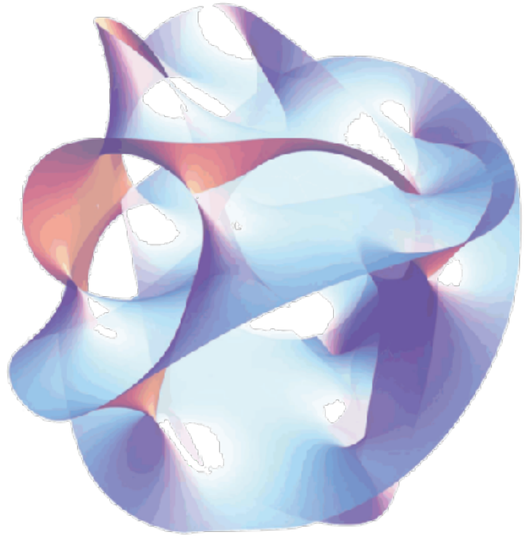
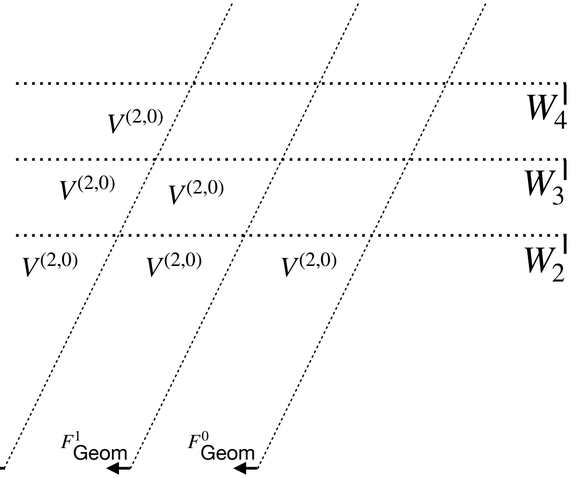
Geometric bookkeeping for ε-factorised DEQs

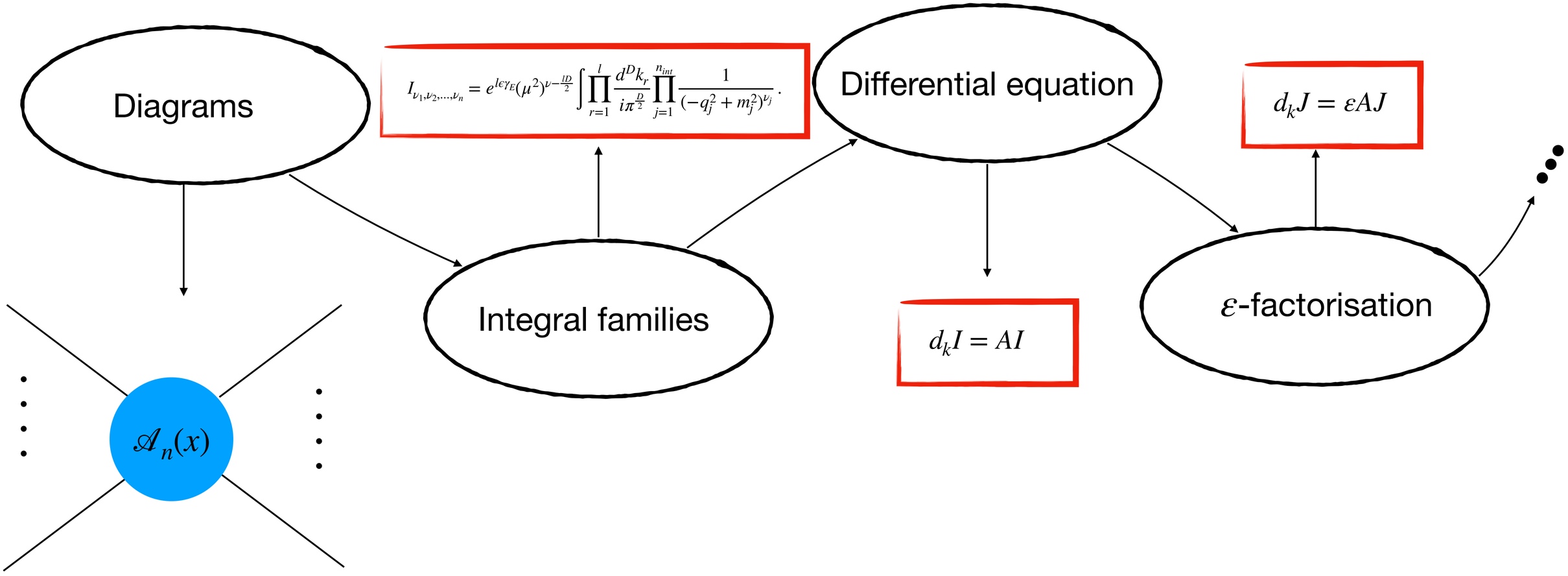


Based on worked done in The ε-Collaboration:
Iris Bree, Federico Gasparotto, Antonela Matijašić, **Pouria Mazloumi**, Dmytro Melnichenko, Sebastian Pögel, Toni Teschke, Xing Wang, Stefan Weinzierl, Konglong Wu and Xiaofeng Xu
Arxiv: 2506.09124

Elliptics and Beyond 25'
3.09.2025, Bonn



Machinery of Feynman Integrals



A lot of Success !

- Analytic method to compute Feynman integrals to arbitrary order in ε.
- Different methods so far are relying on external information about the geometries.

Our Algorithm

- A systematic approach to find ε-factorised differential equations.
- Not reliant on the information (periods, etc) of underlying geometry.

A lot of Success !

[Talks by Christoph Nega, Sven Stawinski, Sara Maggio]

- Analytic method to compute Feynman integrals to arbitrary order in ε.
- Different methods so far are relying on external information about the geometries.

Our Algorithm

- A systematic approach to find ε-factorised differential equations.
- Not reliant on the information (periods, etc) of underlying geometry.

Twisted cohomology and Filtration

Twisted cohomology and Filtration

$$I = \int_{\gamma_n} u \, \underbrace{\Phi_n}$$

$$\nabla_\omega : \Omega^n(X, \omega) \to \Omega^{(n+1)}(X, \omega)$$

$$\omega = d\log(u),$$
$$\nabla_\omega = d + \omega \wedge \quad .$$

$$u = \prod_{j=1}^{m} P_j^{\alpha_j}(z) \longrightarrow \text{Multivalued function}$$

Twisted cohomology and Filtration

$$I = \int_{\gamma_n} u \, \underbrace{\Phi_n}$$

$$\nabla_\omega : \Omega^n(X, \omega) \to \Omega^{(n+1)}(X, \omega)$$

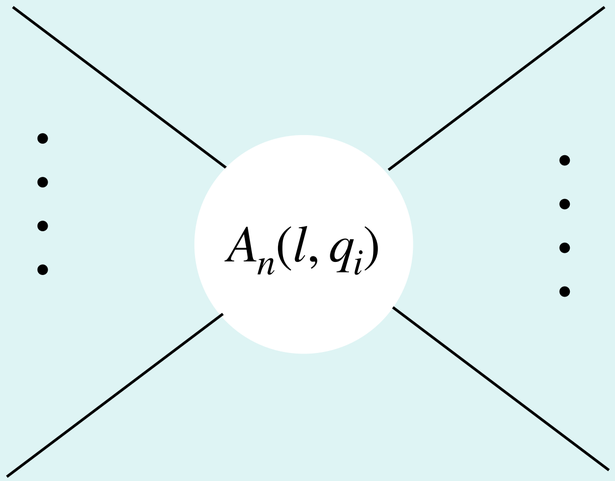
$$\omega = d\log(u),$$
$$\nabla_\omega = d + \omega \wedge \quad .$$

$$u = \prod_{j=1}^{m} P_j^{\alpha_j}(z) \longrightarrow \text{Multivalued function}$$

IBPs:
$$\int_{\gamma_n} u \, \Phi_n = \int_{\gamma_n} u \, [\Phi_n + \nabla_\omega \xi_{n-1}] \, .$$

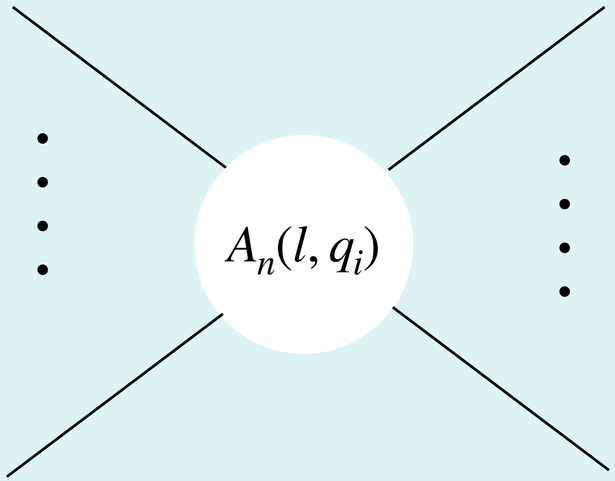
Twisted cohomology and Filtration

Relation to Feynman integrals:



Twisted cohomology and Filtration

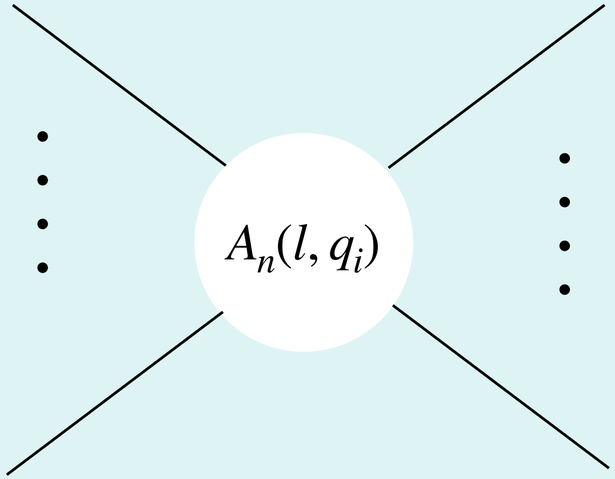
Relation to Feynman integrals:



$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}} .$$

Twisted cohomology and Filtration

Relation to Feynman integrals:



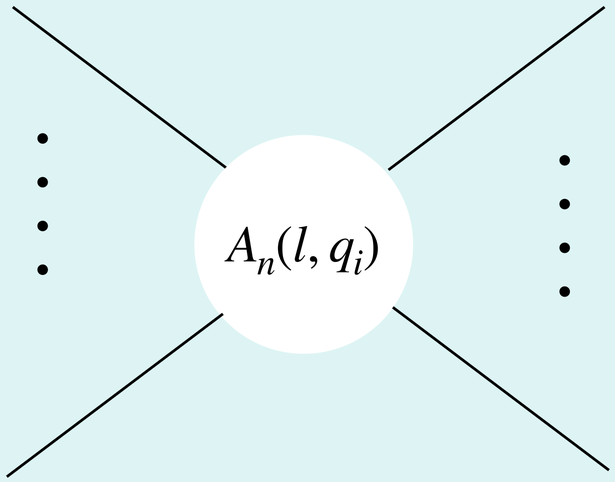
$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}} .$$

$$z_j = -q_j^2 + m_j^2$$

Loop by loop
Baikov

Twisted cohomology and Filtration

Relation to Feynman integrals:



$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}}.$$

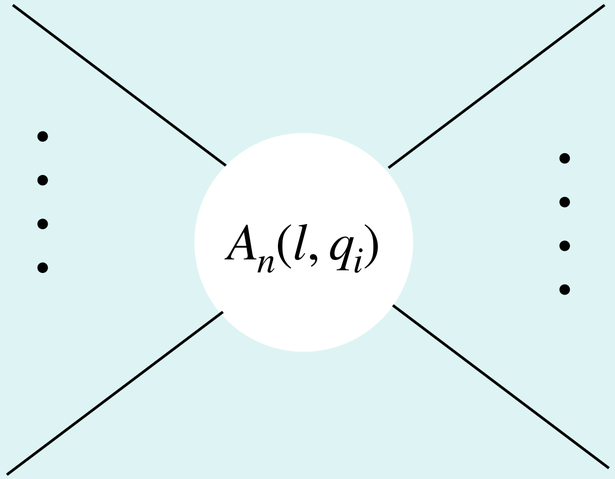
$$z_j = -q_j^2 + m_j^2$$

Loop by loop
Baikov

$$I_{\nu_1,\nu_2,\ldots,\nu_{N_v}} = C_{baikov} \int_{\gamma_B} \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z_i, x) \prod_{i=1}^{N_v} \frac{dz_i}{z_i^{\nu_i}}$$

Twisted cohomology and Filtration

Relation to Feynman integrals:



$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}}.$$

$$z_j = -q_j^2 + m_j^2$$

Loop by loop
Baikov

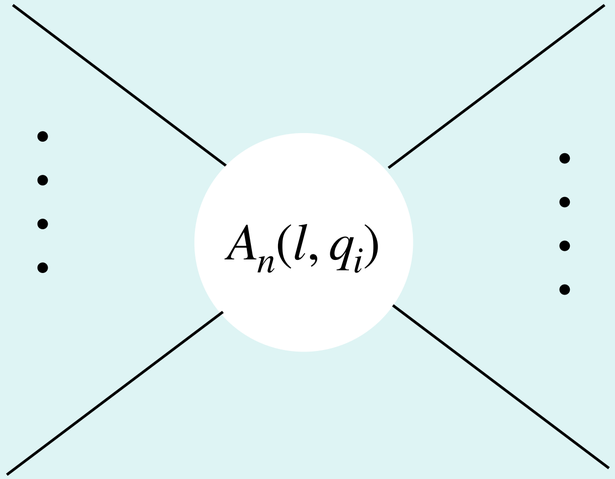
Polynomial

MV- Function $u = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x)$

$$I_{\nu_1,\nu_2,\ldots,\nu_{N_v}} = C_{baikov} \int_{\gamma_B} \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z_i, x) \prod_{i=1}^{N_v} \frac{dz_i}{z_i^{\nu_i}}$$

Twisted cohomology and Filtration

Relation to Feynman integrals:



$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}}.$$

$$z_j = -q_j^2 + m_j^2$$

Loop by loop
Baikov

Polynomial

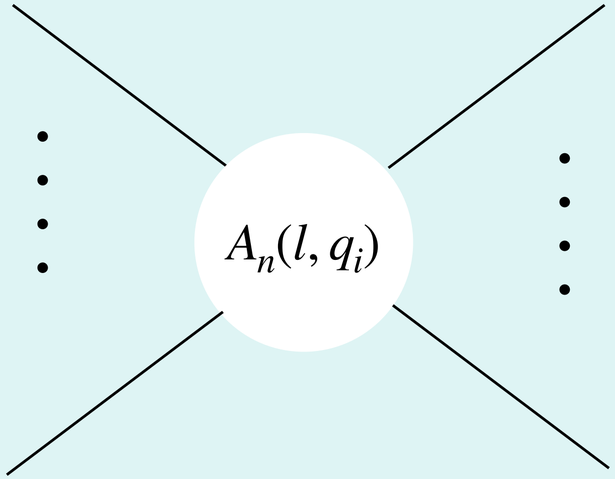
MV- Function $$u = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x)$$

$$I_{\nu_1,\nu_2,\ldots,\nu_{N_v}} = C_{baikov} \int_{\gamma_B} \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z_i, x) \prod_{i=1}^{N_v} \frac{dz_i}{z_i^{\nu_i}}$$

Diff. Form $$\Phi_n = \frac{dz_1 \wedge dz_2 \ldots . \wedge dz_{N_v}}{\prod_i z_i^{\nu_i}}$$

Twisted cohomology and Filtration

Relation to Feynman integrals:



$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}}.$$

$$z_j = -q_j^2 + m_j^2$$

Loop by loop
Baikov

Polynomial

MV- Function $$u = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x)$$

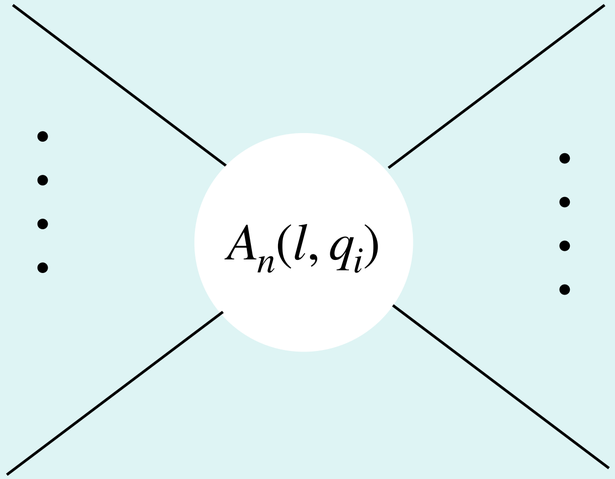
$$I_{\nu_1,\nu_2,\ldots,\nu_{N_v}} = C_{baikov} \int_{\gamma_B} \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z_i, x) \prod_{i=1}^{N_v} \frac{dz_i}{z_i^{\nu_i}}$$

Diff. Form $$\Phi_n = \frac{dz_1 \wedge dz_2 \ldots . \wedge dz_{N_v}}{\prod_i z_i^{\nu_i}}$$

