

New Methods for Feynman Integrals

V.A. Smirnov

Scientific Research Computer Center of Moscow State University

FIESTA

FIESTA

Not a car by Ford

FIESTA

Not a novelle by Hemingway

FIESTA

but...

FIESTA

but...

Feynman Integral Reduction by a Sector decomposition
Approach

Contents:

- Introduction to sector decomposition
- Sector decomposition strategies
- ϵ -expansion and numerical integration problems
- FIESTA internal structure

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

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation:
Binoth, Heinrich — 2002

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2002
- Public code by Bogner, Weinzierl — 2007

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2002
- Public code by Bogner, Weinzierl — 2007
- Public code FIESTA by Smirnov, Tentyukov — 2008

Sector decomposition introduction: Feynman integrals:

$$G(a_1, \dots, a_n) = \int \cdots \int \frac{d^d k_1 \dots d^d k_h}{E_1^{a_1} \dots E_n^{a_n}}.$$

$d = 4 - 2\epsilon$; the denominators E_r are either quadratic or linear with respect to the loop momenta $p_i = k_i$, $i = 1, \dots, h$ or to the independent external momenta $p_{h+1} = q_1, \dots, p_{h+N} = q_N$ of the graph.

Alpha representation:

$$G_{\Gamma}(q_1, \dots, q_n; d; a_1, \dots, a_L) = \frac{i^{a+h(1-d/2)} \pi^{hd/2}}{\prod_l \Gamma(a_l)} \\ \times \int_0^{\infty} \dots \int_0^{\infty} \prod_l \alpha_l^{a_l-1} U^{-d/2} e^{iF/U - i \sum m_l^2 \alpha_l} \mathbf{d}\alpha_1 \dots \mathbf{d}\alpha_L,$$

where L and h is, respectively, the number of lines (edges) and loops (independent circuits) of the graph, $a = \sum a_l$, and U and F a some polynomials of alpha-parameters constructively defined by the Feynman diagram.

Evaluation problems:

Evaluation problems:

- Multi-dimensional integrals depending on ϵ

Evaluation problems:

- Multi-dimensional integrals depending on ϵ
Evaluate them numerically?

Evaluation problems:

- Multi-dimensional integrals depending on ϵ
Evaluate them numerically?
- Need to expand in ϵ , but there are singularities

Evaluation problems:

- Multi-dimensional integrals depending on ϵ
Evaluate them numerically?
- Need to expand in ϵ , but there are singularities

Integration region decomposition, variable replacements and singularities resolutions have to be made before the expansion

First step: *primary sectors*

l -th primary sector is a subregion where $\alpha_i \leq \alpha_l, i \neq l$.

The integration domain falls apart into L primary sectors, the integral is equal to the sum of integrals over primary sectors.

Using the homogeneity properties of the functions in the representation and explicitly integrating over α_l we represent the l -th integral as

$$G^{(l)} = \frac{\left(i\pi^{d/2}\right)^h \Gamma(a - hd/2)}{\prod_l \Gamma(a_l)} \int_0^1 \dots \int_0^1 t_l^{a_l-1} \dots t_l^{a_l-1} \dots t_L^{a_L-1} \\ \times \frac{\hat{\mathcal{U}}^{a-(h+1)d/2}}{\hat{F}^{a-hd/2}} dt_1 \dots dt_{l-1} dt_{l+1} \dots dt_L ,$$

where

$$\hat{U} = U(t_1, \dots, t_{l-1}, 1, t_{l+1}, \dots, t_L),$$

$$\hat{F}_0 = F(t_1, \dots, t_{l-1}, 1, t_{l+1}, \dots, t_L),$$

$$\hat{F} = \left[-\hat{F}_0 + \hat{U} \left(\sum_{l=1}^{L-1} m_l^2 \prod_{l'=l}^{L-1} t_{l'} + m_L^2 \right) \right]$$

Now the integration is over the unit hypercube. Each term is of the form

$$c(\epsilon) \int_0^1 \cdots \int_0^1 \prod_l^{L-1} x_l^{p_l} \frac{\hat{U}^{a(\epsilon)}}{\hat{F}^{b(\epsilon)}} \mathbf{d}x_1 \cdots \mathbf{d}x_{L-1},$$

where \hat{U} and \hat{F} are polynomials of integration variables, p_l are integers and a , b and c are functions of ϵ .

