

INTRODUCTION
TO
DIFFERENTIAL EQUATIONS
FOR
FEYNMAN INTEGRALS

Oleksandr Gituliar
<http://gituliar.net/capp17>



II. Institut für Theoretische Physik
Universität Hamburg

Computer Algebra in Particle Physics 2017
DESY (Hamburg)

Introduction

Feynman Integrals Calculus — became in recent decades a science on its own.

$$\int \underbrace{d^d l_1 \dots d^d l_n}_{\text{loops}} \underbrace{d^d p_1 \delta(p_1^2) \dots d^d p_m \delta(p_m^2)}_{\text{legs}} \frac{1}{D_1^{n_1} \dots D_k^{n_k}} \quad n_i \in \mathbb{Z}$$

Numerical methods

- Sector Decomposition, Subtraction Schemes, ...

Analytical methods

- Feynman/Schwinger/Mellin-Barnes parametrization
- Integration-By-Parts reduction Chetyrkin, Tkachov '81
 - Laporta algorithm Laporta '00: AIR, FIRE, Reduze
 - Symbolic reduction: LiteRed Lee '12
 - private implementations
- Method of Differential Equations Kotikov '91, Remiddi '97
 - Epsilon Form Henn '13
 - Lee algorithm Lee '14: Fuchsia, Epsilon
- ...

Introduction

Feynman Integrals Calculus — became in recent decades a science on its own.

$$\int \underbrace{d^d l_1 \dots d^d l_n}_{\text{loops}} \underbrace{d^d p_1 \delta(p_1^2) \dots d^d p_m \delta(p_m^2)}_{\text{legs}} \frac{1}{D_1^{n_1} \dots D_k^{n_k}} \quad n_i \in \mathbb{Z}$$

Integration-By-Parts reduction

- Integral Families
 - integration momenta
 - * loop – l_1, \dots, l_n only
 - * phase-space – p_1, \dots, p_m only
 - * mixed
 - set of denominators (topology)
 - master integrals
- Reduction
 - any integral (from the family) in terms of masters
 - * **including derivatives**
 - completely analytical
 - highly automated

Plan for Today

You will learn:

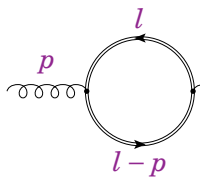
- **Integration-by-Parts** Reduction
 - LiteRed
- Differential Equations in **Epsilon Form**
 - Fuchsia
- **Examples**
 1. One-Loop Integral
 2. Two-Loop Phase-Space Integral
- **Partial Fractioning**
- Expansion of **Hypergeometric Functions**

Method of Differential Equations

1. Construct System of ODE (**medium**)
 - from definition (e.g. special functions)
 - from IBP rules
 - highly automated
 - AIR, FIRE, LiteRed, Reduze2
2. Find Epsilon Form (**hard**)
 - automated
 - Lee method: Fuchsia, epsilon
3. Solve System of ODE (**easy**)
4. Find Constants of Integration (**medium**)
 - depends on the problem

Example 1

One-Loop Massive Self-Energy


$$= \Pi_{ab}^{\mu\nu}(p^2, m) = \delta_{ab} (p^\mu p^\nu - g^{\mu\nu} p^2) \Pi(p^2, m)$$
$$\Pi(p^2, m) = \int d^n l F(p, l, m)$$

- **Arguments:** *from vectors to scalars*

$$F(p, l, m) \rightarrow F(l^2, l \cdot p, p^2, m)$$

- In general, the number of **scalar integration variables** is given by

$$N(L, E) = \frac{L(L+1)}{2} + LE \sim \mathcal{O}(L^2) \leftarrow \begin{array}{l} \text{another source of growing} \\ \text{complexity at higher orders} \end{array}$$

where E – number of *external momenta*, L – number of *loop momenta*

- 1-loop propagator: $N(1, 1) = 2$
- 4-loop propagator: $N(4, 1) = 14$ (ask Jos Vermaseren about details)