Go on cuts

Twisted cohomology and Filtration

Relation to Feynman integrals:



$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}}.$$

$$z_j = -q_j^2 + m_j^2$$

Loop by loop
Baikov

$$I_{\nu_1,\nu_2,\ldots,\nu_{N_\nu}} = C_{baikov} \int_{\gamma_B} \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z_i, x) \prod_{i=1}^{N_\nu} \frac{dz_i}{z_i^{\nu_i}},$$

Polynomial

MV- Function $u = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x)$

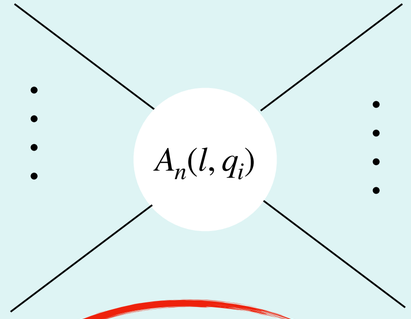
Diff. Form $\Phi_n = \frac{dz_1 \wedge dz_2 \ldots . \wedge dz_{N_\nu}}{\prod_i z_i^{\nu_i}}$

Go on cuts

$$I = \int_{\gamma_n} u \ \Phi_{N_\nu}.$$

Twisted cohomology and Filtration

Relation to Feynman integrals:



$$j : V^n \to H^n_\omega$$

$$I_{\nu_1,\nu_2,\ldots,\nu_n} = e^{l\epsilon\gamma_E}(\mu^2)^{\nu-\frac{lD}{2}} \int \prod_{r=1}^{l} \frac{d^D q_r}{i\pi^{\frac{D}{2}}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}}.$$

$$z_j = -q_j^2 + m_j^2$$

Loop by loop
Baikov

Polynomial

MV- Function $u = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x)$

$$I_{\nu_1,\nu_2,\ldots,\nu_{N_v}} = C_{baikov} \int_{\gamma_B} \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z_i, x) \prod_{i=1}^{N_v} \frac{dz_i}{z_i^{\nu_i}},$$

Diff. Form $\Phi_n = \frac{dz_1 \wedge dz_2 \ldots . \wedge dz_{N_v}}{\prod_i z_i^{\nu_i}}$

Go on cuts

$$I = \int_{\gamma_n} u \ \Phi_{N_v}.$$

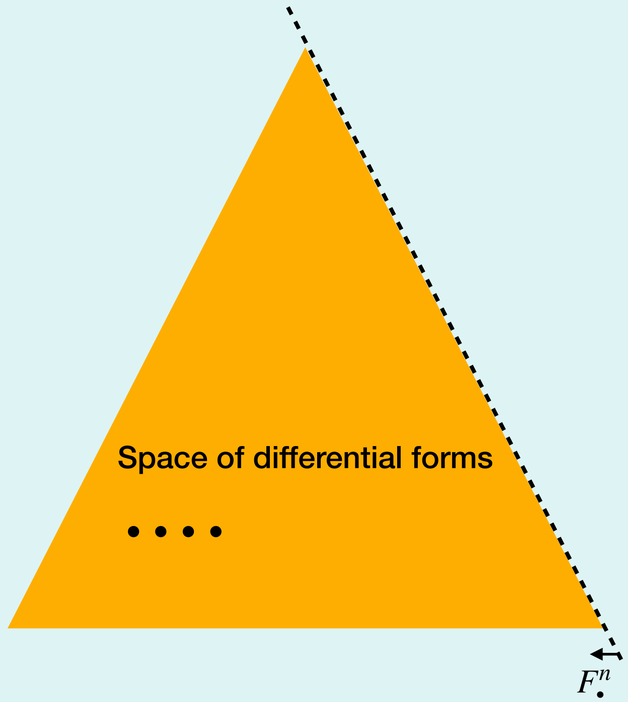
Twisted cohomology and Filtration

We have set of differential forms of the twisted cohomology: $\Omega^p(X, \nabla_\omega)$

Filtration $\mathscr{F}$ of set S is an indexed family $F^i_\bullet$ of sub-objects such that :

$$\text{if } i \leq j \Rightarrow F^i_\bullet \subseteq F^j_\bullet,$$

such that $S = \bigcup_{i \in \mathbb{N}} F^i_\bullet$.



- Increasing or decreasing
- Infinite or finite

We want to categorise differential forms by some index i.

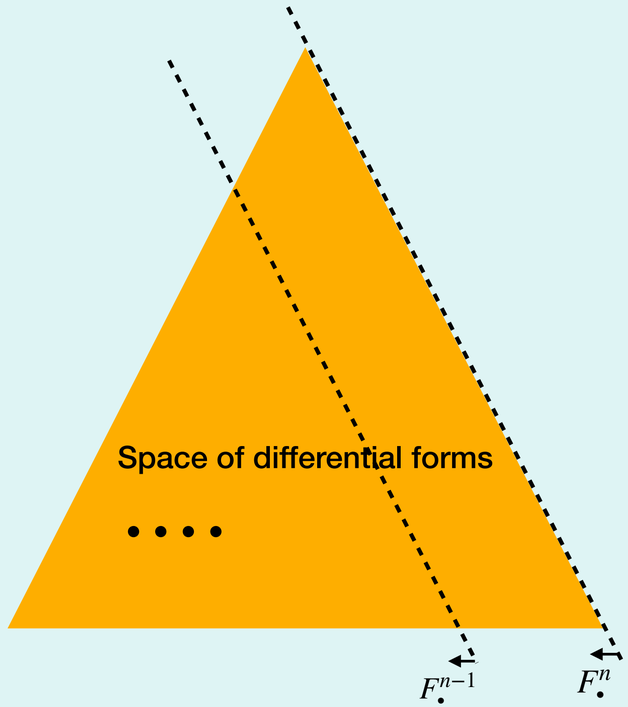
Twisted cohomology and Filtration

We have set of differential forms of the twisted cohomology: $\Omega^p(X, \nabla_\omega)$

Filtration $\mathscr{F}$ of set S is an indexed family $F^i_\bullet$ of sub-objects such that :

$$\text{if } i \leq j \Rightarrow F^i_\bullet \subseteq F^j_\bullet,$$

such that $S = \bigcup_{i \in \mathbb{N}} F^i_\bullet$.



- Increasing or decreasing
- Infinite or finite

We want to categorise differential forms by some index i.

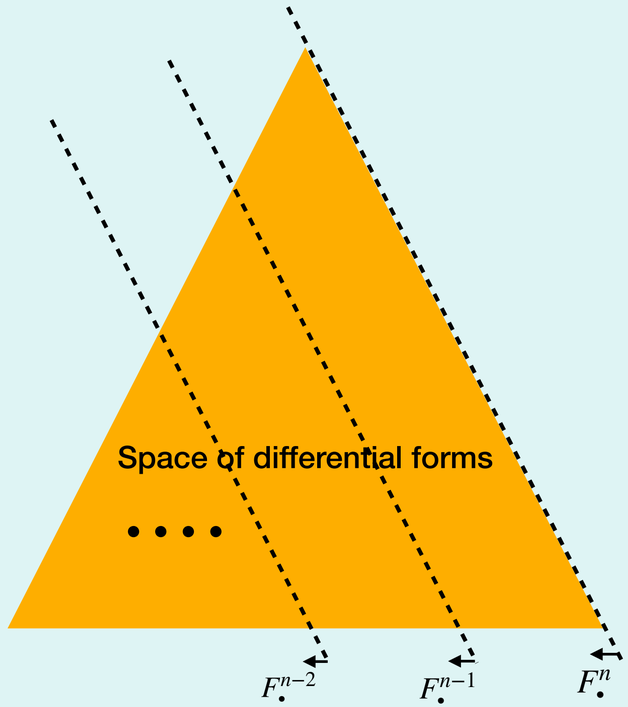
Twisted cohomology and Filtration

We have set of differential forms of the twisted cohomology: $\Omega^p(X, \nabla_\omega)$

Filtration $\mathscr{F}$ of set S is an indexed family $F^i_\bullet$ of sub-objects such that :

$$\text{if } i \leq j \Rightarrow F^i_\bullet \subseteq F^j_\bullet,$$

such that $S = \bigcup_{i \in \mathbb{N}} F^i_\bullet$.



- Increasing or decreasing
- Infinite or finite

We want to categorise differential forms by some index i.

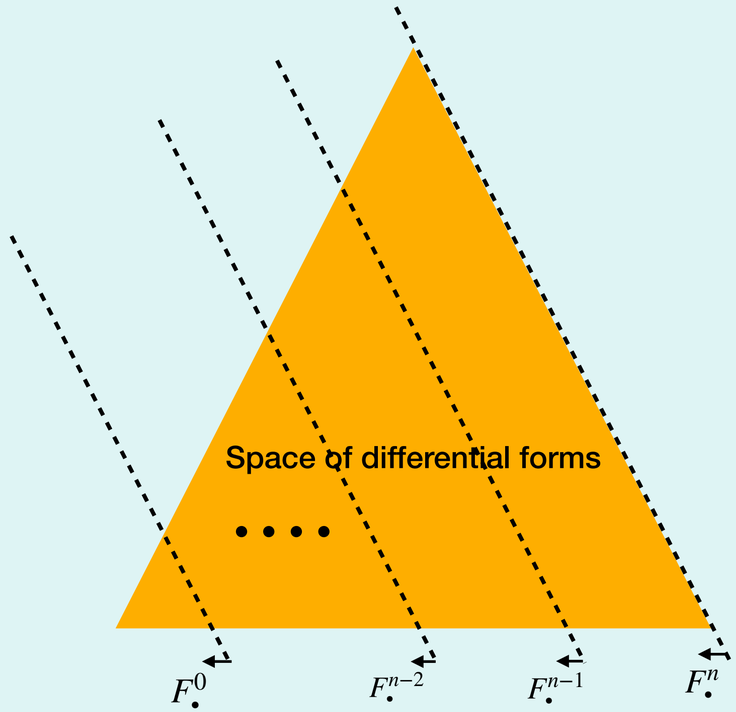
Twisted cohomology and Filtration

We have set of differential forms of the twisted cohomology: $\Omega^p(X, \nabla_\omega)$

Filtration $\mathscr{F}$ of set S is an indexed family $F^i_\bullet$ of sub-objects such that :

$$\text{if } i \leq j \Rightarrow F^i_\bullet \subseteq F^j_\bullet,$$

such that $S = \bigcup_{i \in \mathbb{N}} F^i_\bullet$.



- Increasing or decreasing
- Infinite or finite

We want to categorise differential forms by some index i.

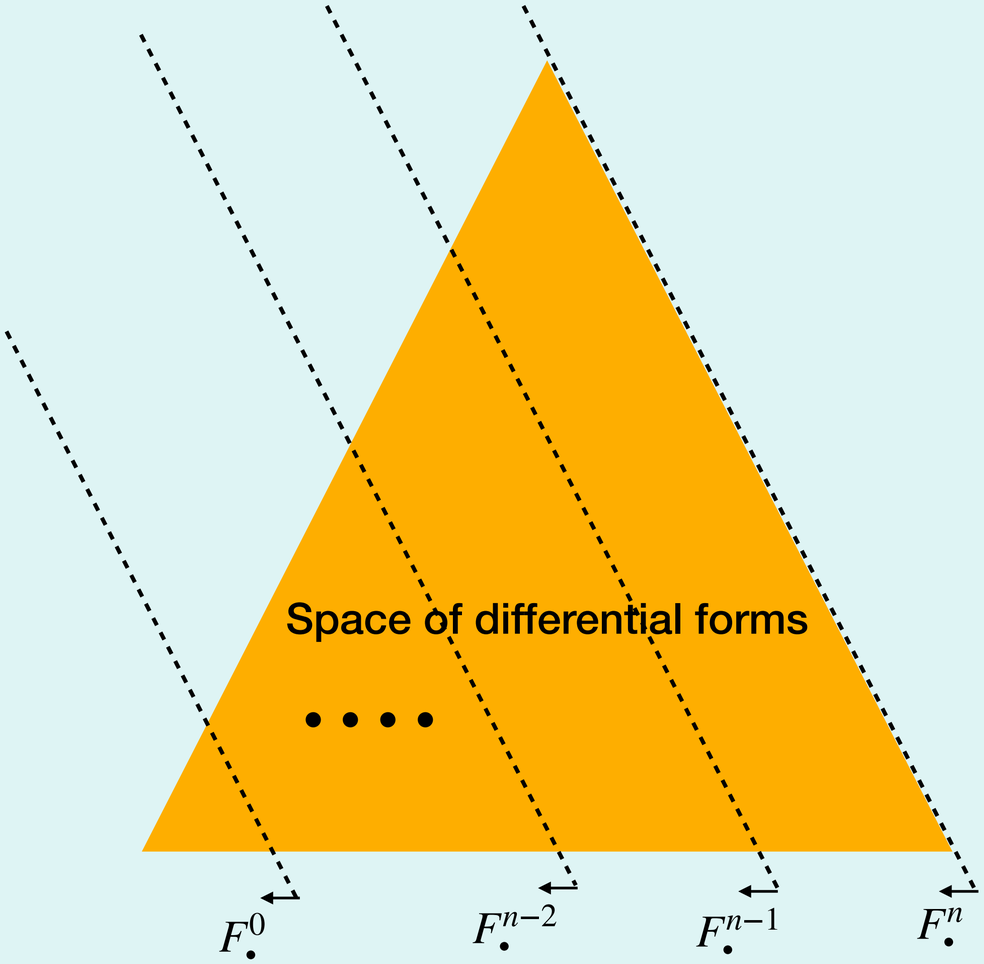
Twisted cohomology and Filtration

Grading

Given the filtration $\mathscr{F}$ we can now introduced graded algebra associated to the filtration it is given by:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n$$



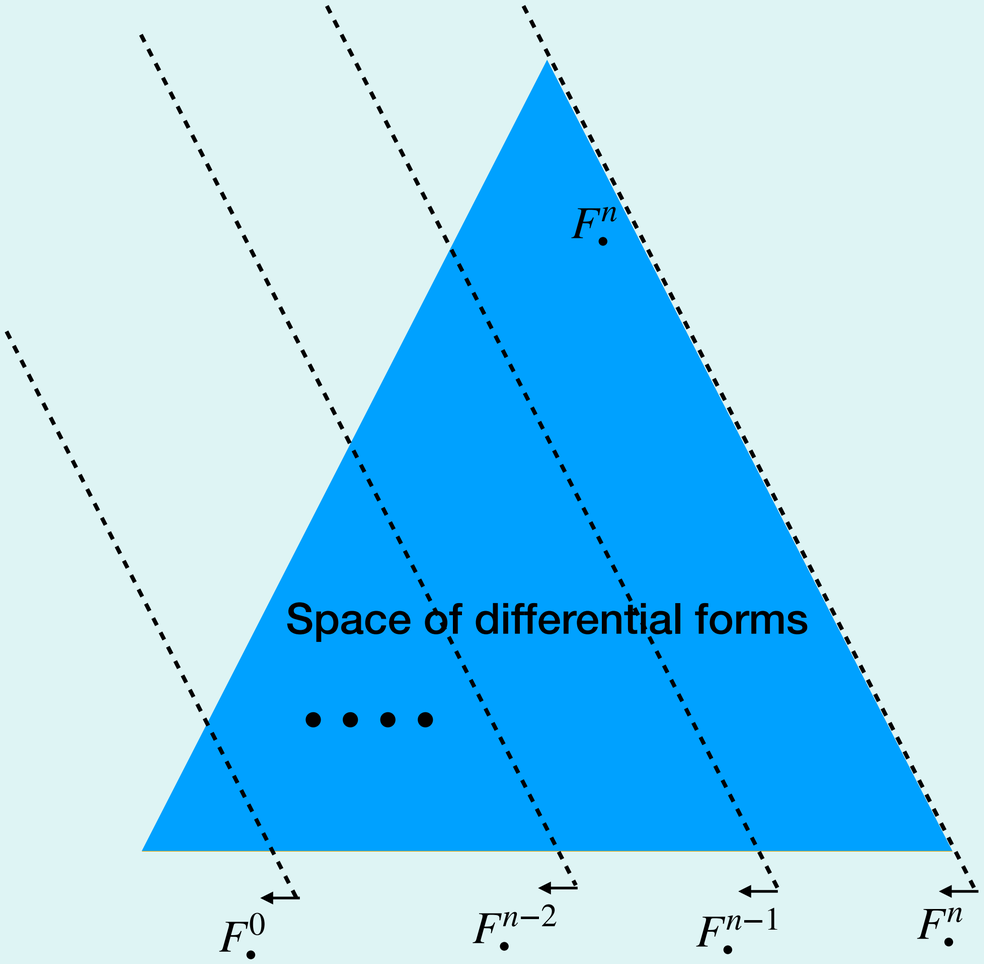
Twisted cohomology and Filtration

Grading

Given the filtration $\mathscr{F}$ we can now introduced graded algebra associated to the filtration it is given by:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n$$



