New Methods for Feynman Integrals

A.V. Smirnov

Scientific Research Computing 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
- Dealing with negative terms
- FIESTA internal structure
- Summary of new features in FIESTA 2
- Applications

- 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 — 2000

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2000
- 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 — 2000
- Public code by Bogner, Weinzierl — 2007
- Public code FIESTA by Smirnov, Tentyukov — 2008

- Initially sector decomposition was used for proving theorems on Feynman integrals.
- Practical sector decomposition for numerical evaluation: Binoth, Heinrich — 2000
- Public code by Bogner, Weinzierl — 2007
- Public code FIESTA by Smirnov, Tentyukov — 2008
- Planning a new release of FIESTA by A. Smirnov, V. Smirnov, Tentyukov — 2009

Sector decomposition introduction:

Feynman integrals:

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

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

Alpha representation:

$$G_\Gamma(q_1,\ldots,q_n;\ \ d\ \ ;a_1\ldots,a_L) = \frac{\mathrm{i}^{a+h(1-d/2)}\pi^{hd/2}}{\prod_l \Gamma(a_l)}$$

$$\times \int_0^\infty \ldots \int_0^\infty \prod_l \alpha_l^{a_l-1} U^{-d/2} \mathrm{e}^{\mathrm{i}F/U - \mathrm{i}\sum m_l^2\alpha_l} \mathrm{d}\alpha_1 \ldots \mathrm{d}\alpha_L\ ,$$

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

$$G^{(l)} = \frac{\left(\mathrm{i}\pi^{d/2}\right)^h \Gamma(a-hd/2)}{\prod_l \Gamma(a_l)} \int_0^1 \ldots \int_0^1 t_l^{a_l-1} \ldots \cancel{t}_l^{a_l-1} \ldots t_L^{a_L-1}$$
$$\times \frac{\hat{\mathcal{U}}^{a-(h+1)d/2}}{\hat{\mathcal{F}}^{a-hd/2}} \mathrm{d}t_1 \ldots \mathrm{d}t_{l-1} \mathrm{d}t_{l+1} \ldots \mathrm{d}t_L \ ,$$

where

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

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

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

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

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

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

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

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

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

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.:

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

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

Example

$$\int_0^1 \int_0^1 \frac{1}{(x+y)^{2-\epsilon}} dydx =$$

Example

$$\int_0^1 \int_0^1 \frac{1}{(x+y)^{2-\epsilon}} dydx =$$

$$2\int_0^1 \int_0^x \frac{1}{(x+y)^{2-\epsilon}} dydx =$$

Example

$$\int_0^1 \int_0^1 \frac{1}{(x+y)^{2-\epsilon}} dydx =$$

$$2 \int_0^1 \int_0^x \frac{1}{(x+y)^{2-\epsilon}} dydx =$$

$$2 \int_0^1 \int_0^x \frac{x}{(x+xz)^{2-\epsilon}} dzdx =$$

where $y = xz$

Example

$$\int_0^1 \int_0^1 \frac{1}{(x+y)^{2-\epsilon}} dy dx =$$

$$2 \int_0^1 \int_0^x \frac{1}{(x+y)^{2-\epsilon}} dy dx =$$

$$2 \int_0^1 \int_0^x \frac{x}{(x+xz)^{2-\epsilon}} dz dx =$$

where $y = xz$

$$2 \int_0^1 \int_0^1 x^{-1+\epsilon} \frac{1}{(1+z)^{2-\epsilon}} dz dx$$

Different strategies:

Different strategies:

- Strategy A (Spyvakovsky, 1983)

Different strategies:

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

Different strategies:

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

Different strategies:

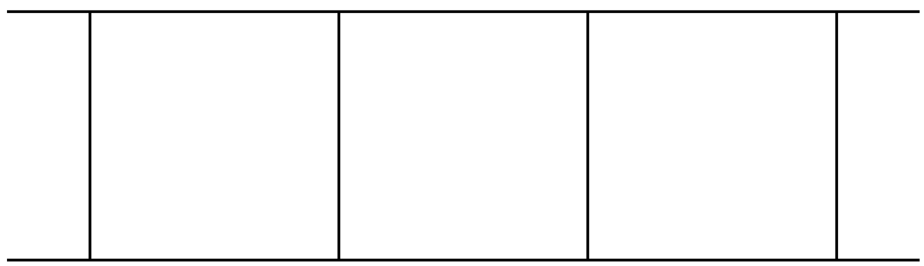
- Strategy A (Spyvakovsky, 1983)
- Strategy B and C by Bogner and Weinzierl (2008)
- Strategies by Binoth and Heinrich (X, ...), 2000-...
- Strategy S in FIESTA (2008)

Different strategies:

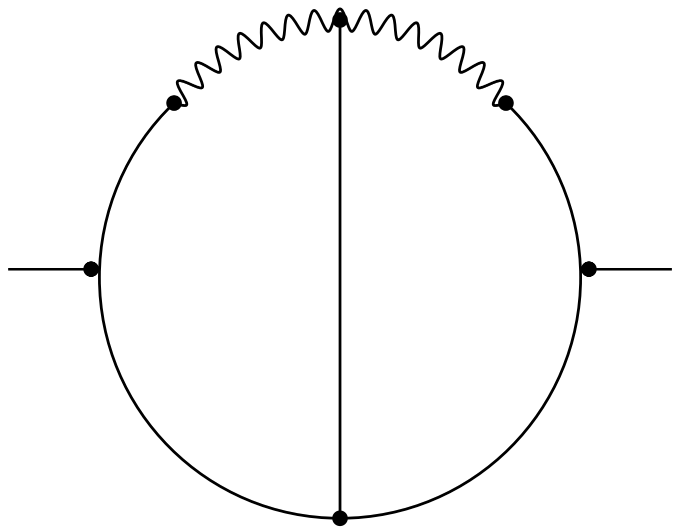
- Strategy A (Spyvakovsky, 1983)
- Strategy B and C by Bogner and Weinzierl (2008)
- Strategies by Binoth and Heinrich (X, ...), 2000-...
- Strategy S in FIESTA (2008)
- Hepp sectors and Speer sectors also can be represented as iterative strategies (A.Smirnov, V. Smirnov, 2009)