Now the integration is over the unit hypercube. Each term is of the form

$$c(\epsilon) \int_0^1 \cdots \int_0^1 \prod_l^{L-1} x_l^{p_l} \frac{\hat{U}^{a(\epsilon)}}{\hat{F}^{b(\epsilon)}} \mathbf{d}x_1 \cdots \mathbf{d}x_{L-1},$$

where \hat{U} and \hat{F} are polynomials of integration variables, p_l are integers and a , b and c are functions of ϵ .

- Easier to integrate numerically

Now the integration is over the unit hypercube. Each term is of the form

$$c(\epsilon) \int_0^1 \cdots \int_0^1 \prod_l^{L-1} x_l^{p_l} \frac{\hat{U}^{a(\epsilon)}}{\hat{F}^{b(\epsilon)}} \mathbf{d}x_1 \cdots \mathbf{d}x_{L-1},$$

where \hat{U} and \hat{F} are polynomials of integration variables, p_l are integers and a , b and c are functions of ϵ .

- Easier to integrate numerically
- But the problem of singularities persists

The goal of the sector decomposition method is to decompose each of the integration domains into subsectors and perform variables replacements in each of those sectors such that

The goal of the sector decomposition method is to decompose each of the integration domains into subsectors and perform variables replacements in each of those sectors such that

- The integration domain in new variables is again a unit hypercube

The goal of the sector decomposition method is to decompose each of the integration domains into subsectors and perform variables replacements in each of those sectors such that

- The integration domain in new variables is again a unit hypercube
- The integrals obtained have only ϵ -singularities arising from the $x_i^{(\epsilon-n)}$ -like factors in the integrands

The goal of the sector decomposition method is to decompose each of the integration domains into subsectors and perform variables replacements in each of those sectors such that

- The integration domain in new variables is again a unit hypercube
- The integrals obtained have only ϵ -singularities arising from the $x_i^{(\epsilon-n)}$ -like factors in the integrands
- The number of sectors is as small as possible

Sector decomposition:

Sector decomposition:

- Sector decomposition usually defined iteratively;

Sector decomposition:

- Sector decomposition usually defined iteratively;
- A sector decomposition strategy is an instruction how to perform one sector decomposition step + a stop condition;

Sector decomposition:

- Sector decomposition usually defined iteratively;
- A sector decomposition strategy is an instruction how to perform one sector decomposition step + a stop condition;
- A sector decomposition strategy is said to be *guaranteed to terminate* if one can prove that after a finite number of steps all resulting sectors and integrands satisfy the stop condition;

Let us suppose that the functions U and F have no negative terms. This is not a rare situation, for example it is true if all the kinematic invariants are non-positive. Then one can derive a sufficient condition to assure that there are no singularities: it is simply the existence of a constant among the summands, e.g.:

$$F = 1 + x_1 + x_2 + \dots$$

It this case there are multiple sector decomposition strategies guaranteed to terminate.

Sector decomposition strategies

The function U or F defines *the set of weights* — the multidegrees of monomials.

Sector decomposition strategies

The function U or F defines *the set of weights* — the multidegrees of monomials.

$$x_1^2 + x_1x_2 + x_2$$

Sector decomposition strategies

The function U or F defines *the set of weights* — the multidegrees of monomials.

$$x_1^2 + x_1x_2 + x_2$$

\leftrightarrow

$$\{\{2, 0\}, \{1, 1\}, \{0, 1\}\}$$

Sector decomposition strategies

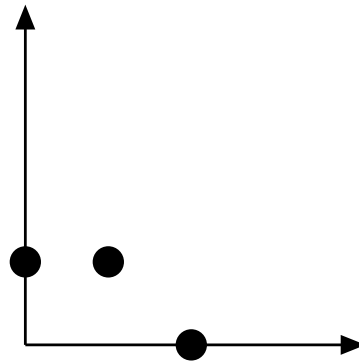
The function U or F defines *the set of weights* — the multidegrees of monomials.