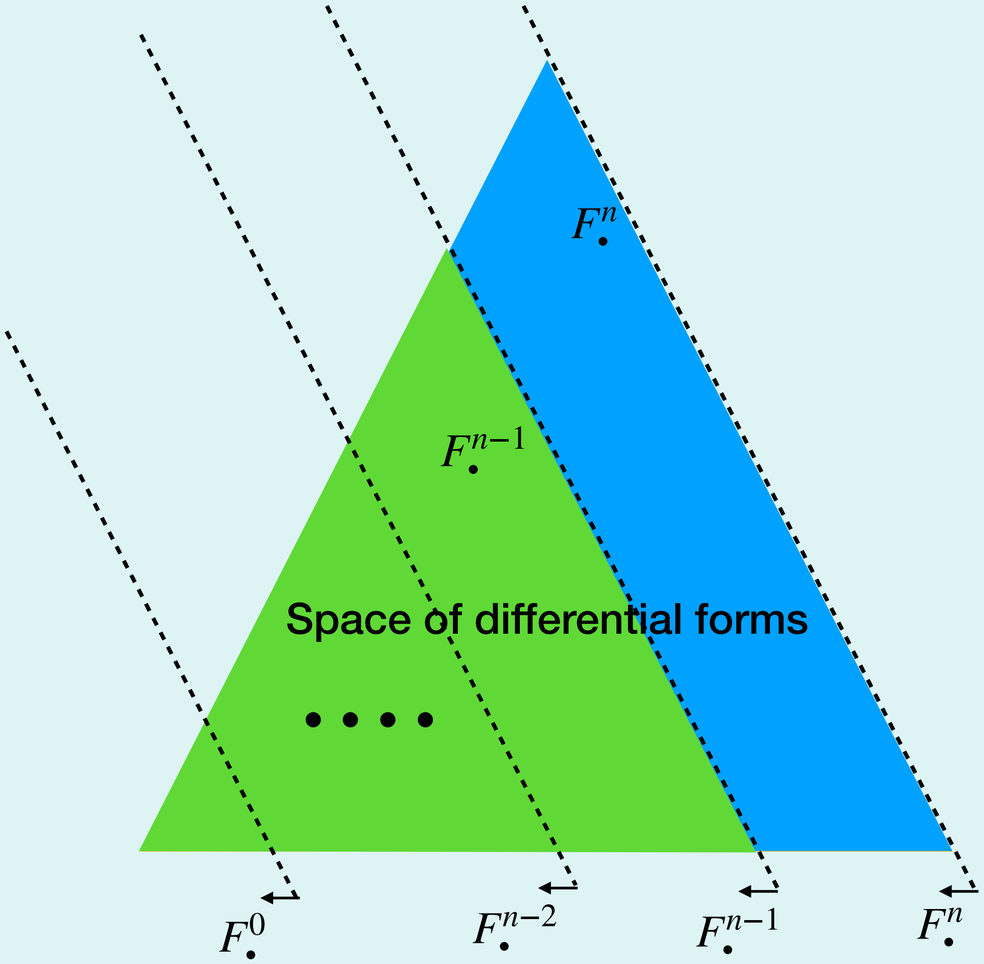
Twisted cohomology and Filtration

Grading

Given the filtration $\mathscr{F}$ we can now introduced graded algebra associated to the filtration it is given by:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n$$



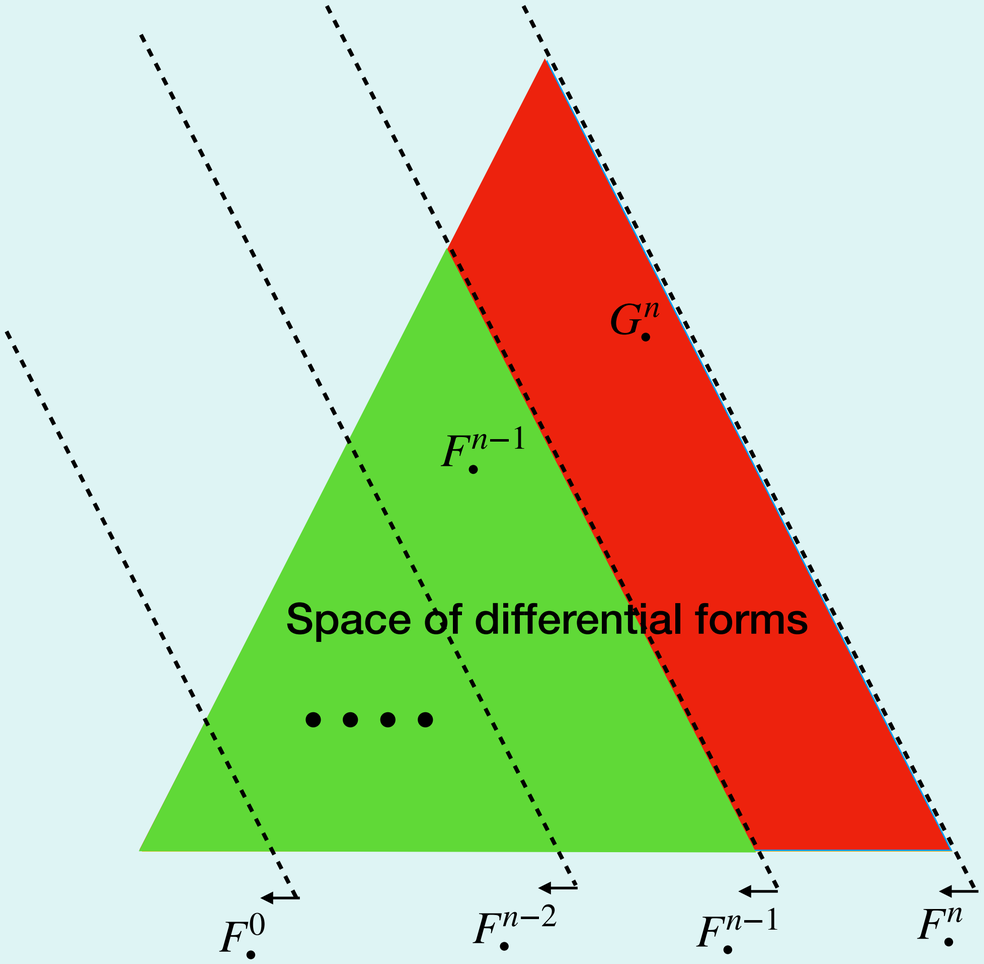
Twisted cohomology and Filtration

Grading

Given the filtration $\mathscr{F}$ we can now introduced graded algebra associated to the filtration it is given by:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n$$



The Algorithm

The Algorithm

The Algorithm

1. Constructing an intermediate basis J where DEQs are Laurent polynomials.

The Algorithm

1. Constructing an intermediate basis J where DEQs are Laurent polynomials.

$$dJ = \sum_{i=-k_{min}}^{1} \varepsilon^i A^{(i)}(x) \cdot J \,.$$

The Algorithm

1. Constructing an intermediate basis J where DEQs are Laurent polynomials.

$$dJ = \sum_{i=-k_{min}}^{1} \varepsilon^i A^{(i)}(x) \cdot J \,.$$

2. Construct a rotation matrix R such that basis $K = R^{-1} \cdot J$ has ε-factorised DEQs.

Definitions:

The minimal Twist

Max cut

$N_v \to n$

Definitions:

The minimal Twist

$(l, s) \in \mathbb{Z}^2$

$$u(z_i, x) = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x) \to \begin{cases} \alpha_j(\varepsilon) & = l\frac{1}{2} + \frac{1}{2} b_j \varepsilon, \qquad j \in I_{odd}, \\ \alpha_j(\varepsilon) & = s + \frac{1}{2} b_j \varepsilon, \qquad j \in I_{even}\,. \end{cases}$$

Max cut

$N_v \to n$

Definitions:

The minimal Twist

$$(l, s) \in \mathbb{Z}^2$$

$$u(z_i, x) = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x) \to \begin{cases} \alpha_j(\varepsilon) & = l\frac{1}{2} + \frac{1}{2} b_j \varepsilon, \qquad j \in I_{odd}, \\ \alpha_j(\varepsilon) & = s + \frac{1}{2} b_j \varepsilon, \qquad j \in I_{even}\,. \end{cases}$$

Max cut

$$N_v \to n$$

$$U = \prod_{j=0}^{N} P_j^{\alpha_j(\varepsilon)}(z_0, z, x)$$

Homogenise

$$z_i \to \frac{z_i}{z_0}$$

$$P_0(z_0, z_i, x) = z_0$$

Definitions:

The minimal Twist

$$(l, s) \in \mathbb{Z}^2$$

$$u(z_i, x) = \prod_{j=1}^{N} P_j^{\alpha_j(\varepsilon)}(z, x) \to \begin{cases} \alpha_j(\varepsilon) & = l\frac{1}{2} + \frac{1}{2} b_j \varepsilon, \qquad j \in I_{odd}, \\ \alpha_j(\varepsilon) & = s + \frac{1}{2} b_j \varepsilon, \qquad j \in I_{even}\,. \end{cases}$$

Max cut

$$N_v \to n$$

$$U = \prod_{j=0}^{N} P_j^{\alpha_j(\varepsilon)}(z_0, z, x)$$

$$l \to 1$$
$$s \to 0$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2} + \frac{1}{2} b_j \varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2} b_j \varepsilon}(z_0, z, x)$$

Minimal Twist

Homogenise

$$z_i \to \frac{z_i}{z_0}$$

$$P_0(z_0, z_i, x) = z_0$$

Definitions:

The Form

$$\Psi_{\mu_1,\ldots,\mu_n}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_n}[Q] \, \eta \, .$$

Definitions:

The Form

$$\Psi_{\mu_1,\ldots,\mu_n}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_n}[Q] \, \eta \, .$$

$$\hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] = \frac{Q}{\prod\limits_{i \in I^0_{all}} P_i^{\mu_i}} \qquad \mu = \{\mu_1, \mu_2, \ldots, \mu_n\}$$

Definitions:

The Form

$$\Psi_{\mu_1,\ldots,\mu_n}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_n}[Q] \, \eta \, .$$

$$\hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] = \frac{Q}{\prod\limits_{i \in I^0_{all}} P_i^{\mu_i}} \qquad \mu = \{\mu_1, \mu_2, \ldots, \mu_n\}$$

$$C_\varepsilon := C_{abs} \times C_{rel} \times C_{clutch} = \begin{cases} C_{abs} & \text{such that } C_{abs} \times C_{Baikov} \text{ has uniform transcendental weight,} \\ C_{clutch} & := \varepsilon^{-|\mu|}, \\ C_{rel} & := \prod\limits_{i \in I^0_{odd}} \left(-\frac{1}{2} + \frac{1}{2} b_i \varepsilon\right)_{\mu_i} \prod\limits_{i \in I^0_{even}} \left(\frac{1}{2} b_i \varepsilon\right)_{\mu_i} . \end{cases}$$

$$(a)_n := \frac{\Gamma(a+1)}{\Gamma(a+1-n)}$$

Definitions:

The Form

$$\Psi_{\mu_1,\ldots,\mu_n}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_n}[Q] \, \eta \, .$$

$$\hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] = \frac{Q}{\prod\limits_{i \in I^0_{all}} P_i^{\mu_i}} \qquad \mu = \{\mu_1, \mu_2, \ldots, \mu_n\}$$

$$C_\varepsilon := C_{abs} \times C_{rel} \times C_{clutch} = \begin{cases} C_{abs} & \text{such that } C_{abs} \times C_{Baikov} \text{ has uniform transcendental weight,} \\ C_{clutch} & := \varepsilon^{-|\mu|}, \\ C_{rel} & := \prod\limits_{i \in I^0_{odd}} \left(-\frac{1}{2} + \frac{1}{2} b_i \varepsilon\right)_{\mu_i} \prod\limits_{i \in I^0_{even}} \left(\frac{1}{2} b_i \varepsilon\right)_{\mu_i} . \end{cases}$$

$$(a)_n := \frac{\Gamma(a+1)}{\Gamma(a+1-n)}$$

$$\eta = \sum_{j=0}^{n} (-1)^j z_j dz_0 \wedge \ldots \wedge \hat{dz}_j \wedge \ldots \wedge dz_n$$

Definitions:

The Form

$$\Psi_{\mu_1,\ldots,\mu_n}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_n}[Q] \, \eta \, .$$

$$\hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] = \frac{Q}{\prod\limits_{i \in I^0_{all}} P_i^{\mu_i}} \qquad \mu = \{\mu_1, \mu_2, \ldots, \mu_n\}$$

$$C_\varepsilon := C_{abs} \times C_{rel} \times C_{clutch} = \begin{cases} C_{abs} & \text{such that } C_{abs} \times C_{Baikov} \text{ has uniform transcendental weight,} \\ C_{clutch} & := \varepsilon^{-|\mu|}, \\ C_{rel} & := \prod\limits_{i \in I^0_{odd}} \left(-\frac{1}{2} + \frac{1}{2} b_i \varepsilon\right)_{\mu_i} \prod\limits_{i \in I^0_{even}} \left(\frac{1}{2} b_i \varepsilon\right)_{\mu_i} . \end{cases}$$

$$(a)_n := \frac{\Gamma(a+1)}{\Gamma(a+1-n)}$$

$$\eta = \sum_{j=0}^{n} (-1)^j z_j dz_0 \wedge \ldots \wedge \hat{dz}_j \wedge \ldots \wedge dz_n$$

Given a twist $U(z)$ one can generate a system of IBPs which has a particularly simple form.

Definitions:

The Form

$$\Psi_{\mu_1,\ldots,\mu_n}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_n}[Q] \, \eta \, .$$

Given a twist $U(z)$ one can generate a system of IBPs which has a particularly simple form.

Definitions:

The Form

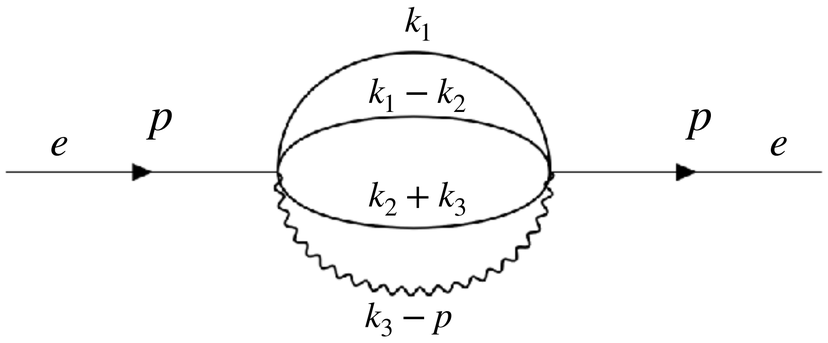
$$\Psi_{\mu_1,\ldots,\mu_n}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_n}[Q] \, \eta \, .$$

$$\frac{1}{\varepsilon}\Psi_{\mu_1,\ldots,\mu_j,\ldots,\mu_n}[dQ] + \sum_j \Psi_{\mu_1,\ldots,\mu_j+1,\ldots,\mu_n}[QdP_j] = 0.$$

Given a twist $U(z)$ one can generate a system of IBPs which has a particularly simple form.

Electron—Self energy

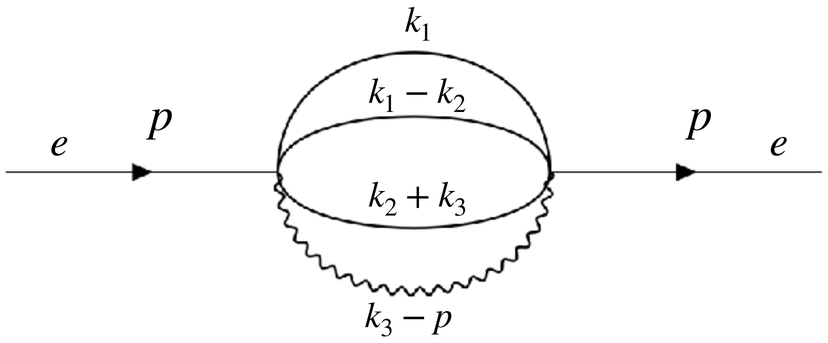
sector: 15



[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl ’24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) \, .$$

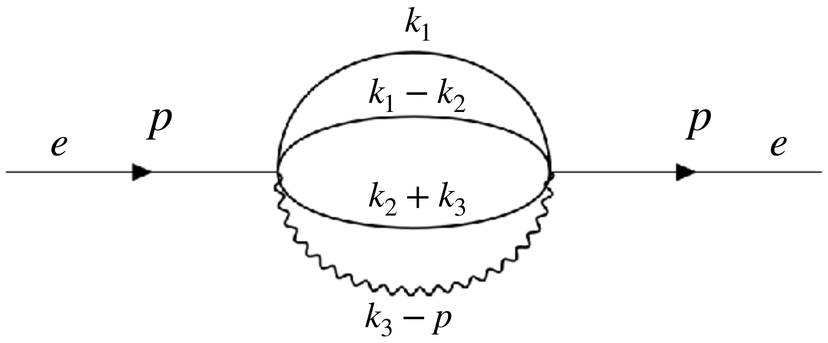
(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x$.