- The problem contains two denominators

$$D_1 = l^2 - m^2 \quad D_2 = (l - p)^2 - m^2$$

which map into our integration invariants in a unique way

$$F(p, l, m) \rightarrow F(l^2, l \cdot p, p^2, m) \rightarrow F(D_1, D_2, p^2, m)$$

- One integral family

$$F(n_1, n_2) = \int d^n l \frac{1}{D_1^{n_1} D_2^{n_2}}$$

```
<<LiteRed‘
```

```
SetDim[n];
```

```
Declare[{m2}, Number, {l,p}, Vector];
```

```
NewBasis[$b, {sp[l]-m2, sp[l-p]-m2}, {l}, Directory->"b.ibp"];
```

```
GenerateIBP[$b];
```

```
AnalyzeSectors[$b];
```

```
FindSymmetries[$b];
```

Example 1

Integration-by-Parts Reduction

In dimensional regularization the integral over a total derivative is zero.

$$\int d^n l_i \frac{d}{d l_i^\mu} (q^\mu F(p_1, \dots, l_1, \dots))$$

where q is arbitrary external or internal momenta.

IBP [5b]


```
SolvejSectors /@ UniqueSectors[$b]
```

```
MIs[$b]
```

```
> {j[$b,0,1], j[$b,1,1]}
```

- We obtain two master integrals

$$F_1 = F(0,1) = \int d^n l \frac{1}{(l-p)^2 - m^2} \quad F_2 = F(1,1) = \int d^n l \frac{1}{(l^2 - m^2)((l-p)^2 - m^2)}$$

- Any other integral is a linear combination of only these two, e.g.,

$$F(2,1) = \frac{n-2}{2m^2(p^2-4m^2)} F_1 + \frac{n-3}{p^2-4m^2} F_2$$

- We can check that since we can do $l \rightarrow l+p$ transformation

$$F(0,1) = F(1,0)$$

```
$ds = Dinv[#,sp[p,p]]& /@ MIs[$b] // IBPReduce;
$ode = Coefficient[#, MIs[$b]]& /@ $ds;
```

- This code produces a system of differential equations

$$\frac{dF_1}{dp^2} = 0$$

$$\frac{dF_2}{dp^2} = \frac{2-2\epsilon}{p^2(p^2-4m^2)}F_1 + \frac{2m^2-\epsilon p^2}{p^2(p^2-4m^2)}F_2$$

where we work in $n = 4 - 2\epsilon$ space-time dimensions

This system is simple and we could solve it right away using *<your favourite>* method. Today, I want to demonstrate you how this and many other systems can be solved through using their ϵ -form. As you will see this is a highly automated task.

Exercise

Derive another system of differential equations, but this time in m^2 . (Hint: use Fromj, D, and Toj functions instead of Dinv).

I. Epsilon Form

- Classical Notation

$$\frac{dF_1}{dx} = A_{11}(x, \epsilon)F_1 + A_{12}(x, \epsilon)F_2$$

$$\frac{dF_2}{dx} = A_{21}(x, \epsilon)F_1 + A_{22}(x, \epsilon)F_2$$

- Matrix Notation

$$\frac{d\bar{F}}{dx} = A(x, \epsilon)\bar{F} \quad \text{where} \quad A = \begin{pmatrix} A_{11}(x, \epsilon) & A_{12}(x, \epsilon) \\ A_{21}(x, \epsilon) & A_{22}(x, \epsilon) \end{pmatrix} \quad \text{and} \quad \bar{F} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$$

It is very convenient to have our system in the epsilon form

$$\frac{dG}{dx} = \epsilon B(x)G$$

since in this case we can easily find the solution to any order in ϵ parameter, as we will see on the next slide.

Some physical examples may lead to systems with ~ 500 equations. Hence, it is very important to make this task automatic.