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:

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

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

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

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

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

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

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

One integration can be taken analytically!

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

One integration can be taken analytically!

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

Can be numerically integrated but still might result in problems!

Original FIESTA solved this problem with IfCuts.

For small values of variables one could integrate not the original expression, but its expansion by this variable in a Taylor series.

Non-efficient

- Announcement: in FIESTA 2 we are introducing multi-precision evaluations to handle numerical instability.

- Announcement: in FIESTA 2 we are introducing multi-precision evaluations to handle numerical instability.
- Reasoning: a sum of potentially huge terms is of normal size. Therefore all those terms have to be evaluated with high precision, otherwise after summing we don't obtain a proper result.

- Announcement: in FIESTA 2 we are introducing multi-precision evaluations to handle numerical instability.
- Reasoning: a sum of potentially huge terms is of normal size. Therefore all those terms have to be evaluated with high precision, otherwise after summing we don't obtain a proper result.
- Warning: high precision calculations during the integration don't mean you are going to get many digits as an answer. It is just a method to handle numerical instability.

Dealing with negative terms

What can we do if F has negative terms.

Three situations:

Dealing with negative terms

What can we do if F has negative terms.
Three situations:

- before the threshold - the function F is still always positive - in this case normal strategies do succeed

Dealing with negative terms

What can we do if F has negative terms.
Three situations:

- before the threshold - the function F is still always positive - in this case normal strategies do succeed
- above the threshold - the function F is sometimes negative - the result is a complex number - one might try to run FIESTA with UsingC=False and get say 2 highest poles. Nothing more can be done at the moment

Dealing with negative terms

What can we do if F has negative terms.
Three situations:

- before the threshold - the function F is still always positive - in this case normal strategies do succeed
- above the threshold - the function F is sometimes negative - the result is a complex number - one might try to run FIESTA with UsingC=False and get say 2 highest poles. Nothing more can be done at the moment
- at the threshold - the function F is non-negative, but might turn to zero, for example, if a pair of variables is equal to each other - recursive resolution with full squares (improved in FIESTA 2)

FIESTA internal structure

- Mathematica
- QLink
- CIntegrate on C
- Communication via Mathlink

As a result, we have:

As a result, we have:

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

As a result, we have:

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

As a result, we have:

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

As a result, we have:

- 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

As a result, we have:

- 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
- Cross-platform

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},{-kˆ2,-(k+p1)ˆ2,-(k+p1+p2)ˆ2,
        -(k+p1+p2+p4)ˆ2},{p1ˆ2->0,p2ˆ2->0,p4ˆ2->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 ep-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)/ep^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
                    ep
```

Summary of new features in FIESTA 2

Summary of new features in FIESTA 2

- Multiprecision evaluations to handle numerical instability

Summary of new features in FIESTA 2

- Multiprecision evaluations to handle numerical instability
- Improved parallelization (Usage of Mathematica 7 subkernels and integrating by multiple computers)

Summary of new features in FIESTA 2

- Multiprecision evaluations to handle numerical instability
- Improved parallelization (Usage of Mathematica 7 subkernels and integrating by multiple computers)
- New integrators (Thomas Hahn Cuba library)

Summary of new features in FIESTA 2

- Multiprecision evaluations to handle numerical instability
- Improved parallelization (Usage of Mathematica 7 subkernels and integrating by multiple computers)
- New integrators (Thomas Hahn Cuba library)
- Not only integral evaluation but also asymptotic behavior calculations

Summary of new features in FIESTA 2

- Multiprecision evaluations to handle numerical instability
- Improved parallelization (Usage of Mathematica 7 subkernels and integrating by multiple computers)
- New integrators (Thomas Hahn Cuba library)
- Not only integral evaluation but also asymptotic behavior calculations
- Speer sectors as a strategy

Summary of new features in FIESTA 2

- Multiprecision evaluations to handle numerical instability
- Improved parallelization (Usage of Mathematica 7 subkernels and integrating by multiple computers)
- New integrators (Thomas Hahn Cuba library)
- Not only integral evaluation but also asymptotic behavior calculations
- Speer sectors as a strategy
- Integrals at the threshold

Applications

Applications

- FIESTA 1 - used in many works for cross-checks

Applications

- FIESTA 1 - used in many works for cross-checks
- Systematic crosschecks in the calculation of the three-loop static quark potential (Smirnov, Smirnov, Steinhauser)

Applications

- FIESTA 1 - used in many works for cross-checks
- 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)

Applications

- FIESTA 1 - used in many works for cross-checks
- 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

Applications

- FIESTA 1 - used in many works for cross-checks
- 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
- Analytical results for the 28 master integrals for four-loop massless propagator integrals (Baikov, Chetyrkin) have been checked by FIESTA 2

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