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl ’24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) \, .$$

(z_0, z_1, z_2)

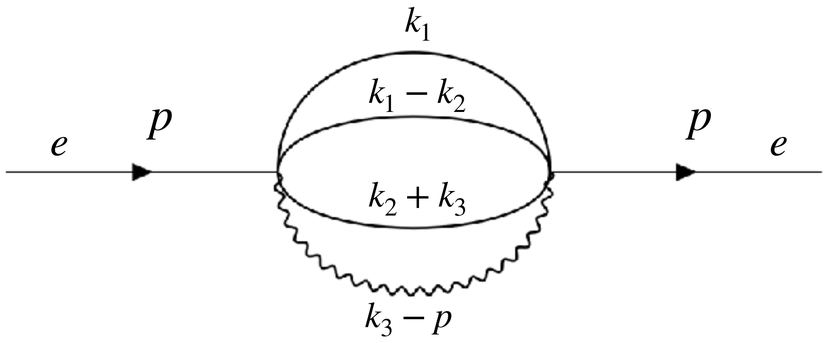
$p^2 \to 1,$
$m_e \to x \, .$

$$U_m(z_0, z, x) = P_0^{4\varepsilon} P_1^{\varepsilon} P_2^{-2\varepsilon} P_3^{-\frac{1}{2}} P_4^{-\frac{1}{2}-\varepsilon} P_5^{-\frac{1}{2}-\varepsilon} \, .$$

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) \, .$$

(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x$.

$$U_m(z_0, z, x) = P_0^{4\varepsilon} P_1^{\varepsilon} P_2^{-2\varepsilon} P_3^{-\frac{1}{2}} P_4^{-\frac{1}{2}-\varepsilon} P_5^{-\frac{1}{2}-\varepsilon} \, .$$

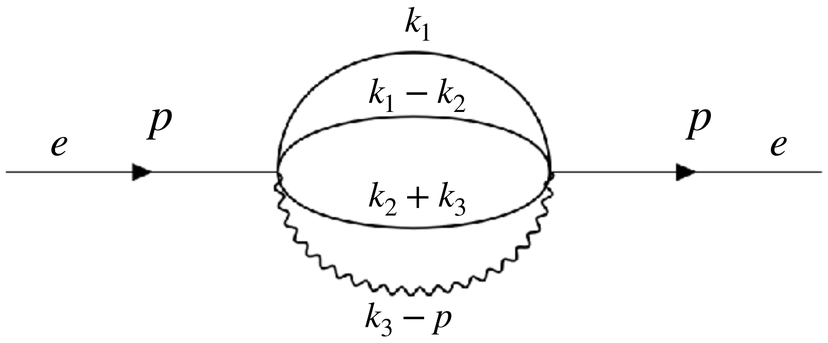
$$C_{baikov} = \frac{2^{6+6\varepsilon} \pi^{\frac{9}{2}} e^{3\gamma_E \varepsilon}}{\Gamma(\frac{1}{2} - \varepsilon)^3},$$

$$C_{abs} = \varepsilon^3 \, .$$

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) \, .$$

(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x$.

Even

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2) \, . \end{cases}$$

$$U_m(z_0, z, x) = P_0^{4\varepsilon} P_1^{\varepsilon} P_2^{-2\varepsilon} P_3^{-\frac{1}{2}} P_4^{-\frac{1}{2}-\varepsilon} P_5^{-\frac{1}{2}-\varepsilon} \, .$$

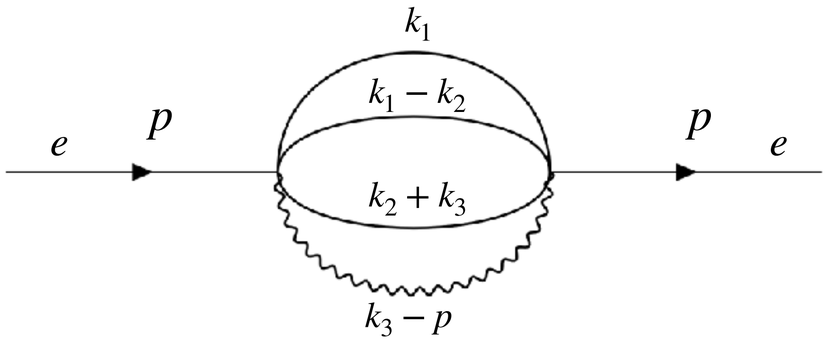
$$C_{baikov} = \frac{2^{6+6\varepsilon} \pi^{\frac{9}{2}} e^{3\gamma_E \varepsilon}}{\Gamma(\frac{1}{2} - \varepsilon)^3},$$

$$C_{abs} = \varepsilon^3 \, .$$

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) .$$

(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x .$

Even

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2) . \end{cases}$$

$$U_m(z_0, z, x) = P_0^{4\varepsilon} P_1^{\varepsilon} P_2^{-2\varepsilon} P_3^{-\frac{1}{2}} P_4^{-\frac{1}{2}-\varepsilon} P_5^{-\frac{1}{2}-\varepsilon} .$$

Odd

$$\begin{cases} P_3 = z_1, \\ P_4 = (4xz_0 + z_1), \\ P_5 = z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2 . \end{cases}$$

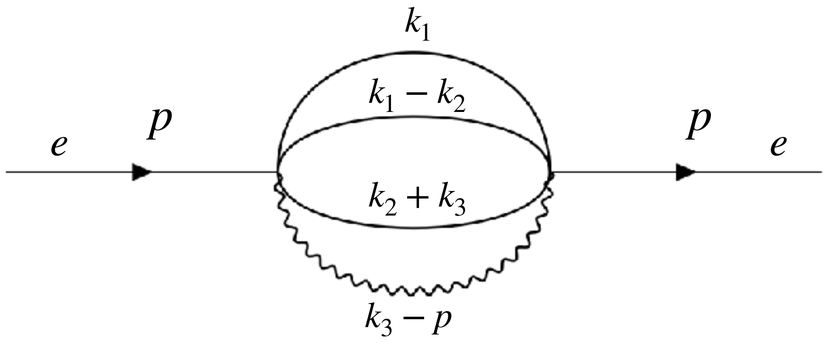
$$C_{baikov} = \frac{2^{6+6\varepsilon} \pi^{\frac{9}{2}} e^{3\gamma_E \varepsilon}}{\Gamma(\frac{1}{2} - \varepsilon)^3},$$

$$C_{abs} = \varepsilon^3 .$$

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) \, .$$

(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x$.

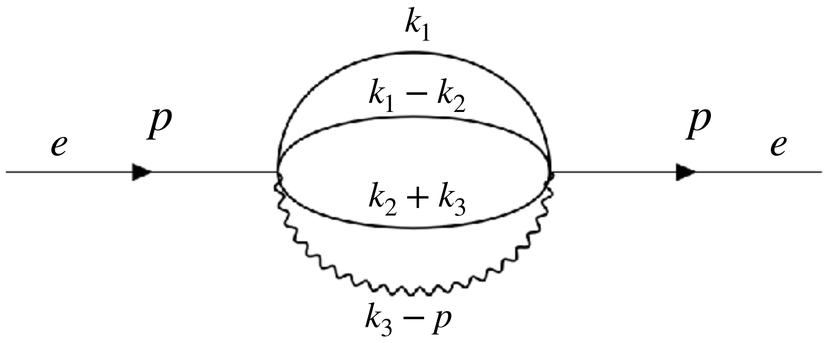
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2) \, . \end{cases}$$

$$\begin{cases} P_3 &= z_1, \\ P_4 &= (4xz_0 + z_1), \\ P_5 &= z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2 \, . \end{cases}$$

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) \, .$$

(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x$.

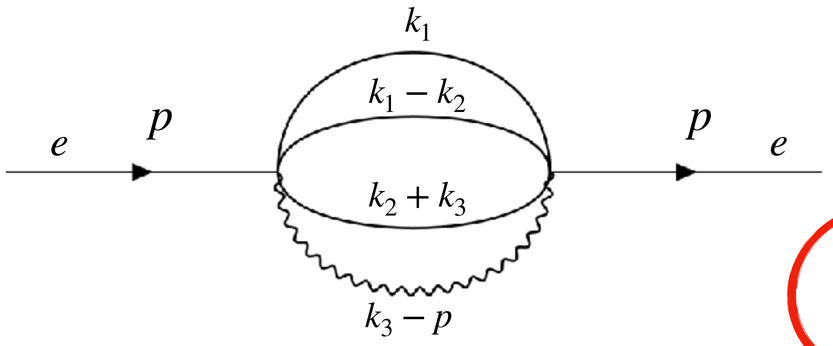
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2) \, . \end{cases}$$

$$\begin{cases} P_3 &= z_1, \\ P_4 &= (4xz_0 + z_1), \\ P_5 &= z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2 \, . \end{cases}$$

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x) .$$

(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x$.

$z_0 = 1,$

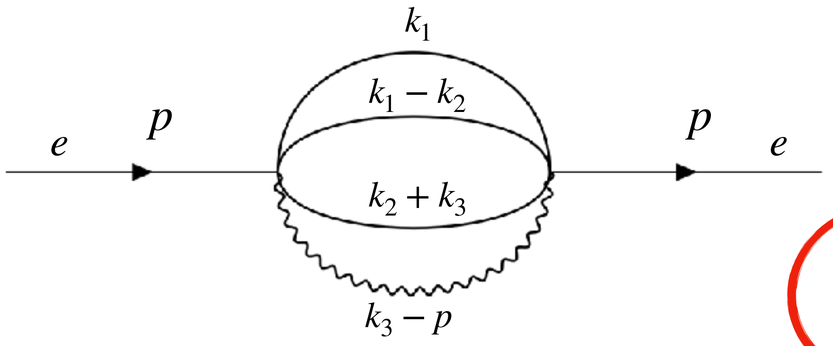
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2) . \end{cases}$$

$$\begin{cases} P_3 &= z_1, \\ P_4 &= (4xz_0 + z_1), \\ P_5 &= z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2 . \end{cases}$$

[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Electron—Self energy

sector: 15



Max cut
$d = 2 - 2\varepsilon$.

$z_0 = 1,$

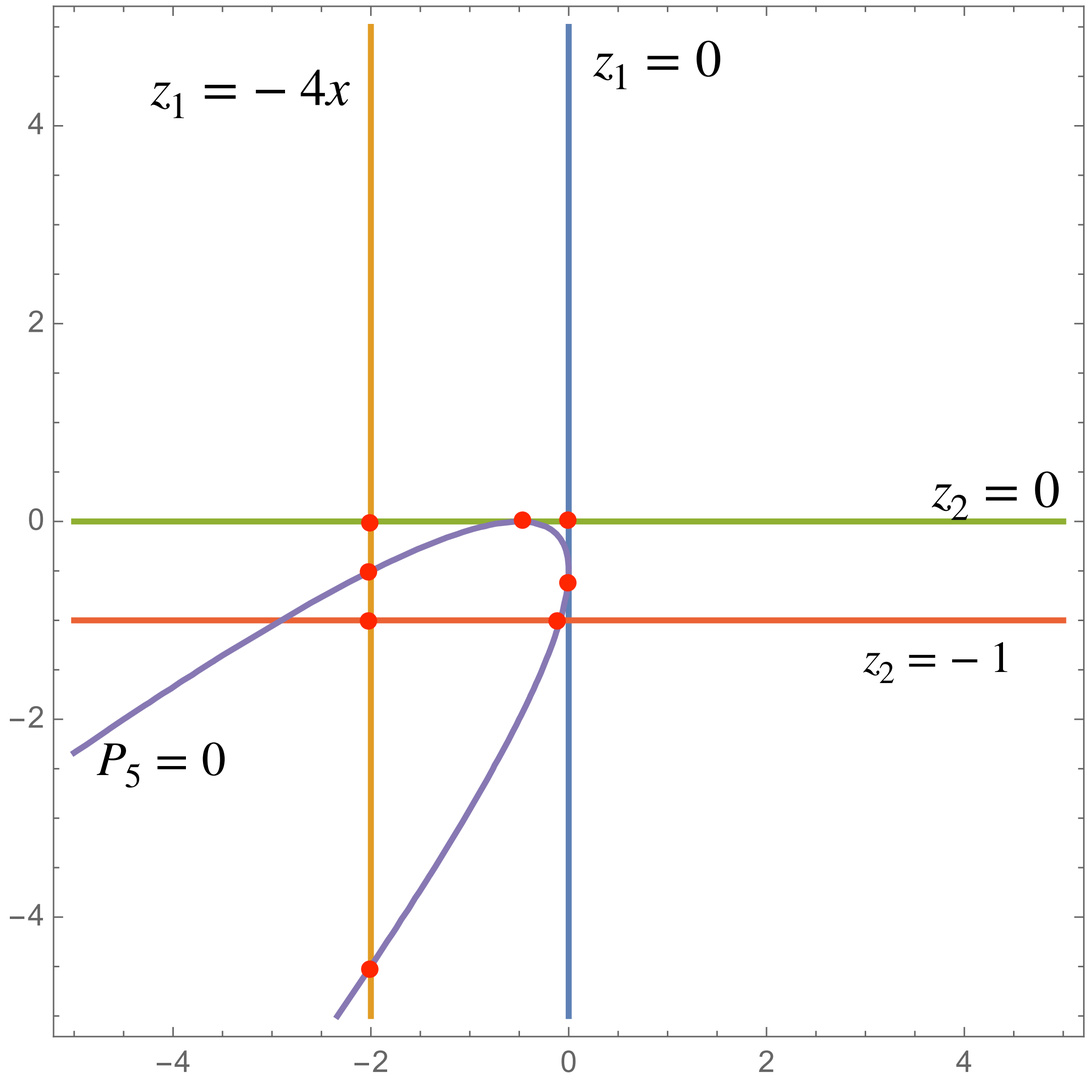
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$\begin{cases} P_3 &= z_1, \\ P_4 &= (4xz_0 + z_1), \\ P_5 &= z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2\,. \end{cases}$$

$$U = \prod_{j=0}^{5} P_j^{\alpha_j(\varepsilon)}(z_0, z, x)\,.$$

(z_0, z_1, z_2)

$p^2 \to 1,$
$m_e \to x\,.$



[Claude Duhr, Federico Gasparotto, Christoph Nega, Lorenzo Tancredi and Stefan Weinzierl '24]

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

Number of Residues

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

Number of Residues

The maximal number of non-zero consecutive residues of $\Psi^0_{\mu_1,..,\mu_N}[Q]$.

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

$u_m(z_1, z_2) = (z_1 - p_1)^{-\frac{1}{2}-\varepsilon}(z_1 - p_2)^{-\varepsilon}(z_1 - p_3)^{2\varepsilon}$

Number of Residues

The maximal number of non-zero consecutive residues of $\Psi^0_{\mu_1,..,\mu_N}[Q]$.

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

$u^0(z_1, z_2) = (z_1 - p_1)^{-\frac{1}{2}}$

Number of Residues

The maximal number of non-zero consecutive residues of $\Psi^0_{\mu_1,..,\mu_N}[Q]$.

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

$$u^0(z_1, z_2) = (z_1 - p_1)^{-\frac{1}{2}}$$

$$\Psi^0 = \frac{C_\bullet \, dz_1 \wedge dz_2}{(z_1-p_1)^{\frac{1}{2}}(z_1-p_3)(z_2-p_2)}$$

Number of Residues

The maximal number of non-zero consecutive residues of $\Psi^0_{\mu_1,..,\mu_N}[Q]$.

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

$u^0(z_1, z_2) = (z_1 - p_1)^{-\frac{1}{2}}$

$$\Psi^0 = \frac{C_\bullet \, dz_1 \wedge dz_2}{(z_1-p_1)^{\frac{1}{2}}(z_1-p_3)(z_2-p_2)}$$

$$o = \left| \lfloor \mathrm{Min}\{-\frac{3}{2}, -2\} \rfloor \right| = 2$$

Number of Residues

The maximal number of non-zero consecutive residues of $\Psi^0_{\mu_1,..,\mu_N}[Q]$.

Definitions: Pole order and Residue

For every differential form $\Psi_{\mu_1,..,\mu_N}[Q]$ we define two attributes the pole order and number of residues:

$$\Psi^0_{\mu_1,...,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U_m(z) \, \hat{\Phi}_{\mu_1,...,\mu_D}[Q] \, \eta \Big|_{\varepsilon \to 0}.$$

Pole order

The differential form has a Laurent series at every point p:

$$\Psi^0_{\mu_1,..,\mu_N}[Q] = \sum_{l_p} a_{l_p} (z-p)^{\overrightarrow{l_p}},$$

We define the pole order as:

$$o = \left| \lfloor \mathrm{Min}(|\overrightarrow{l_p}|) \rfloor \right|$$

$u^0(z_1, z_2) = (z_1 - p_1)^{-\frac{1}{2}}$

$$\Psi^0 = \frac{C_\bullet \, dz_1 \wedge dz_2}{(z_1-p_1)^{\frac{1}{2}}(z_1-p_3)(z_2-p_2)}$$

$$o = \left| \lfloor \mathrm{Min}\{-\frac{3}{2}, -2\} \rfloor \right| = 2$$

Number of Residues

The maximal number of non-zero consecutive residues of $\Psi^0_{\mu_1,..,\mu_N}[Q]$.