$$x_1^2 + x_1 x_2 + x_2$$

\leftrightarrow

$$\{\{2, 0\}, \{1, 1\}, \{0, 1\}\}$$

or graphically



Sector decomposition strategies

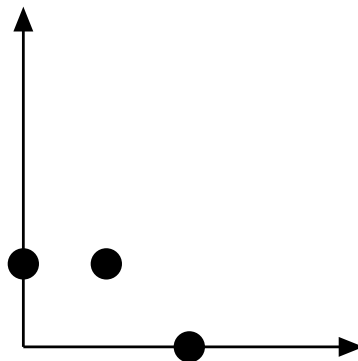
The function U or F defines *the set of weights* — the multidegrees of monomials.

$$x_1^2 + x_1x_2 + x_2$$

\leftrightarrow

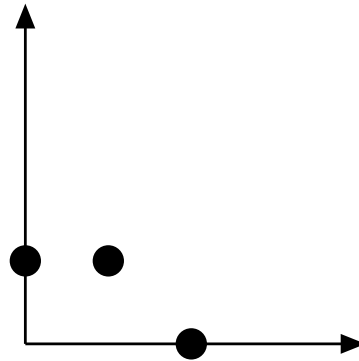
$$\{\{2, 0\}, \{1, 1\}, \{0, 1\}\}$$

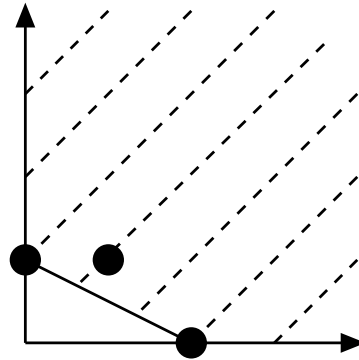
or graphically



There is a constant among the summands if and only if there is a unique minimum point in the set of weights (factorizing a monomial is equivalent to moving the picture)

The set of weights M





Its positive convex hull Δ — we are interested in its minimal points M_C

What sector decompositions can be done?

Main idea:

A subset $I = \{i_1, \dots, i_m\}$ of $\{1, \dots, n\}$, m sectors are defined by

$$S_l = \{(x_1, \dots, x_n) \in \mathcal{U} \mid x_{i_l} \geq x_{i_k} \forall i_k \in I\},$$

($l = 1 \dots m$).

What sector decompositions can be done?

Main idea:

A subset $I = \{i_1, \dots, i_m\}$ of $\{1, \dots, n\}$, m sectors are defined by

$$S_l = \{(x_1, \dots, x_n) \in \mathcal{U} \mid x_{i_l} \geq x_{i_k} \forall i_k \in I\},$$

($l = 1 \dots m$). The variable replacement in S_l is defined by

$$x_i = x'_i \quad \forall i \notin I$$

$$x_{i_l} = x'_{i_l}$$

$$x_{i_k} = x'_{i_l} x'_{i_k} \quad \forall i \in I, k \neq l$$

It is easy to verify that the integration region in the new variables x' is again a unitary cube.

Strategy A:

Strategy A:

- $W = \{v_i\}$ — vectors, connecting M_C

Strategy A:

- $W = \{v_i\}$ — vectors, connecting M_C
- $L(v) = \max v_i - \min v_i$

Strategy A:

- $W = \{v_i\}$ — vectors, connecting M_C
- $L(v) = \max v_i - \min v_i$
- $N(v) = \#\{i, v_i = \max v_i \text{ or } \min v_i\}$

Strategy A:

- $W = \{v_i\}$ — vectors, connecting M_C
- $L(v) = \max v_i - \min v_i$
- $N(v) = \#\{i, v_i = \max v_i \text{ or } \min v_i\}$
- We find a vector v_0 with the lowest pair $L(v), N(v)$ under the lexicographic ordering

Strategy A:

- $W = \{v_i\}$ — vectors, connecting M_C
- $L(v) = \max v_i - \min v_i$
- $N(v) = \#\{i, v_i = \max v_i \text{ or } \min v_i\}$
- We find a vector v_0 with the lowest pair $L(v), N(v)$ under the lexicographic ordering
- Now I is formed by the minimal and maximal coordinates of this vector

Strategy A:

- $W = \{v_i\}$ — vectors, connecting M_C
- $L(v) = \max v_i - \min v_i$
- $N(v) = \#\{i, v_i = \max v_i \text{ or } \min v_i\}$
- We find a vector v_0 with the lowest pair $L(v), N(v)$ under the lexicographic ordering
- Now I is formed by the minimal and maximal coordinates of this vector
- It can be shown that the triple $\{\#M_C, L, N\}$ decreases under the lexicographic ordering

Different strategies:

Different strategies:

- Strategy A (explained above)

Different strategies:

- Strategy A (explained above)
- Strategy B and C by Bogner and Weinzierl

Different strategies:

- Strategy A (explained above)
- Strategy B and C by Bogner and Weinzierl
- Strategies by Binoth and Heinrich

Different strategies:

- Strategy A (explained above)
- Strategy B and C by Bogner and Weinzierl
- Strategies by Binoth and Heinrich
- Strategy S in FIESTA (going to explain it)

Different strategies:

- Strategy A (explained above)
- Strategy B and C by Bogner and Weinzierl
- Strategies by Binoth and Heinrich
- Strategy S in FIESTA (going to explain it)
- Hepp sectors and Speer sectors also can be represented as iterative strategies (to be explained in a future paper by A.Smirnov, V. Smirnov)

We are extending the set of possible regions and variable replacements the following way: for any vector (v_1, \dots, v_n) in the positive quadrant with at least two non-zero coordinates we consider the set $I = \{i | v_i \neq 0\}$ and separate \mathcal{U} into m parts by

$$S_l = \{(x_1, \dots, x_n) \in \mathcal{U} | x_{i_l}^{a_{i_l}} \geq x_{i_k}^{a_{i_k}} \forall i_k \in I\},$$

where $\{i_1, \dots, i_m\} = I$ and the exponents a_i are defined by...

where $\{i_1, \dots, i_m\} = I$ and the exponents a_i are defined by

$$\begin{pmatrix} a_{i_1} \\ a_{i_2} \\ a_{i_3} \\ \vdots \\ a_{i_m} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 0 \end{pmatrix}^{-1} \begin{pmatrix} v_{i_1} \\ v_{i_2} \\ v_{i_3} \\ \vdots \\ v_{i_m} \end{pmatrix} \quad (-3)$$

The variable replacement in S_l is defined by

$$x_i = x'_i \forall i \notin I$$

$$x_{i_l} = x_{i_l}^{v_{i_l}}$$

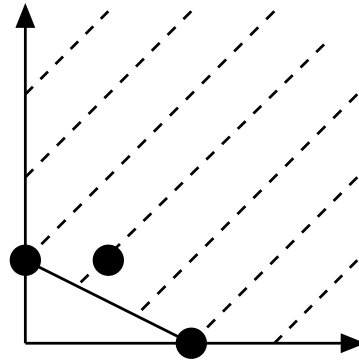
$$x_{i_k} = x_{i_l}^{v_{i_k}} x'_{i_k} \quad \forall i \in I, k \neq l.$$

It is only a little bit more complicated to verify that the integration region in the variables x' is still the unite hypercube.

What is the reason to use these replacements?

If you choose the vector v orthogonal to the face visible from the origin, the face collapses after such a sector decomposition step, and we result in less minimal points.

What is the reason to use these replacements?



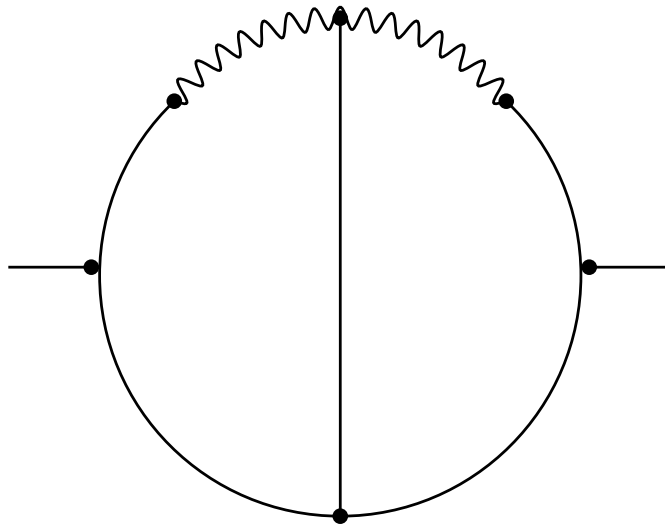
If you choose the vector v orthogonal to the face visible from the origin, the face collapses after such a sector decomposition step, and we result in less minimal points.