II. A few words on Fuchsia

Input

- System of Ordinary Differential Equations $A(x, \epsilon, \dots)$, i.e.,

$$\frac{dF}{dx} = A(x, \epsilon, \dots) F(x, \epsilon, \dots)$$

Output

- Equivalent System in the Epsilon Form

$$\frac{dG}{dx} = \epsilon B(x, \dots) G(x, \epsilon, \dots)$$

- Corresponding Basis Transformation

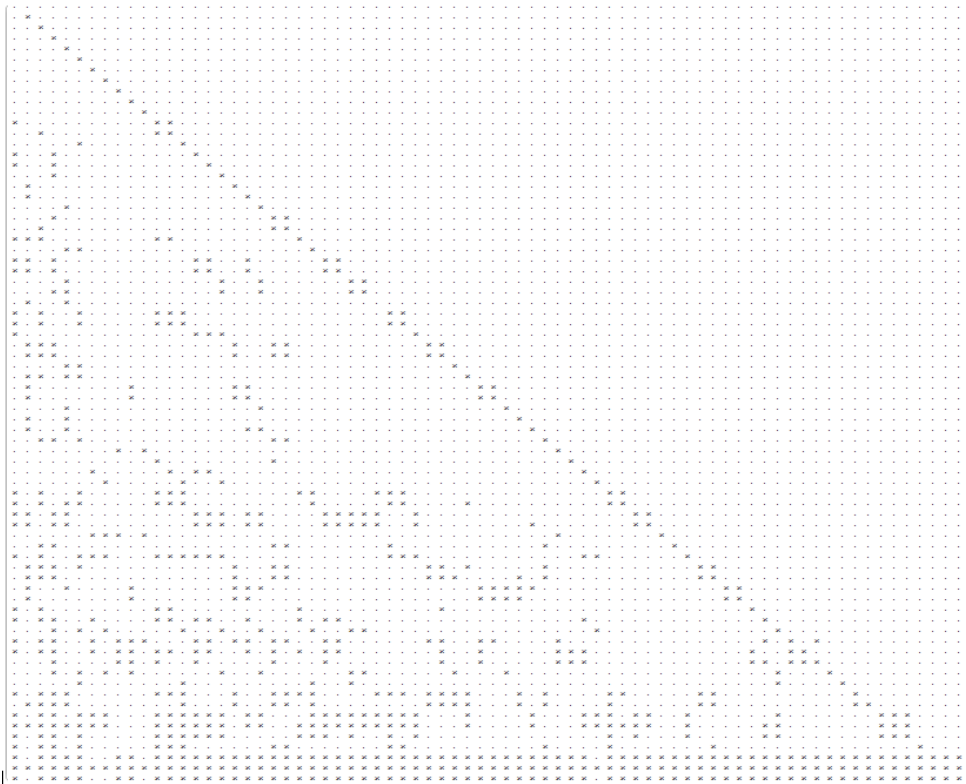
$$F(x, \epsilon, \dots) = T(x, \epsilon, \dots) \times G(x, \epsilon, \dots)$$

- Other Operations
 - apply custom transformation
 - variable change
 - "sort" to block-diagonal form

II. A few words on Fuchsia

- Based on the *Lee algorithm* Lee '14
 - support additional symbols
 - alternative implementation: epsilon
- Open-Source and Free Gituliar, Magerya '16 '17
 - <http://github.com/gituliar/fuchsia>
- Implemented in Python
 - SageMath
 - Maxima
 - Maple (optional)
- Algorithm
 1. **Fuchsification** (Jordan form)
Get rid of apparent singularities
 2. **Normalization** (eigenvalues, eigenvectors)
Balance eigenvalues to $\alpha \epsilon$ form
 3. **Factorization** (solve linear equations)
Reduce to the epsilon form

II. A few words on Fuchsia



Example 1

Epsilon Form by Fuchsia

Let us introduce a new variable y , such that

$$p^2 = -4m^2 \frac{y^2}{1-y^2}$$

The new equations look as

$$\begin{aligned}\frac{dF_1}{dy} &= 0 \\ \frac{dF_2}{dy} &= \frac{1-\epsilon}{y m^2} F_1 + \left(\frac{\epsilon}{1-y} - \frac{\epsilon}{1+y} - \frac{1}{y} \right) F_2\end{aligned}$$