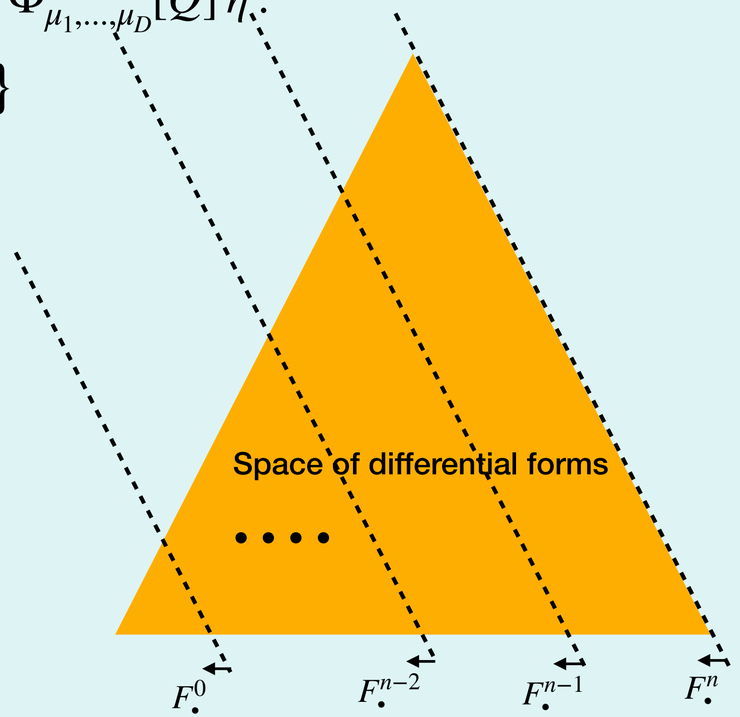
$$r = 2$$

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta_\cdot$$

$$\{r, o, |\mu|, n\}$$

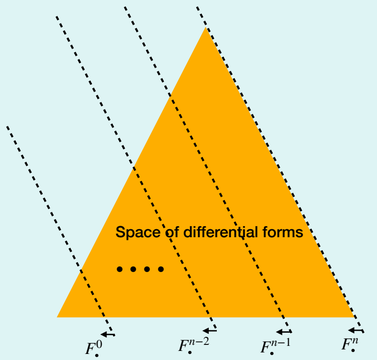
- Combinatorics Filtration
- Geometric Filtration
- Weight Filtration



Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$

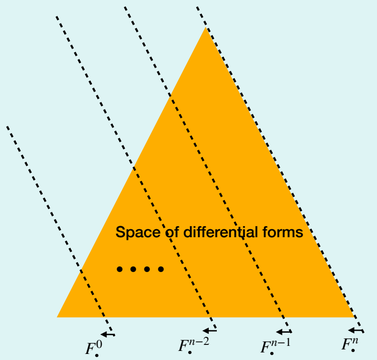


- Combinatorics Filtration
- Geometric Filtration
- Weight Filtration

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

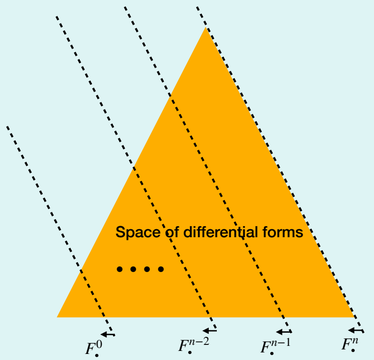
$$\{r, o, |\mu|, n\}$$



Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$



- Combinatorics Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{comb}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{comb}\Omega^n_\omega, \quad n - |\mu| \geq p \, .$$

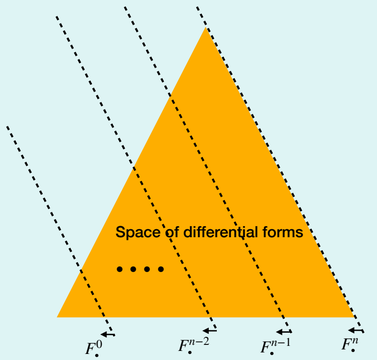
Filters by $|\mu|$.

$$\{0\} \subset F^n_{comb}\Omega^n_\omega \subset F^{n-1}_{comb}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{comb}\Omega^n_\omega \, .$$

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$



- Combinatorics Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{comb}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{comb}\Omega^n_\omega, \quad n - |\mu| \geq p \, .$$

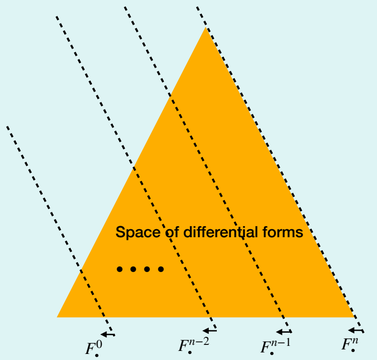
Filters by $|\mu|$.

$$\{0\} \subset F^n_{comb}\Omega^n_\omega \subset F^{n-1}_{comb}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{comb}\Omega^n_\omega \, .$$

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$



- Geometric Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{geom}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{geom}\Omega^n_\omega, \quad n + r - o \geq p \, .$$

Filters by the pole order o.

$$\{0\} \subset F^n_{geom}\Omega^n_\omega \subset F^{n-1}_{geom}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{geom}\Omega^n_\omega \, .$$

- Combinatorics Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{comb}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{comb}\Omega^n_\omega, \quad n - |\mu| \geq p \, .$$

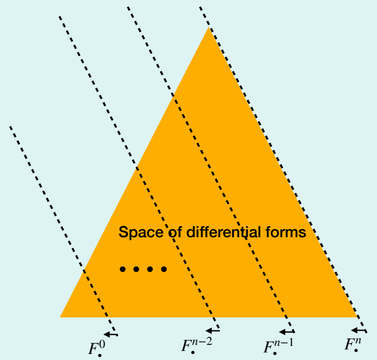
Filters by $|\mu|$.

$$\{0\} \subset F^n_{comb}\Omega^n_\omega \subset F^{n-1}_{comb}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{comb}\Omega^n_\omega \, .$$

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$



- Combinatorics Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{comb}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{comb}\Omega^n_\omega, \quad n - |\mu| \geq p \, .$$

Filters by $|\mu|$.

$$\{0\} \subset F^n_{comb}\Omega^n_\omega \subset F^{n-1}_{comb}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{comb}\Omega^n_\omega \, .$$

- Geometric Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{geom}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{geom}\Omega^n_\omega, \quad n + r - o \geq p \, .$$

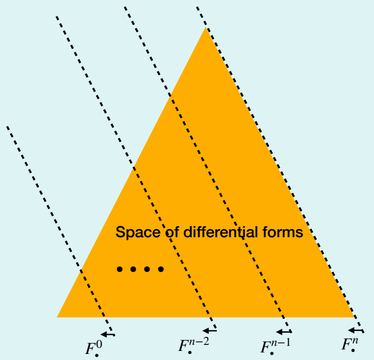
Filters by the pole order o.

$$\{0\} \subset F^n_{geom}\Omega^n_\omega \subset F^{n-1}_{geom}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{geom}\Omega^n_\omega \, .$$

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$



- Weight Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $W_w\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in W_w\Omega^n_\omega, \quad n + r \leq w \, .$$

Filters by the residue number r.

$$\{0\} \subset W_n\Omega^n_\omega \subset W_{n+1}\Omega^n_\omega \subset \ldots \subset W_\infty\Omega^n_\omega \, .$$

- Combinatorics Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{comb}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{comb}\Omega^n_\omega, \quad n - |\mu| \geq p \, .$$

Filters by $|\mu|$.

$$\{0\} \subset F^n_{comb}\Omega^n_\omega \subset F^{n-1}_{comb}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{comb}\Omega^n_\omega \, .$$

- Geometric Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{geom}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{geom}\Omega^n_\omega, \quad n + r - o \geq p \, .$$

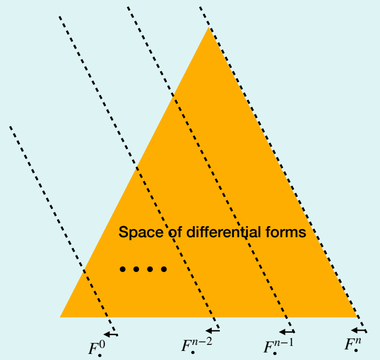
Filters by the pole order o.

$$\{0\} \subset F^n_{geom}\Omega^n_\omega \subset F^{n-1}_{geom}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{geom}\Omega^n_\omega \, .$$

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$



- Combinatorics Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{comb}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{comb}\Omega^n_\omega, \quad n - |\mu| \geq p \, .$$

Filters by $|\mu|$.

$$\{0\} \subset F^n_{comb}\Omega^n_\omega \subset F^{n-1}_{comb}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{comb}\Omega^n_\omega \, .$$

- Geometric Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{geom}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{geom}\Omega^n_\omega, \quad n + r - o \geq p \, .$$

Filters by the pole order o.

$$\{0\} \subset F^n_{geom}\Omega^n_\omega \subset F^{n-1}_{geom}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{geom}\Omega^n_\omega \, .$$

- Weight Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $W_w\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in W_w\Omega^n_\omega, \quad n + r \leq w \, .$$

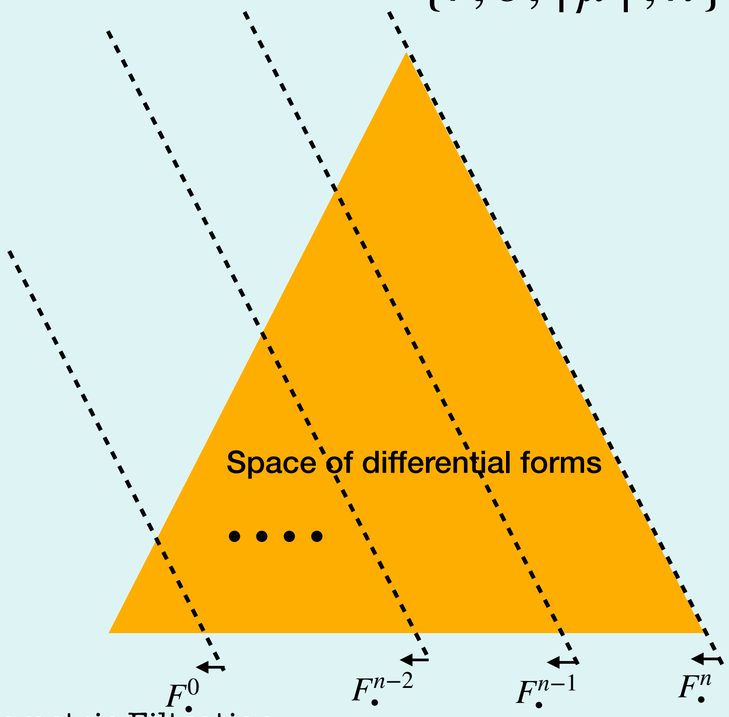
Filters by the residue number r.

$$\{0\} \subset W_n\Omega^n_\omega \subset W_{n+1}\Omega^n_\omega \subset \ldots \subset W_\infty\Omega^n_\omega \, .$$

Definitions: Filtration

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] = C_{Baikov} C_\varepsilon \, U(z) \, \hat{\Phi}_{\mu_1,\ldots,\mu_D}[Q] \, \eta \, .$$

$$\{r, o, |\mu|, n\}$$



- Combinatorics Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{comb}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{comb}\Omega^n_\omega, \quad n - |\mu| \geq p \, .$$

Filters by $|\mu|$.

$$\{0\} \subset F^n_{comb}\Omega^n_\omega \subset F^{n-1}_{comb}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{comb}\Omega^n_\omega \, .$$

- Geometric Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $F^P_{geom}\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in F^P_{geom}\Omega^n_\omega, \quad n + r - o \geq p \, .$$

Filters by the pole order o.

$$\{0\} \subset F^n_{geom}\Omega^n_\omega \subset F^{n-1}_{geom}\Omega^n_\omega \subset \ldots \subset F^{-\infty}_{geom}\Omega^n_\omega \, .$$

- Weight Filtration

We define a differential form $\Psi_{\mu_1,\ldots,\mu_D}[Q]$ belongs to $W_w\Omega^n_\omega$ iff

$$\Psi_{\mu_1,\ldots,\mu_D}[Q] \in W_w\Omega^n_\omega, \quad n + r \leq w \, .$$

Filters by the residue number r.

$$\{0\} \subset W_n\Omega^n_\omega \subset W_{n+1}\Omega^n_\omega \subset \ldots \subset W_\infty\Omega^n_\omega \, .$$

Grading:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n\,.$$

Grading:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n\,.$$

We define two Gradings :

Grading:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n\,.$$

We define two Gradings :

$$\Omega^{p,q}_{geom} = \mathrm{Gr}^p_{F_{geom}} \mathrm{Gr}^{p+q}_W \Omega^n_\omega,$$

Grading:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n\,.$$

We define two Gradings :

$$\Omega^{p,q}_{geom} = \mathrm{Gr}^p_{F_{geom}} \mathrm{Gr}^{p+q}_W \Omega^n_\omega,$$

$$\Omega^{p',q'}_{comb} = \mathrm{Gr}^{p'}_{F_{comb}} \mathrm{Gr}^{p'+q'}_W \Omega^n_\omega\,.$$

Grading:

$$\mathrm{Gr}_\bullet^n = \frac{F_\bullet^n}{F_\bullet^{n-1}},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n\,.$$

We define two Gradings :

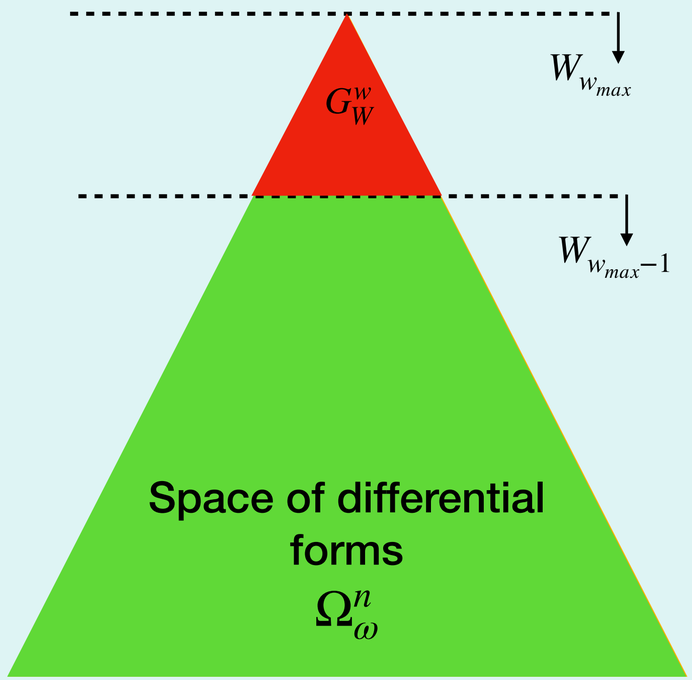
$$\Omega^{p,q}_{geom} = \mathrm{Gr}^p_{F_{geom}} \mathrm{Gr}^{p+q}_W \Omega^n_\omega,$$

$$\Omega^{p',q'}_{comb} = \mathrm{Gr}^{p'}_{F_{comb}} \mathrm{Gr}^{p'+q'}_W \Omega^n_\omega\,.$$

Grading:

$$\mathrm{Gr}^n_\bullet = \frac{F^n_\bullet}{F^{n-1}_\bullet},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n\,.$$



We define two Gradings :

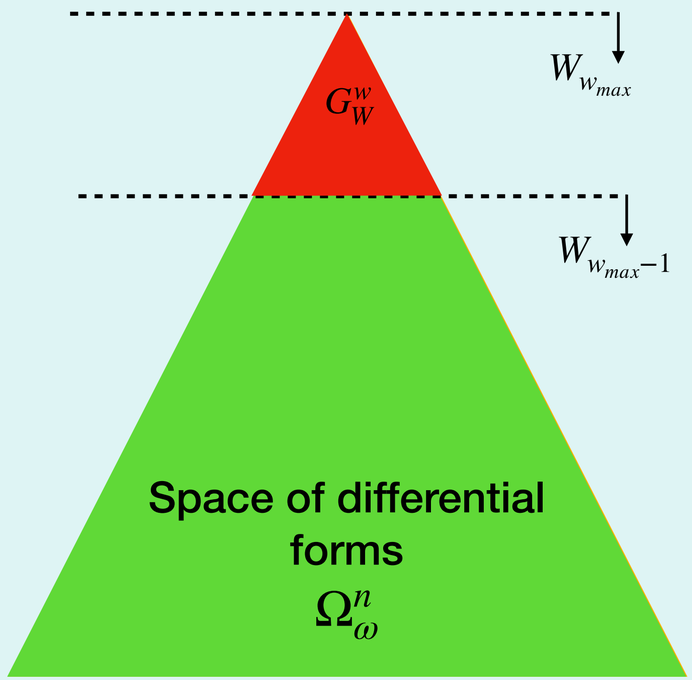
$$\Omega^{p,q}_{geom} = \mathrm{Gr}^p_{F_{geom}} \mathrm{Gr}^{p+q}_W \Omega^n_\omega,$$

$$\Omega^{p',q'}_{comb} = \mathrm{Gr}^{p'}_{F_{comb}} \mathrm{Gr}^{p'+q'}_W \Omega^n_\omega\,.$$

Grading:

$$\mathrm{Gr}^n_\bullet = \frac{F^n_\bullet}{F^{n-1}_\bullet},$$

$$\mathscr{G}(S) = \bigoplus_{n\in\mathbb{N}} G_n\,.$$



We define two Gradings :

$$\Omega^{p,q}_{geom} = \mathrm{Gr}^p_{F_{geom}} \mathrm{Gr}^{p+q}_W \Omega^n_\omega,$$

$$\Omega^{p',q'}_{comb} = \mathrm{Gr}^{p'}_{F_{comb}} \mathrm{Gr}^{p'+q'}_W \Omega^n_\omega\,.$$

The grading corresponds to the differential forms with fixed r.

