hours on 8 cores Intel Xeon (CPU E5472, 3.00GHz, 1600FSB, 2x6MB L2 cache, 32 GB RAM).
- Numerical part is parallized on a cluster.
- Practical sector decomposition was introduced in papers of T. Binoth and G. Heinrich, Nucl. Phys. B, 585 (2000) 741 [hep-ph/0004013]; Nucl. Phys. B, 680 (2004) 375 [hep-ph/0305234]; Nucl. Phys. B, 693 (2004) 134 [hep-ph/0402265]. A good review see G. Heinrich, Int. J. of Modern Phys. A, 23 (2008) 10 [arXiv:0803.4177]
- The first public algorithm has been published by C. Bogner and S. Weinzierl, Comput. Phys. Comm., 178 (2008) 596 [arXiv:0709.4092]. Algorithm of G. Heinrich *et al* is in preparation for publication.