With the help of Fuchsia we find a new basis G_1, G_2 given by the system

$$\begin{aligned}F_1 &= \frac{4(1-2\epsilon)}{3(1-\epsilon)} G_1 \\ F_2 &= \frac{4}{3m^2} G_1 - \frac{2}{y} G_2\end{aligned}$$

For this basis the differential equations are the epsilon form

$$\begin{aligned}\frac{dG_1}{dy} &= 0 \\ \frac{dG_2}{dy} &= \frac{2}{3m^2} \left(\frac{\epsilon}{1+y} + \frac{\epsilon}{1-y} \right) G_1 - \left(\frac{\epsilon}{1-y} - \frac{\epsilon}{1+y} \right) G_2\end{aligned}$$

III. Solutions

We are looking for the solution of a given system of ordinary differential equations in the epsilon form

$$\frac{dG}{dx} = \epsilon B(x) G$$

as a Laurent series in ϵ

$$G(x, \epsilon) = G_0(x) + G_1(x) \epsilon + G_2(x) \epsilon^2 + \dots$$

Let us put this "solution" into the initial equation

$$\frac{dG_0}{dx} + \frac{dG_1}{dx} \epsilon + \frac{dG_2}{dx} \epsilon^2 + \dots = \epsilon B(x) G_0 + \epsilon^2 B(x) G_1$$

we get

$$\frac{dG_0}{dx} = 0, \quad \frac{dG_1}{dx} = B(x)G_0, \quad \frac{dG_2}{dx} = B(x)G_1 \quad \dots \quad \frac{dG_n}{dx} = B(x)G_{n-1}$$

This system can be easily solved (as promised)

$$G_0 = C_0, \quad G_1 = C_1 + \int dx B(x) C_0, \quad G_2 = C_2 + \int dx B(x) \left(C_1 + \int dx B(x) C_0 \right) \quad \dots$$

$$G_n(x) = C_n + \int dx B(x) G_{n-1}$$

III. Solutions

My implementation of the solution algorithm, which I use to get results for the next slide.

```
SolveODE[m_, x_, ep_, n_, c_] := Module[
  {$i, $j, $n, $sol, $sol0, $sol1},

  $n = Length[m];
  $sol[0] = Table[c[$j,0], {$j,1,$n}];
  For[$i=1, $i<=n, $i++,
    $sol0 = Table[c[$j,$i], {$j,1,$n}];
    $sol1 = Integrate[Dot[#, $sol[$i-1]], x] & /@ m;
    $sol[$i] = $sol0 + $sol1;
  ];
  Sum[ep^$i*$sol[$i], {$i,0,n}]
];
```

- Master #1

$$F_1(y, m^2) = \frac{4}{3}C_1^{(0)} + \frac{4}{3}\left(C_1^{(1)} - C_1^{(0)}\right)\epsilon + \dots$$

- Master #2

$$F_2(y, m^2) = \frac{4C_1^{(0)}}{3m^2} - \frac{C_2^{(0)}}{y} + \frac{\epsilon}{3m^2y} \left(4yC_1^{(1)} - 6m^2C_2^{(1)} + (4C_1^{(1)} - 6m^2C_2^{(0)}) \ln\left(\frac{1-y}{1+y}\right) \right)$$

- Finally, we need to find unknown integration constants which are functions of m^2 and ϵ , i.e.

$$C_1^{(0)}(m^2, \epsilon), \quad C_1^{(1)}(m^2, \epsilon), \quad \dots$$

$$C_2^{(0)}(m^2, \epsilon), \quad C_2^{(1)}(m^2, \epsilon), \quad \dots$$

Example 1

Integration Constants #1

Master #1 (from Fuchsia)

$$F_1(y, m^2) = \frac{4}{3}C_1^{(0)} + \frac{4}{3}(C_1^{(1)} - C_1^{(0)})\epsilon + \dots$$