By localising on even polynomials

$$\{P_i = 0\}, \qquad i \in I_{even},$$

$$U^n_m(z) \xrightarrow{\mathrm{loc}_r} \tilde{U}^{n-(r)}_m(\tilde{z})\,.$$

Find basis of forms of at *least* r residues.

The Algorithm

For any given $U(z)$:

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

Setup:

1. Generate forms.
2. Compute r and o.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

Setup:

1. Generate forms.
2. Compute r and o.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0, w, o, |\mu|, \dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

Setup:

1. Generate forms.
2. Compute r and o.

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a_i = r$ for all the pre image of the **basis forms**.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a, w, o, |\mu|, \dots)$. a from the **previous step** and rest of the forms of $a = 0$. Then assign $a = -(r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0, w, o, |\mu|, \dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = r$ for all the pre image of the **basis forms**.

Step l:

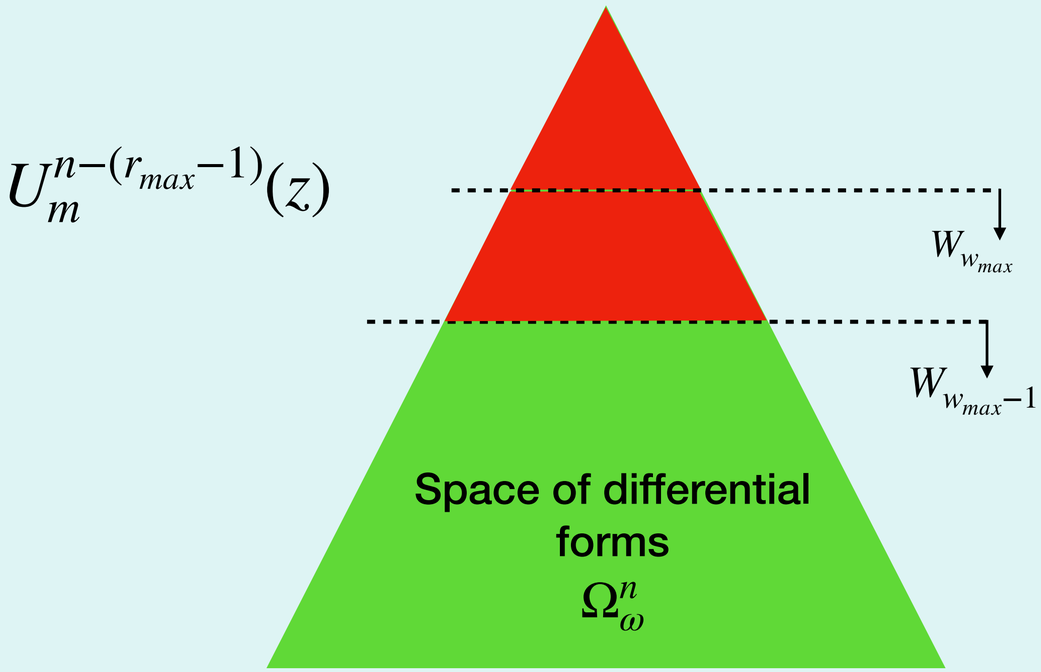
1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = (r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

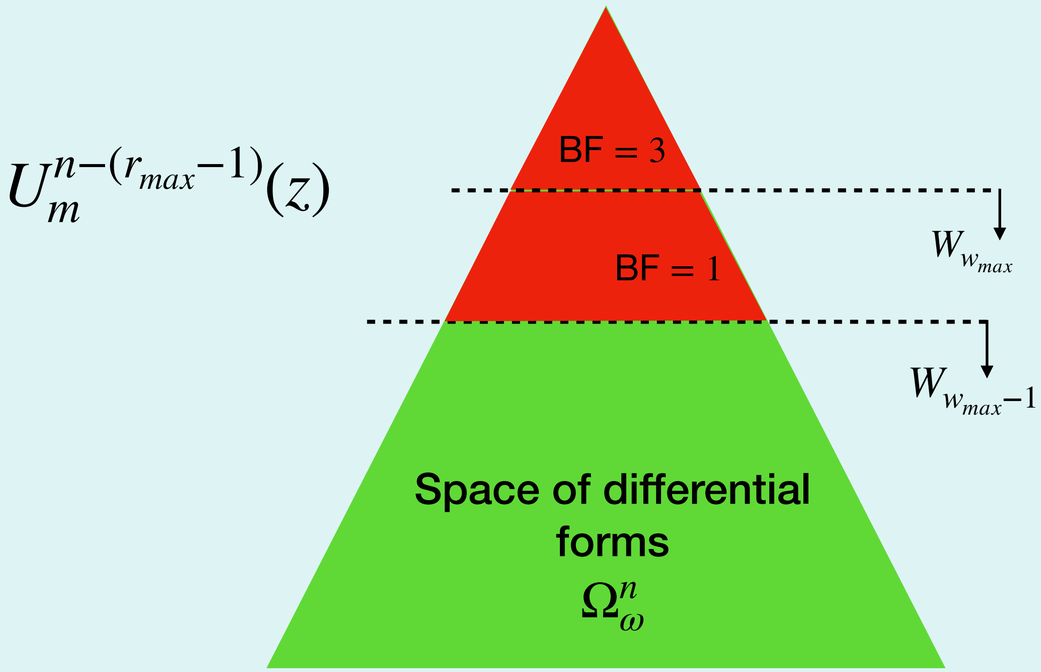
Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = -(r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

Step l:

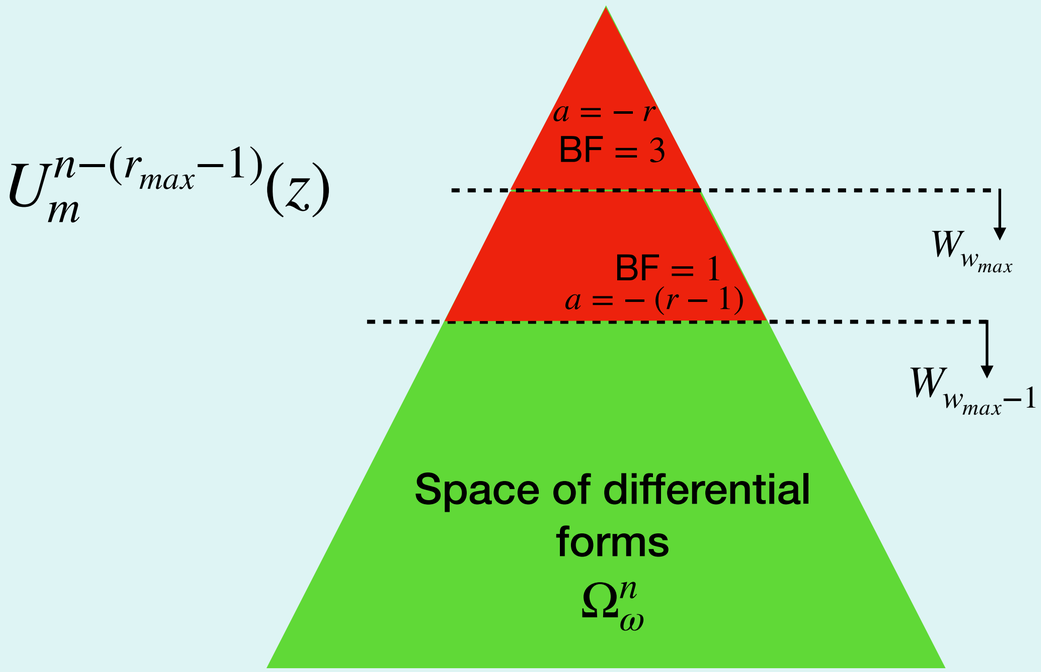
1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = -(r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

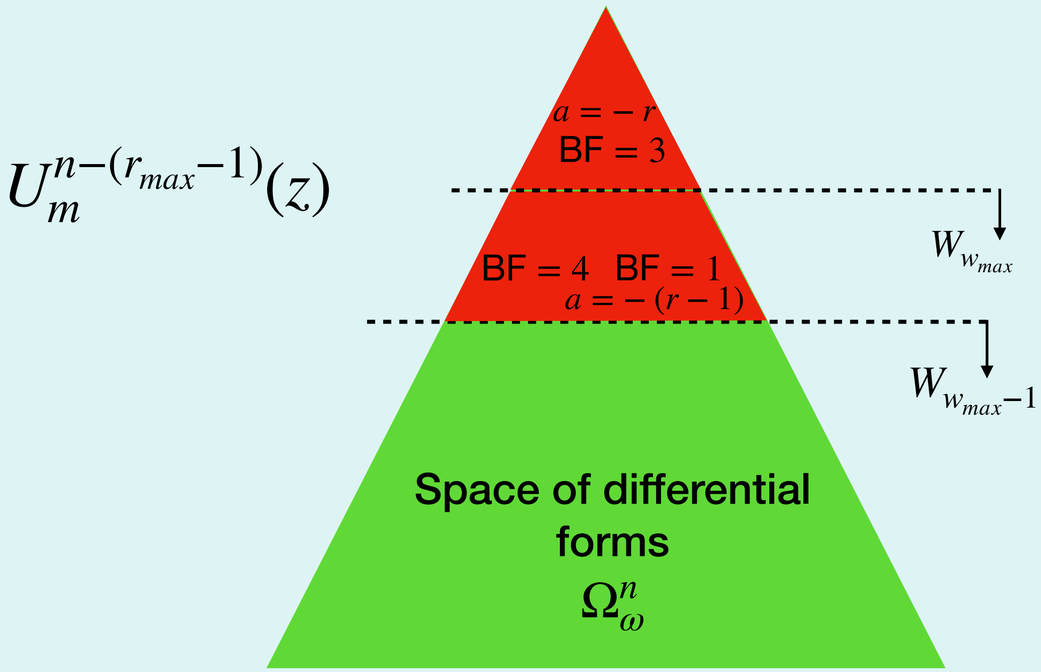
Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = -(r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

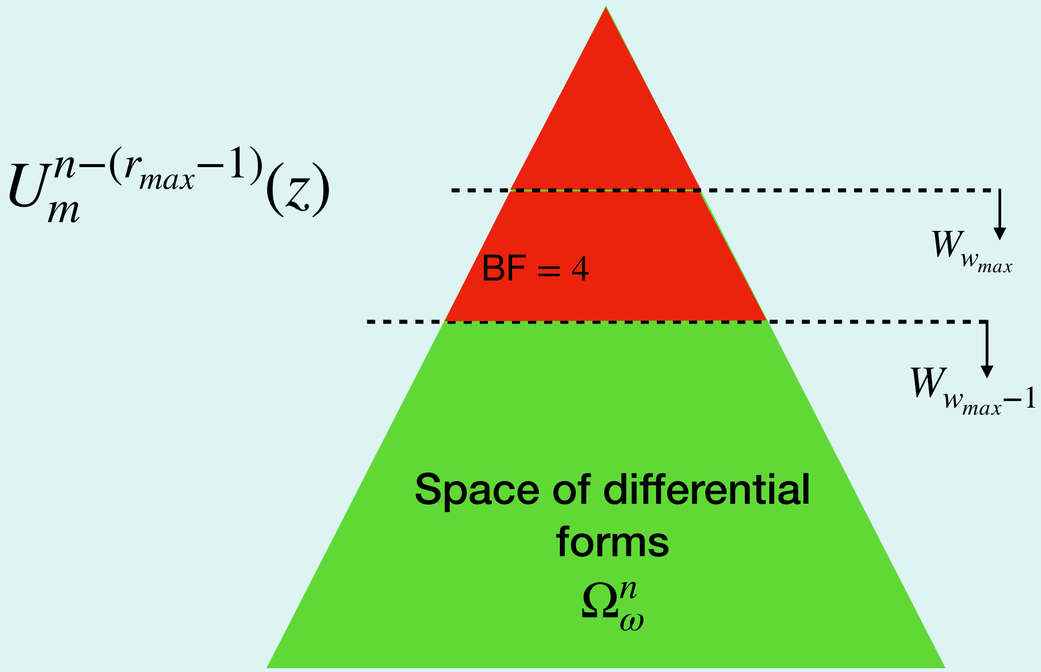
Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = -(r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = r$ for all the pre image of the **basis forms**.

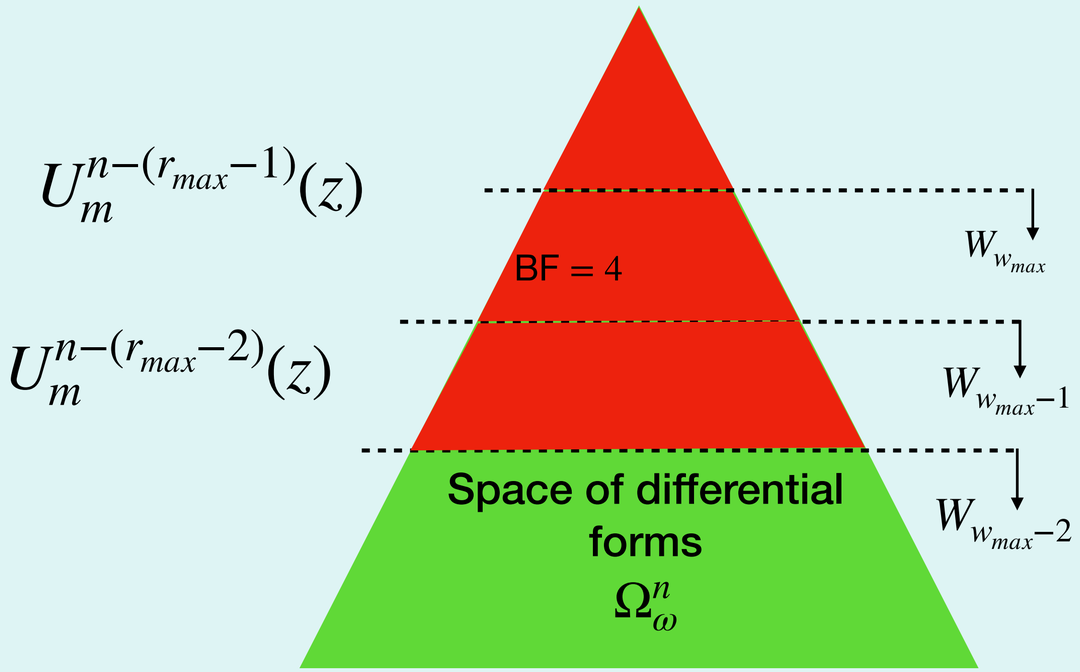
Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = (r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

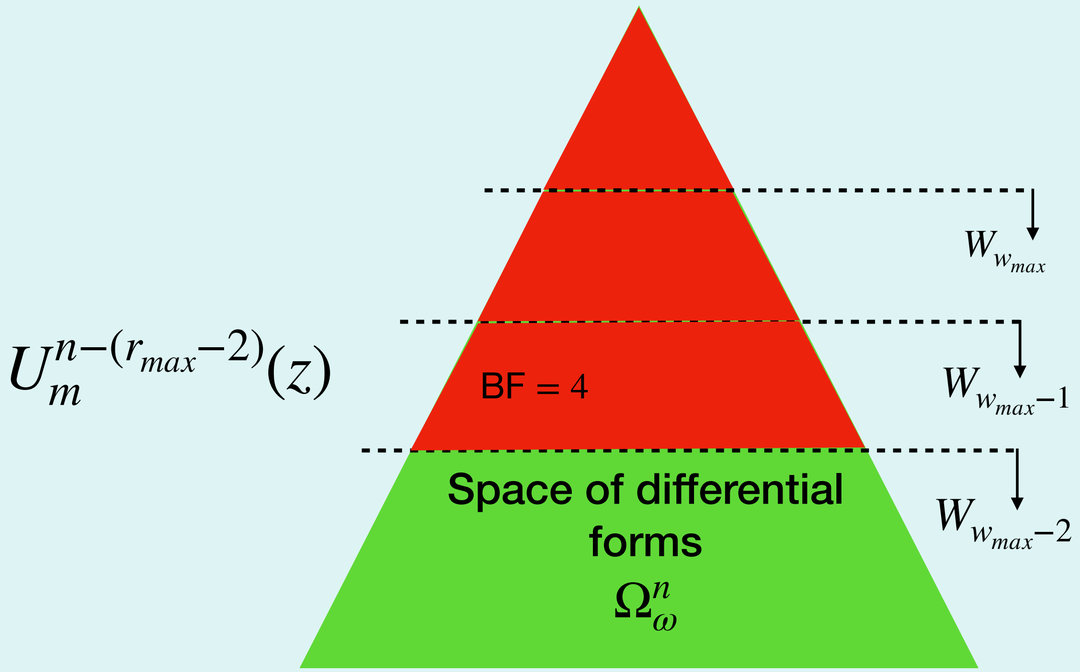
Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = -(r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = r$ for all the pre image of the **basis forms**.