Strategies comparison:

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

--	--	--	--	--

D420 is a diagram contributing to the 2-loop static quark potential with the following set of propagators:
 $\{-k^2, -(k - q)^2, -l^2, -(l - q)^2, -(k - l)^2, -vk, -vl\}$, where k and l are loop momenta, $q^2 = -1$, $qv = 0$ and $v^2 = 1$.



ϵ -expansion and numerical integration problems

After the sector decomposition the integrands have the following form:

$$\int_{x_j=0}^1 dx_i \dots dx_n \left(\prod_{j=1}^n x_j^{a_j-1+b_j\epsilon} \right) Z,$$

where Z has no singularities.

Let us assume that $a_i \leq 0$ for some i and treat the integrand as a function

$$x_i^{a_i-1+b_i\epsilon} Y(x_i)$$

We replace $x^{a-1+b\epsilon}Y(x)$ by

We replace $x^{a-1+b\epsilon}Y(x)$ by

$$Y(0)x^{a-1+b\epsilon} + Y'(0)x^{a+b\epsilon} + \dots + \frac{1}{(-a)!}Y^{(-a)}(0)x^{-1+b\epsilon} +$$

We replace $x^{a-1+b\epsilon}Y(x)$ by

$$Y(0)x^{a-1+b\epsilon} + Y'(0)x^{a+b\epsilon} + \dots + \frac{1}{(-a)!}Y^{(-a)}(0)x^{-1+b\epsilon} +$$

$$x^{a-1+b\epsilon}(Y(a) - Y(0) - Y'(0)x - \dots - \frac{1}{(-a)!}Y^{(-a)}(0)x^{-a})$$

$$Y(0)x^{a-1+b\epsilon} + Y'(0)x^{a+b\epsilon} + \dots + \frac{1}{(-a)!} Y^{(-a)}(0)x^{-1+b\epsilon}$$

One integration can be taken analytically!

$$Y(0)x^{a-1+b\epsilon} + Y'(0)x^{a+b\epsilon} + \dots + \frac{1}{(-a)!}Y^{(-a)}(0)x^{-1+b\epsilon}$$

One integration can be taken analytically!

$$x^{a-1+b\epsilon}(Y(a) - Y(0) - Y'(0)x - \dots - \frac{1}{(-a)!}Y^{(-a)}(0)x^{-a})$$

Can be numerically integrated but still might result in problems!

FIESTA internal structure

- Mathematica + C
- QLink
- CIntegrate - Vegas
- Parallelization and sums of integrations
- Integration cuts

Mathematica:

- User-friendly
- Easy handling of lists and algebraic objects

Mathematica:

- User-friendly
- Easy handling of lists and algebraic objects

C:

- Fast
- Full control of memory usage

QLink: tool useful for anybody working with Mathematica

QLink: tool useful for anybody working with Mathematica
F[1]="fadaewf", F[2]="wrgqwrgaw",... with lots of values

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

After initialization the commands can be replaced:

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

After initialization the commands can be replaced:

In Mathematica $F[1]=\text{"dfgsjfl"}$ with

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

After initialization the commands can be replaced:

```
QPut["F", "1", "dfgsjfk"]
```

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

After initialization the commands can be replaced:

```
QPut["F", "1", "dfgsjfk"]
```

In Mathematica result=F[1] with

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

After initialization the commands can be replaced:

```
QPut["F", "1", "dfgsjfk"]
```

```
result=QGet["F", "1"]
```

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

After initialization the commands can be replaced:

```
QPut["F", "1", "dfgsjfk"]
```

```
result=QGet["F", "1"]
```

In Mathematica `F[1]=.` with

QLink: tool useful for anybody working with Mathematica

An external key - value database on disk

After initialization the commands can be replaced:

```
QPut["F", "1", "dfgsjfk"]
```

```
result=QGet["F", "1"]
```

```
QRemove["F", "1"]
```


External integration on C:

External integration on C:

- Faster than in Mathematica

External integration on C:

- Faster than in Mathematica
- Higher precision than in Mathematica, algorithm choice

External integration on C:

- Faster than in Mathematica
- Higher precision than in Mathematica, algorithm choice
- Parallelization - no licences counted

External integration on C:

- Faster than in Mathematica
- Higher precision than in Mathematica, algorithm choice
- Parallelization - no licences counted
- Might be parallelized even to multiple computers via TCP

External integration on C:

- Faster than in Mathematica
- Higher precision than in Mathematica, algorithm choice
- Parallelization - no licences counted
- Might be parallelized even to multiple computers via TCP
- We do not compile - a personal interpreter

Integrating sums or summing integrals?