Closed-form solution from the literature (see Smirnov's book)

$$F(0, n) = (-1)^n \frac{\Gamma(n-2+\epsilon)}{\Gamma(n)} (m^2)^{2-\epsilon-n}$$

$$F_1(y, m^2) = F(0, 1) = \frac{m^2}{\epsilon} + m^2(1 - \gamma_E - \ln m^2) + \dots$$

Result #1

$C_1^{(0)} = \frac{3m^2}{4\epsilon} \quad C_1^{(1)} = \frac{3m^2(2 - \gamma_E - \ln m^2)}{4\epsilon}$

Example 1

Integration Constants #2

Result #1

$$C_1^{(0)} = \frac{m^2}{\epsilon} \quad C_1^{(1)} = \frac{m^2(1 - \gamma_E - \ln m^2)}{\epsilon}$$

Master #2 (with Result #1 substituted)

$$F_2(y, m^2) = \frac{1}{\epsilon} + \frac{2y - \gamma_E y - 2C_2^{(0)} - y \ln m^2 + \ln\left(\frac{1-y}{1+y}\right)}{y} + \dots$$

We require that at the limit $y \rightarrow 0$ ($p^2 \rightarrow 0$) our result is regular. This leads to the solution

$$C_2^0 = 0$$

Result #2

$$F_2(y, m^2) = \frac{1}{\epsilon} + 2 - \gamma_E - \ln m^2 + \frac{1}{y} \ln\left(\frac{1-y}{1+y}\right) + \dots$$

This is in agreement with T.Riemann Monday's lecture!

We have seen how to

- generate IBP rules for a given graph
- construct differential equations
- find epsilon form
- solve differential equations
- find integration constants

Exercies

- using LiteRed choose some two-loop (massless and massive) propagator and find corresponding masters
- solve Example #1, but using equations in m^2 (for help see Smirnov's book)

Example 2

Splitting Functions from e^+e^- -annihilation

In this example, I will show how to calculate a gluon-quark splitting function

$$P_{gq} = \frac{1+(1-x)^2}{x}$$

Using this technique you will be able to calculate remaining splitting functions P_{qq} , P_{qg} , and P_{gg} as well as higher-order corrections to these quantities.



$$e^+(q_1) + e^-(q_2) \rightarrow q(p_1) + \bar{q}(p_2) + g(p_3)$$

Mass-factorization theorem

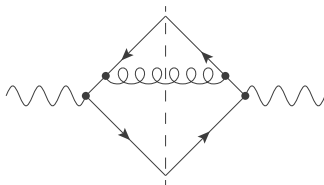
$$\frac{d\sigma_i}{dx} = \frac{P_{iq}}{\epsilon} + a_i + b_i\epsilon + \dots$$

where $q = q_1 + q_2$ and

$$\frac{d\sigma_i}{dx} = \int d^n p_1 \delta(p_1^2) d^n p_2 \delta(p_2^2) d^n p_3 \delta(p_3^2) \delta\left(x - \frac{2q \cdot p_i}{q^2}\right) \sigma(q_1, q_2, p_1, p_2, p_3)$$

Example 2

By their structure phase-space integrals are very similar to loop integrals (compare to the one-loop propagator from Example I), except that we apply on-shell conditions $\delta(p_i^2)$ to the cut lines as shown in the following **cut graph**



$$\frac{d\sigma_g}{dx} = \int d^n p_1 \delta(p_1^2) d^n p_2 \delta(p_2^2) d^n p_3 \delta(p_3^2) \delta\left(x - \frac{2q \cdot p_3}{q^2}\right) \sigma(q_1, q_2, p_1, p_2, p_3)$$

where

$$\sigma(q_1, q_2, p_1, p_2, p_3) = N \frac{(p_1 \cdot q_1)^2 + (p_2 \cdot q_1)^2 + (p_1 \cdot q_2)^2 + (p_2 \cdot q_2)^2}{p_1 \cdot p_3 p_2 \cdot p_3}$$