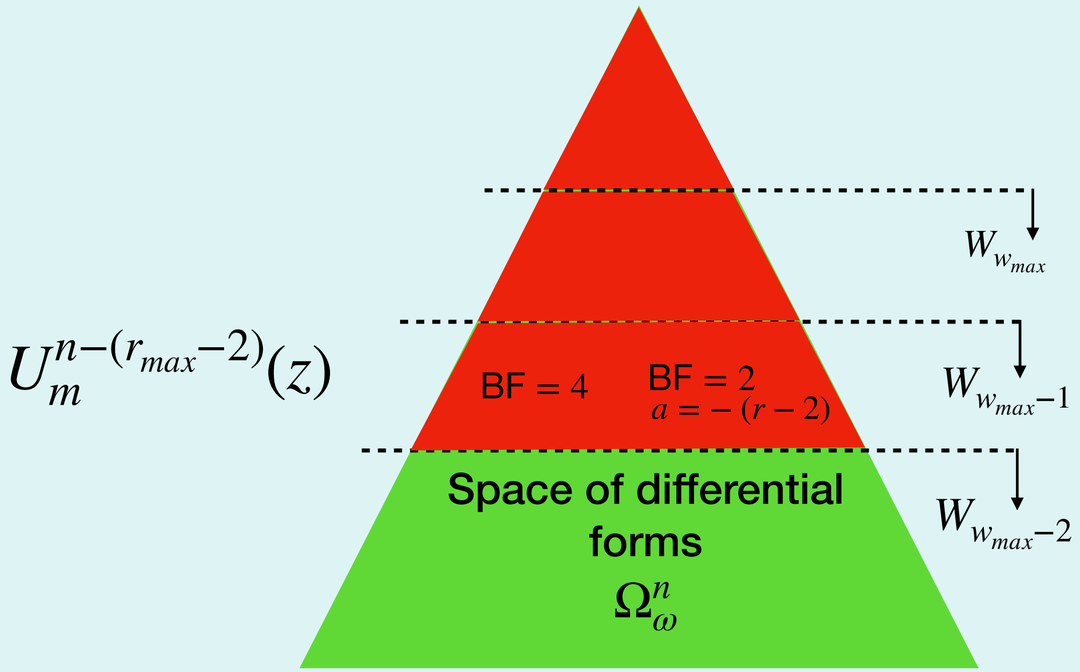
Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = (r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = -r$ for all the pre image of the **basis forms**.

Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = -(r_{max}-l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = r$ for all the pre image of the **basis forms**.

Step l:

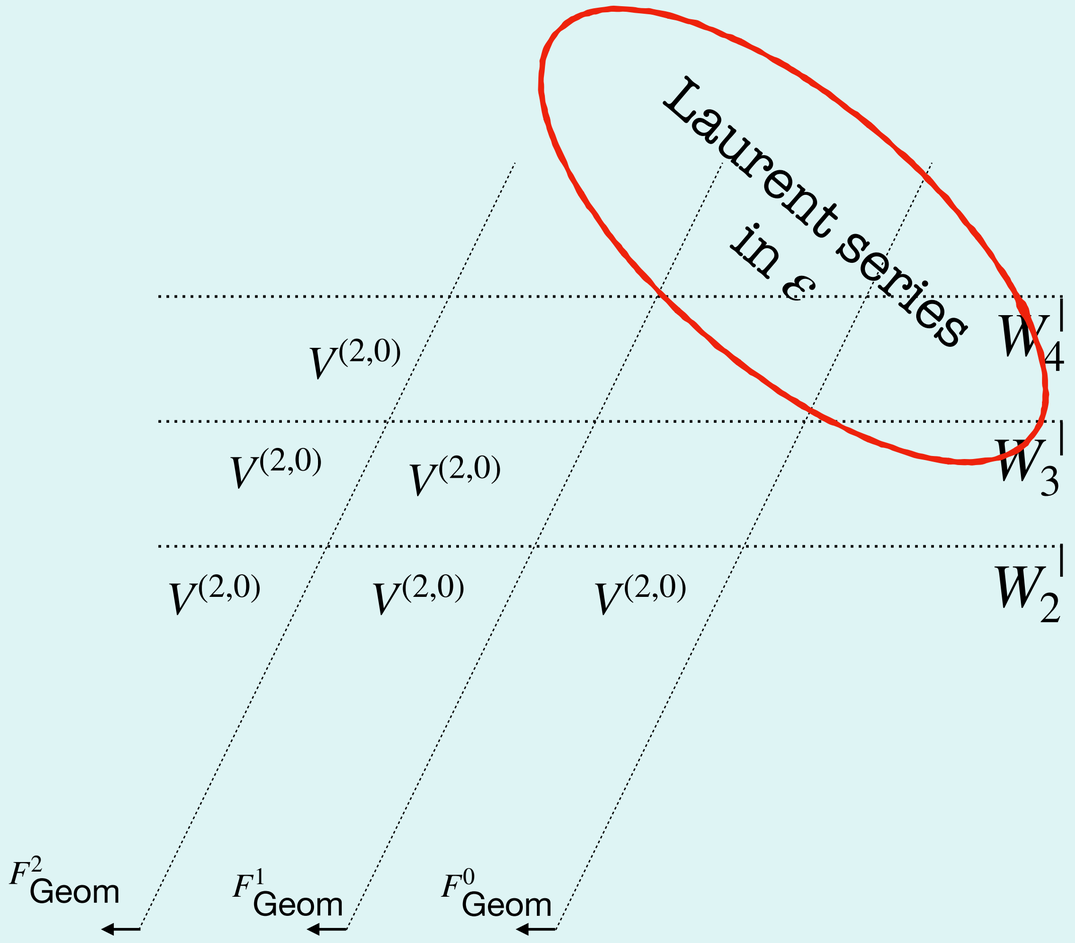
1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of 0. Then assign $a = (r_{max} - l)$ for all the pre image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = r$ for all the pre image of the **basis forms**.

Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

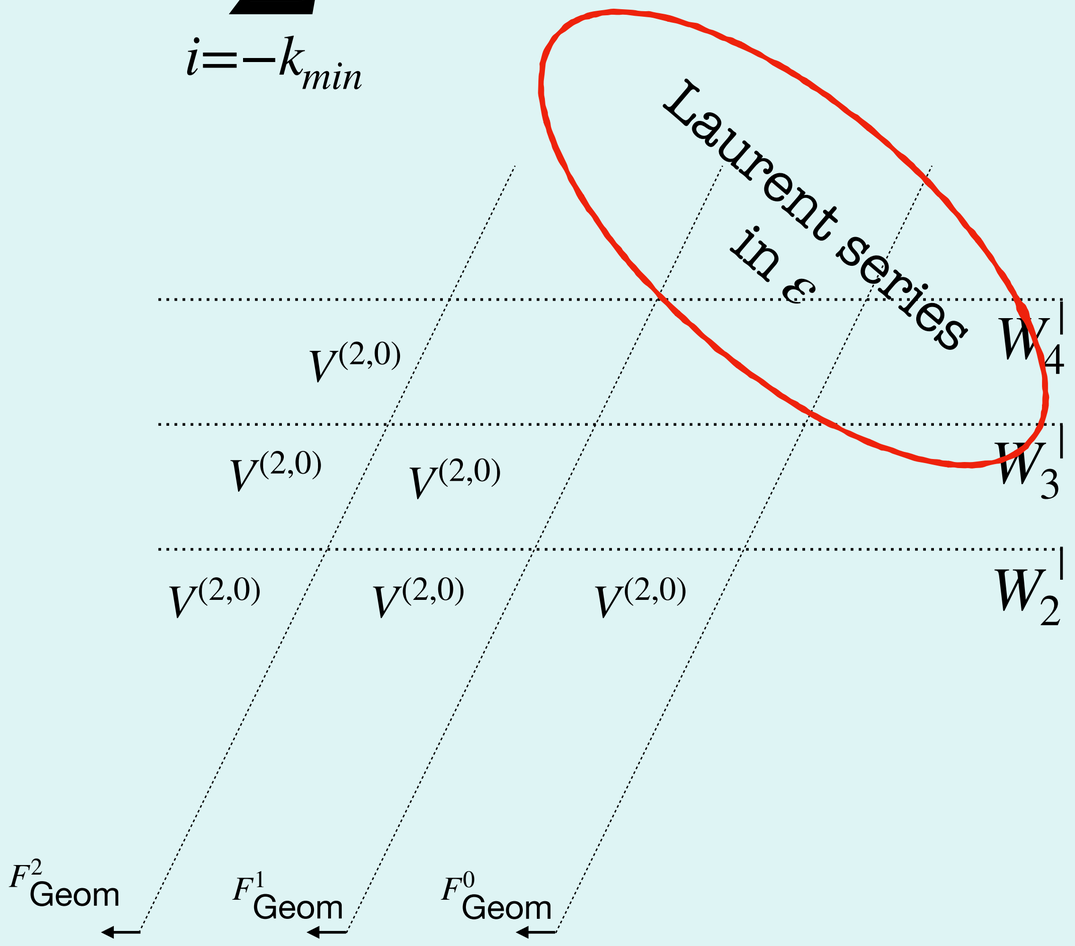
[The ε-Collaboration '25]

The Algorithm

For any given $U(z)$:

- IBP generator
- Linear system solver
- Pole order counter
- Residue counter

$$dJ = \sum_{i=-k_{min}}^{1} \varepsilon^i A^{(i)}(x) \cdot J$$



Setup:

1. Generate forms.
2. Compute r and o.

Step I:

1. Start at the top $\{U_m^{n-(r_{max})}(z)\}$, generate the IBPs of that system.
2. Use the Laporta with the ordering $(0,w,o,|\mu|,\dots)$. assign $a = r$ for all the pre image of the **basis forms**.

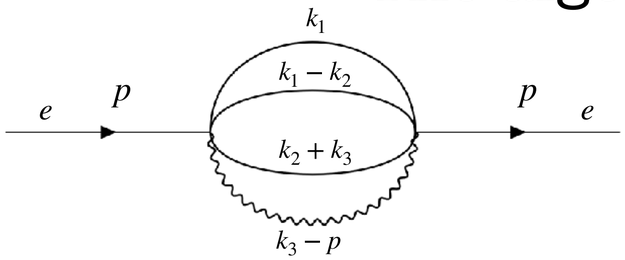
Step l:

1. Consider now $\{U_m^{n-(r_{max}-l)}(z)\}$ generate the IBPs of the system.
2. Use the Laporta with the ordering $(a,w,o,|\mu|,\dots)$. a from the **previous step** and rest of the forms of image of the new **basis forms**.
3. Repeat the step until $U_m^n(z)$.

[The ε-Collaboration '25]

Electron—Self energy

The algorithm



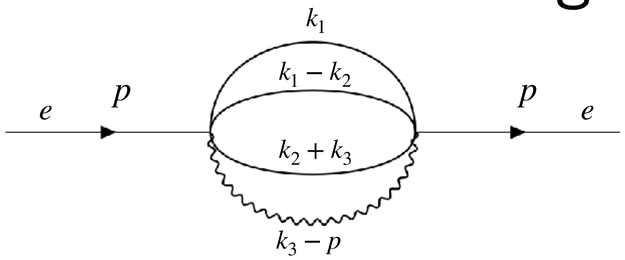
$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

Electron—Self energy

The algorithm

$$n + r_{max} = w_{max} = 4.$$

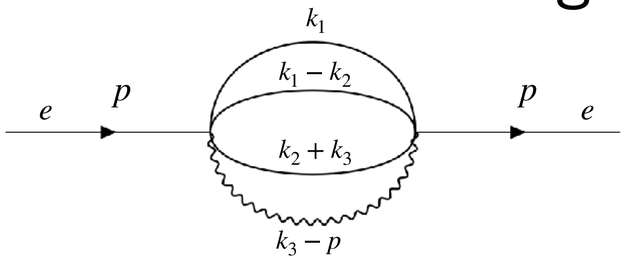


$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

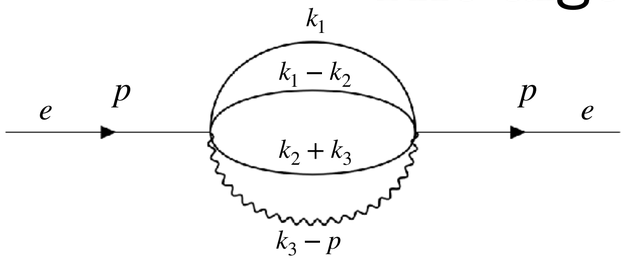
$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_0 = z_0 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

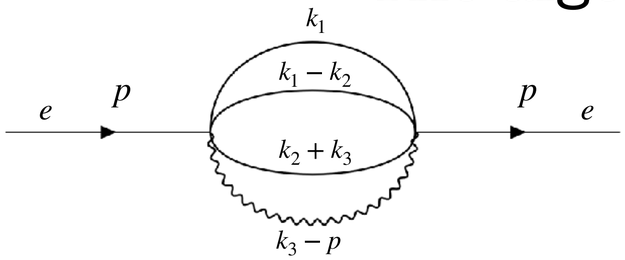
$$P_0 = z_0 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_0 \to 0} = z_2^{-\varepsilon} z_1^{3\varepsilon} (z_2 - z_1)^{-2\varepsilon}\,.$$

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_0 = z_0 = 0$$

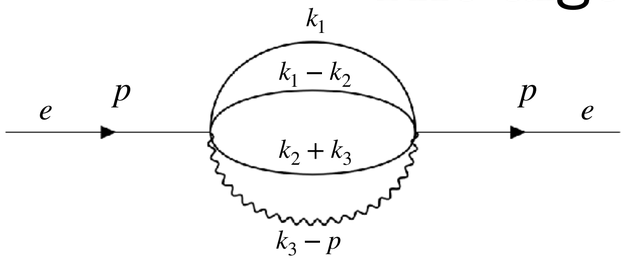
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_0 \to 0} = z_2^{-\varepsilon} z_1^{3\varepsilon} (z_2 - z_1)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

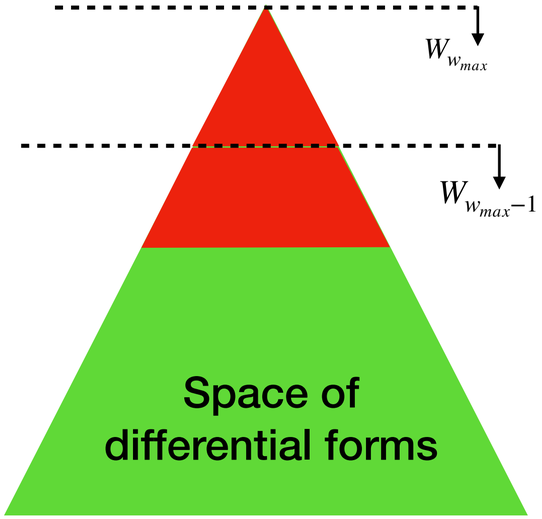
$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_0 = z_0 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

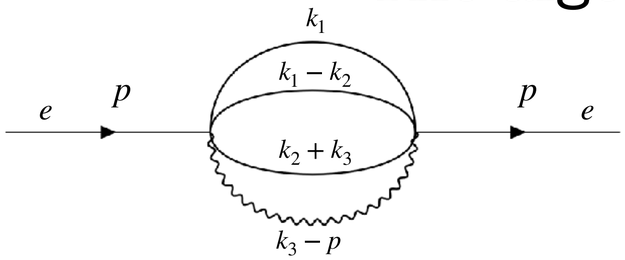
$$U_m(z_0, z, x)\Big|_{z_0 \to 0} = z_2^{-\varepsilon} z_1^{3\varepsilon} (z_2 - z_1)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:



Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

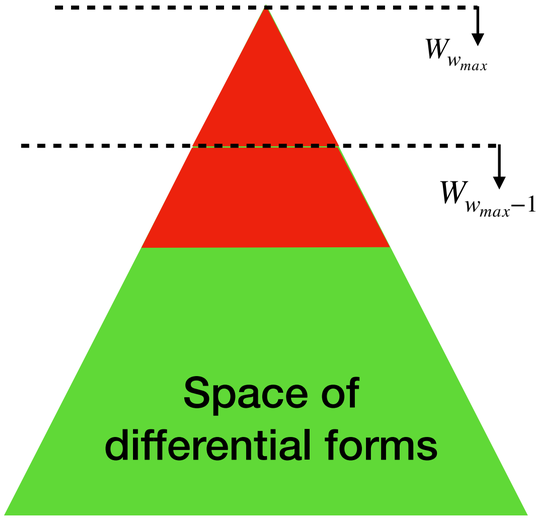
$$P_0 = z_0 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_0 \to 0} = z_2^{-\varepsilon} z_1^{3\varepsilon} (z_2 - z_1)^{-2\varepsilon}\,.$$