Integrating sums or summing integrals?

Integrating sums:

- Some terms might be grouped
- Potentially better precision

Integrating sums or summing integrals?

Integrating sums:

- Some terms might be grouped
- Potentially better precision

Summing integrals:

- The algorithm can adapt to all the peaks
- Good for parallelization

Log with an example (box):

```
In[1]:= << FIESTA_1.0.0.m
```

```
FIESTA, version 1.0.0
```

```
In[2]:= UsingQLink=False;
```

```
In[3]:= 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]
```

Log with an example (box):

```
In[1]:= << FIESTA_1.0.0.m
```

```
FIESTA, version 1.0.0
```

```
In[2]:= UsingQLink=False;
```

```
In[3]:= 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]
```

```
External integration ready! Use CIntegrate to perform calls
```

```
FIESTA, version 1.0.0
```

```
UsingC: True
```

```
NumberOfLinks:1
```

```
UsingQLink: False
```

```
IntegrationCut: 0
```

```
IfCut: 0.
```

```
Strategy: STRATEGY_S
```

```
Integration has to be performed up to order 0
```

```
Sector decomposition.....0.0138 seconds; 12 sectors.
```

```
Variable substitution.....0.0055 seconds.
```

Decomposing ϵ -independent term.....0.0025 seconds
Pole resolution.....0.0081 seconds; 40 terms.
Expression construction.....0.0021 seconds.
Replacing variables.....0.0063 seconds.
Epsilon expansion.....0.0123 seconds.
Expanding.....0.0002 seconds.
Counting variables: 0.0002 seconds.
Preparing integration string.....0.0004 seconds.
Terms of order -2: 8 (1-fold integrals).
Numerical integration: 8 parts; 1 links;
Integrating.....0.106322 seconds; returned answer:
1.333333

Integration of order -2: 1.333333
(1.333333)/ ϵ^2
Expanding.....0.0005 seconds.
Counting variables: 0.0005 seconds.
Preparing integration string.....0.001 seconds.
Terms of order -1: 28 (2-fold integrals).

```

Numerical integration: 12 parts; 1 links;
Integrating.....0.409080 seconds; returned answer:
                                -2.065743 +- 5.*^-6
Integration of order -1: -2.065743 +- 5.*^-6
(1.333333)/ep^2 + (-0.73241 +- 5.*^-6)/ep
Expanding.....0.0008 seconds.
Counting variables: 0.0009 seconds.
Preparing integration string.....0.0022 seconds.
Terms of order 0: 40 (2-fold integrals).
Numerical integration: 12 parts; 1 links;
Integrating.....0.862786 seconds; returned answer:
                                -3.417375 +- 0.000012
Integration of order 0: -3.417375 +- 0.000012
(1.333333)/ep^2 + (-0.73241 +- 5.*^-6)/ep +
                                (-4.386496 +- 0.000013)
Total time used: 1.52991 seconds.
                                -6
                                1.33333      -0.73241 + 5. 10      pm46
Out[3]= -4.3865 + ----- + -----
                                2                        ep

```

To summarize:

To summarize:

- Completely automatic — only the list of propagators has to be on input

To summarize:

- Completely automatic — only the list of propagators has to be on input
- Parallelization

To summarize:

- Completely automatic — only the list of propagators has to be on input
- Parallelization
- Ability to handle large amounts of data

To summarize:

- Completely automatic — only the list of propagators has to be on input
- Parallelization
- Ability to handle large amounts of data
- Methods to handle the numerical instability

To summarize:

- Completely automatic — only the list of propagators has to be on input
- Parallelization
- Ability to handle large amounts of data
- Methods to handle the numerical instability
- NO complex numbers support at the moment

Everything available at <http://science.sander.su>