This integration is equivalent to the 2-loop propagator, since we can eliminate one of the integration momenta using momentum conservation

$$q_1 + q_2 = p_1 + p_2 + p_3$$

Example 2

In order to integrate the cross-section we need a new IBP basis. Let us define one as

```
NewBasis[ $\$a$ , {sp[p1], sp[p3], sp[q1+q2-p1-p3], s*x-2sp[q1+q2,p3], sp[p1,p3]},  
         {p1, p3}, Append -> True];  
  
GenerateIBP[ $\$a$ ];  
AnalyzeSectors[ $\$a$ , {___,0,0}, CutDs -> {1,1,1,1,0,0,0}];  
FindSymmetries[];  
  
SolvejRules /@ UniqueSectors[ $\$a$ ];
```

Note additional arguments in AnalyzeSectors routine:

- in {___,0,0} 0's represent invariants which appear in numerators only
- in CutDs -> {1,1,1,1,0,0,0} 1's represent "cut" propagators. It means that all integrals with at least one non-positive indices in these places vanish.

We get only one master integral

$$F_1(x, \epsilon) = \int d^n p_1 \delta(p_1^2) d^n p_2 \delta(p_2^2) d^n p_3 \delta(p_3^2) \delta\left(x - \frac{2q \cdot p_3}{q^2}\right)$$

IV. Partial Fractioning

Given a set of denominators, being a *linear combination* of the kinematic invariants s_{ij} , make a partial fraction such that

$$\frac{1}{D_1 \dots D_n} \rightarrow \frac{a_1}{D_2 \dots D_n} + \frac{a_2}{D_1 D_3 \dots D_n} + \dots + \frac{a_n}{D_1 \dots D_{n-1}}$$

All we need is to solve a linear system of equations

$$a_1 D_1 + \dots + a_n D_n = N$$

where the coefficient in front of every s_{ij} is zero and N is some number.

In particaulr, for

$$A = \frac{1}{(x+1)(y+1)(x+y+1)}$$

we write down

$$(a_1 + a_3)x + (a_2 + a_3)y + a_1 + a_2 + a_3 = N$$

the solution is

$$a_1 = -a_3 \quad a_2 = -a_3 \quad N = -a_3$$

which gives

$$A = \frac{1}{(y+1)(x+y+1)} + \frac{1}{(x+1)(x+y+1)} - \frac{1}{(x+1)(y+1)}$$

Example 2

Now we can convert the initial cross-section into the j -form and make IBP reduction.

```
M2 = (sp[p1,q1]^2+sp[p1,q2]^2+sp[p2,q1]^2+sp[p2,q2]^2)/(x*sp[q1,q2]*sp[p1,p3]
PS2 = x / (sp[p1]*sp[p3]*sp[q1+q2-p1-p3]*(s*x-2*sp[q1+q2,p3]));
jM2 = Toj[$a, PS2*M2];
jM2 = jM2 // IBPReduce
Pgg = Series[jM2 /. {m -> 4-2*eps}, {eps, 0, -1}]
```

This gives us

$$P_{gq} \sim \frac{2 - 2x + x^2}{x^2} F_1(x)$$

which contains one x factor more in the denominator than we expect.

Maybe $F_1(x) \sim x$? Let us check...

```
F1 = j[$a, 1, 1, 1, 1, 0, 0, 0];  
dF1 = Toj[$a, D[Fromj[$F1], x]] // IBPReduce;
```

This code produces the following equation

$$\frac{dF_1}{dx} = \left(\frac{\epsilon}{1-x} + \frac{1-2\epsilon}{x} \right) F_1$$

Of course we could use Fuchsia and find the ϵ -form, but we can solve this in a closed form

$$F_1 = C(\epsilon) (1-x)^\epsilon x^{1-2\epsilon}$$

which confirms our assumption from the previous slide.

The final result is

$$P_{gq} \sim \frac{2-2x+x^2}{x}$$

Now you also now how to calculate phase-space integrals.

Exercise

Redefine x as

$$x = \frac{2q \cdot p_1}{q^2}$$

and find a well-known result

$$P_{qq} = \frac{1+x^2}{1-x}$$

for the quark-quark splitting function.

V. Holonomic Functions

A function $f = f(x)$ is called *holonomic* if there exist polynomials $a_n(x), \dots, a_0(x)$ such that

$$a_n(x)f^{(n)} - a_{n-1}(x)f^{(n-1)} - \dots - a_0(x)f = 0$$

holds for all x . Hence, the holonomic function is uniquely defined by

- the differential equation
- a number of initial values $f(x_0), f'(x_0), \dots, f^{(n-1)}(x_0)$

Examples of holonomic functions:

- *all algebraic functions*
- *Generalized Hypergeometric functions*
 - *polylogarithms*
 - *Elliptic functions*
- *Bessel functions*
- *Airy functions*
- *Legendre and Chebyshev polynomials*
- *Heun functions*
- and many others that have no name and no closed form

V. Holonomic Functions

A function $f = f(x)$ is called *holonomic* if there exist polynomials $a_n(x), \dots, a_0(x)$ such that

$$a_n(x)f^{(n)} - a_{n-1}(x)f^{(n-1)} - \dots - a_0(x)f = 0$$

holds for all x . Hence, the holonomic function is uniquely defined by

- the differential equation
- a number of initial values $f(x_0), f'(x_0), \dots, f^{(n-1)}(x_0)$

Conclusion

- simple representation
 - polynomials
 - ordinary differential equations
- define many complicated functions
 - no closed form
 - non-trivial integration representation
- represent Feynman integrals
- alternative for direct integration

V. Holonomic Functions

We can easily rewrite a n^{th} -order linear ODE given by

$$y^{(n)} - a_1(x)y^{(n-1)} - \dots - a_n(x)y = 0 \quad (1)$$

as an $n \times n$ system of the form

$$\frac{d\bar{y}}{dx} = A(x)\bar{y}$$

where

$$A(x) = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ a_n(x) & a_{n-1}(x) & \cdots & a_2(x) & a_1(x) \end{bmatrix} \quad \text{and} \quad \bar{y} = \begin{pmatrix} y \\ y' \\ \vdots \\ y^{(n-2)} \\ y^{(n-1)} \end{pmatrix}$$

However, the inverse operation is not as easy anymore.

VI. Hypergeometric Functions

The Generalized Hypergeometric Function

$${}_{p+1}F_p \left(\begin{matrix} a_1, a_2, \dots, a_{p+1} \\ b_1, b_2, \dots, b_p \end{matrix}; x \right) = \prod_{i=1}^p \frac{\Gamma(b_i)}{\Gamma(a_i)\Gamma(b_i - a_i)} \int_0^1 \frac{t_i^{a_i-1} (1-t_i)^{b_i-a_i-1}}{(1-x t_1 \dots t_p)^{a_{p+1}}} dt_i$$

is a solution to the differential equation

$$[D(D+b_1-1) \cdots (D+b_p-1) - x(D+a_1) \cdots (D+a_{p+1})] y = 0$$

where

$$D = x \frac{d}{dx}$$

Exercise

Using your favourite CAS write a routine which for a given Generalized Hypergeometric Function, defined by the list $\{a_1, \dots, a_{p+1}, b_1, \dots, b_p\}$, returns a corresponding ODE, defined by the list $\{a_1(x), \dots, a_p(x)\}$, in accordance with notation of eq. (1).

Reading List

- **Feynman Integral Calculus** by V. Smirnov
- **Lectures on Differential Equations for Feynman Integrals** by J. Henn
- **Formal Power Series and Linear Systems of Meromorphic Ordinary Differential Equations** by W. Balser
- **Computer Algebra in Particle Physics** by S. Weinzierl
- **Introduction to Loop Calculations** by G. Heinrich
- **Structure and Interpretation of Computer Programs** by H. Abelson and G. Sussman with J. Sussman