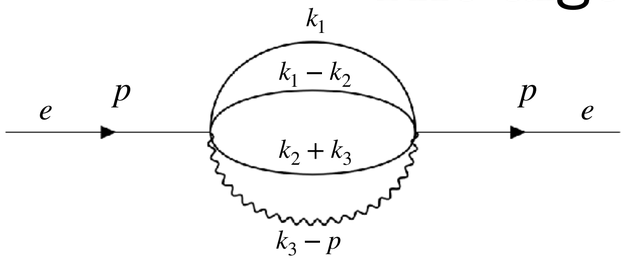
Running Laporta we get one dimension with its pre-image in the full system:

$$\chi = 1 \; dz_2 \longrightarrow \Phi = \frac{1}{z_0}\,\eta,$$



Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

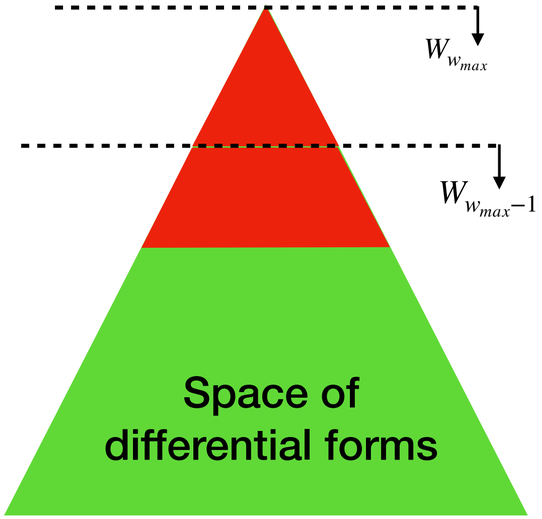
$$P_0 = z_0 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_0 \to 0} = z_2^{-\varepsilon} z_1^{3\varepsilon} (z_2 - z_1)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

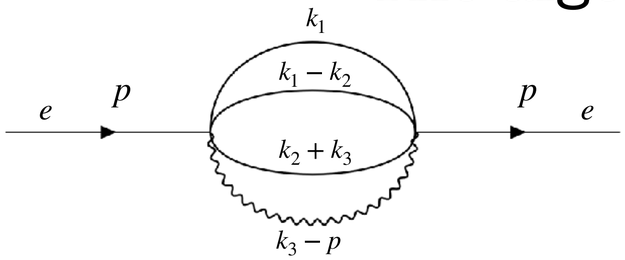
$$\chi = 1 \; dz_2 \longrightarrow \Phi = \frac{1}{z_0}\,\eta,$$



$$|\mu| = 1,\; r = 2,\; o = 2.$$

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

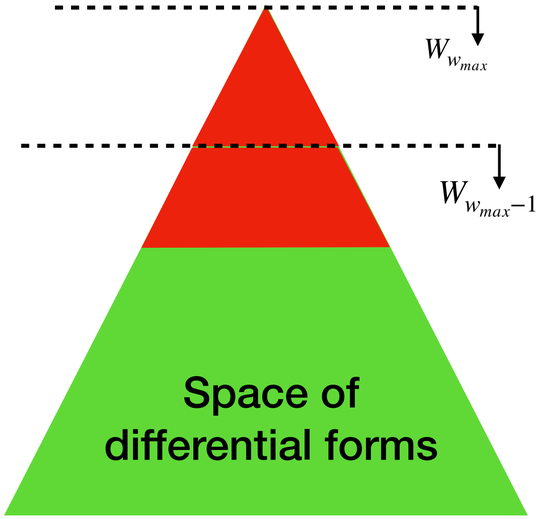
$$P_0 = z_0 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_0 \to 0} = z_2^{-\varepsilon} z_1^{3\varepsilon} (z_2 - z_1)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

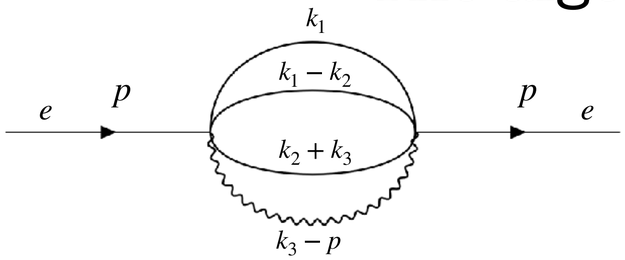
$$\chi = 1 \; dz_2 \longrightarrow \Phi = \frac{1}{z_0}\,\eta,$$



$$\Psi_{1,0,0,0,0,0}[1], \qquad |\mu| = 1, \; r = 2, \; o = 2.$$

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

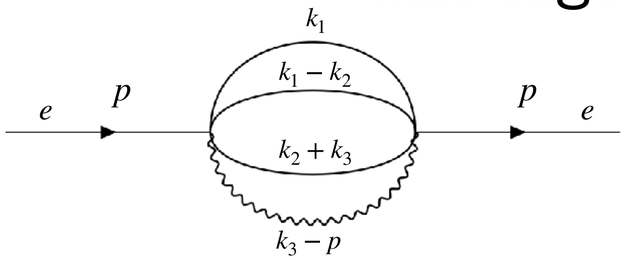
$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$\Psi_{1,0,0,0,0,0}[1], \qquad |\mu| = 1,\ r = 2,\ o = 2.$$

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

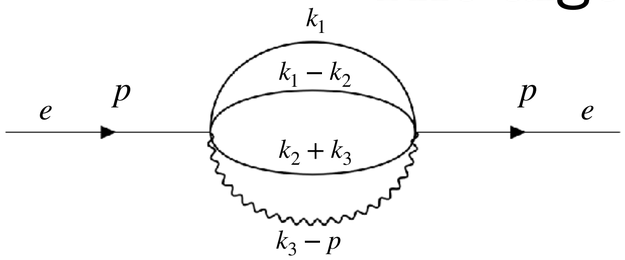
$$\Psi_{1,0,0,0,0,0}[1],$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$|\mu| = 1,\ r = 2,\ o = 2.$$

Electron—Self energy

The algorithm



$$n + r_{max} = w_{max} = 4.$$

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

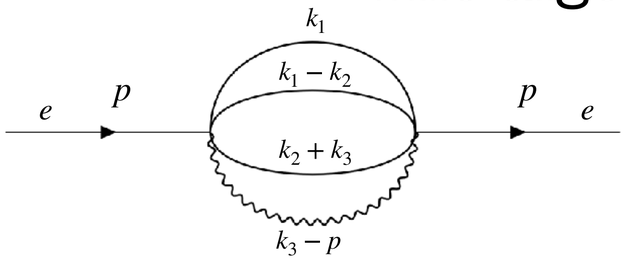
$$\Psi_{1,0,0,0,0,0}[1],$$
$$|\mu| = 1, \ r = 2, \ o = 2.$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

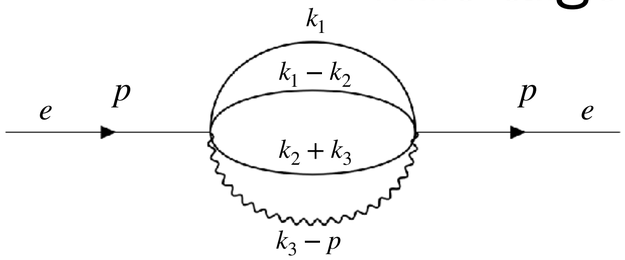
$$\Psi_{1,0,0,0,0,0}[1],$$
$$|\mu| = 1, \ r = 2, \ o = 2.$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_1 = z_2 = 0$$

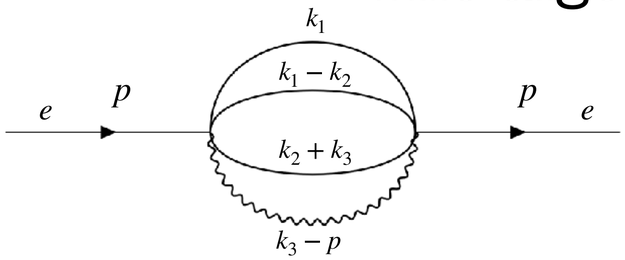
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$\Psi_{1,0,0,0,0,0}[1],$$
$$|\mu| = 1,\ r = 2,\ o = 2.$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

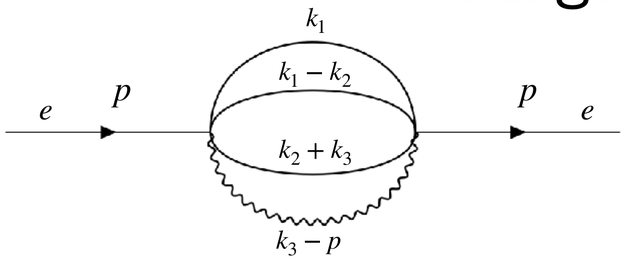
$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

$$\Psi_{1,0,0,0,0,0}[1],$$
$$|\mu| = 1,\ r = 2,\ o = 2.$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

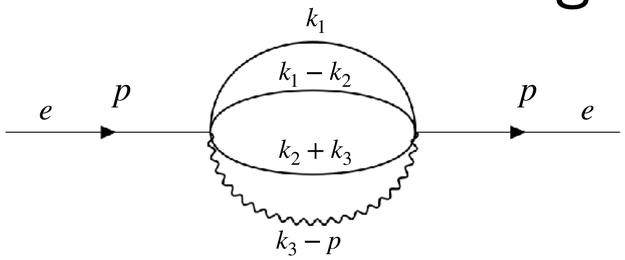
Running Laporta we get one dimension with its pre-image in the full system:

$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

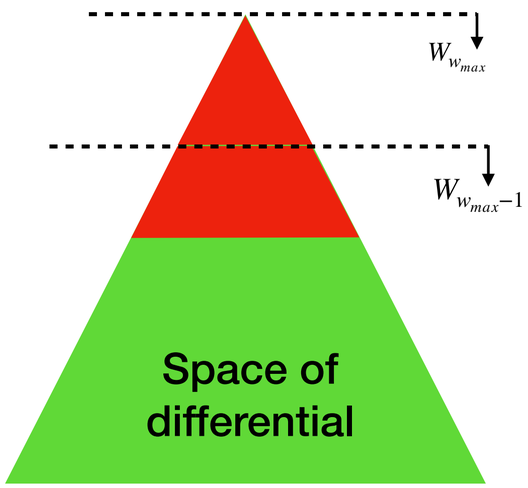
$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

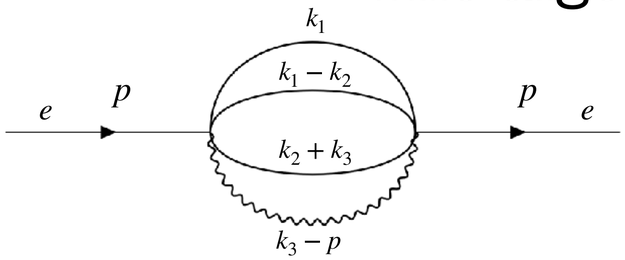
Running Laporta we get one dimension with its pre-image in the full system:



Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

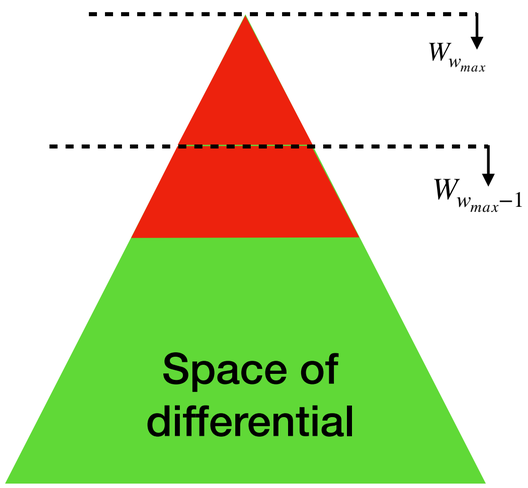
$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

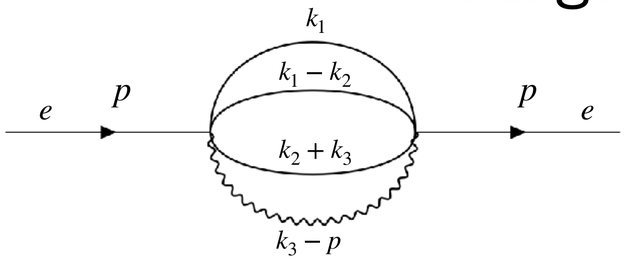
$$\chi = 1 \ dz_2 \longrightarrow \Phi = \frac{1}{z_2}\eta\,.$$



Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$\Psi_{1,0,0,0,0,0}[1],$$
$$|\mu| = 1, \ r = 2, \ o = 2.$$

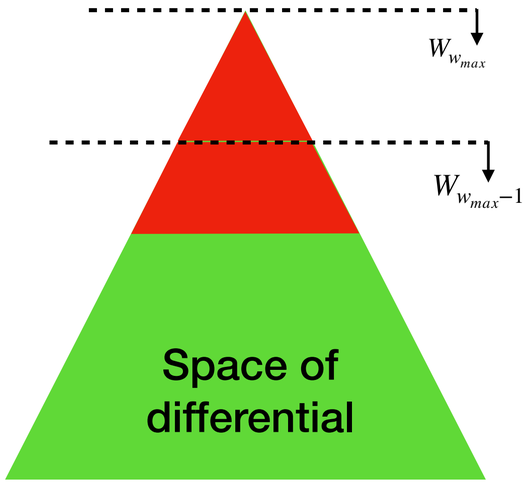
$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

$$\chi = 1 \ dz_2 \longrightarrow \Phi = \frac{1}{z_2}\eta\,.$$

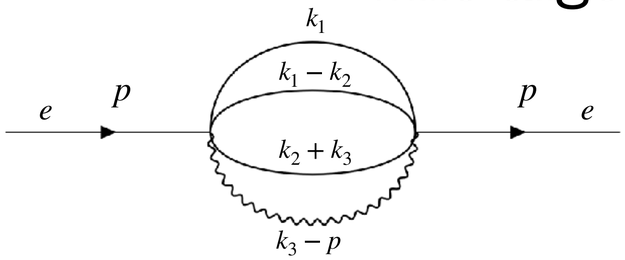


$$|\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$\Psi_{1,0,0,0,0,0}[1],$$
$$|\mu| = 1, \ r = 2, \ o = 2.$$

$$P_1 = z_2 = 0$$

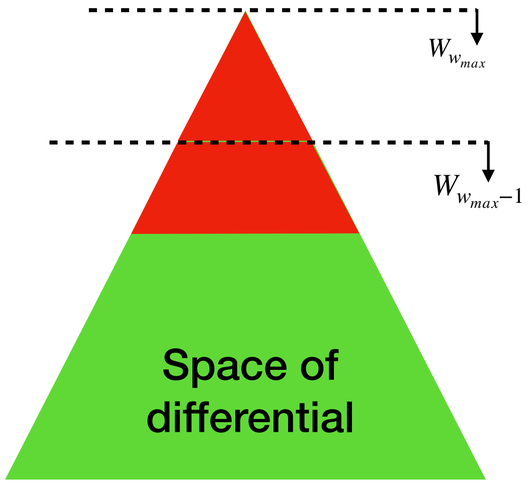
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

$$\chi = 1 \ dz_2 \longrightarrow \Phi = \frac{1}{z_2}\eta\,.$$

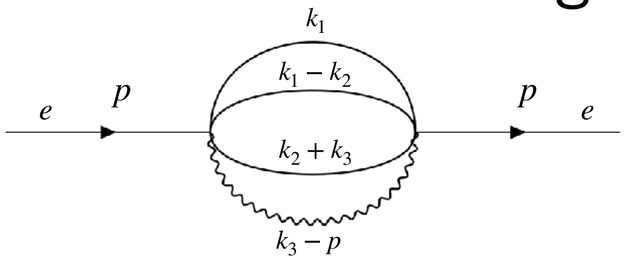
$$\Psi_{0,1,0,0,0,0}[1],$$
$$|\mu| = 1, \ r = 2, \ o = 2.$$



Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

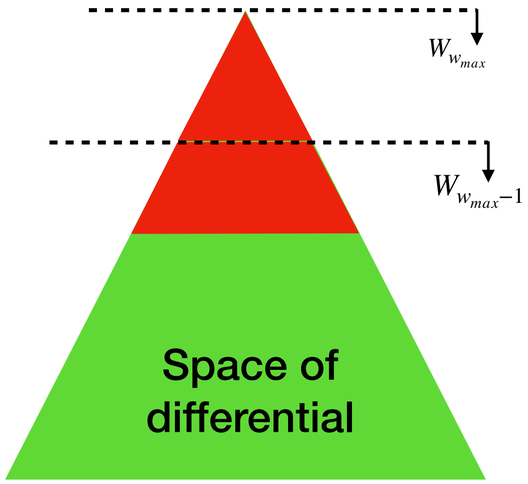
$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

$$\chi = 1 \; dz_2 \longrightarrow \Phi = \frac{1}{z_2}\,\eta\,.$$



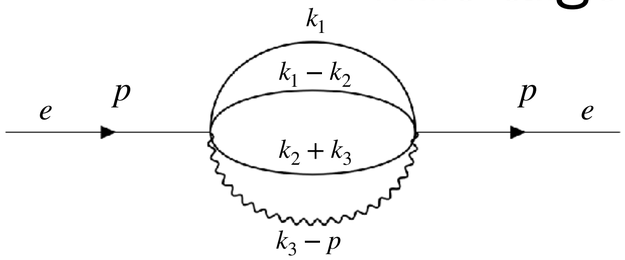
$$|\mu| = 1,\ r = 2,\ o = 2.$$

$$\frac{\Psi_{1,0,0,0,0,0}[1],}{\Psi_{0,1,0,0,0,0}[1],} \quad |\mu| = 1,\ r = 2,\ o = 2.$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

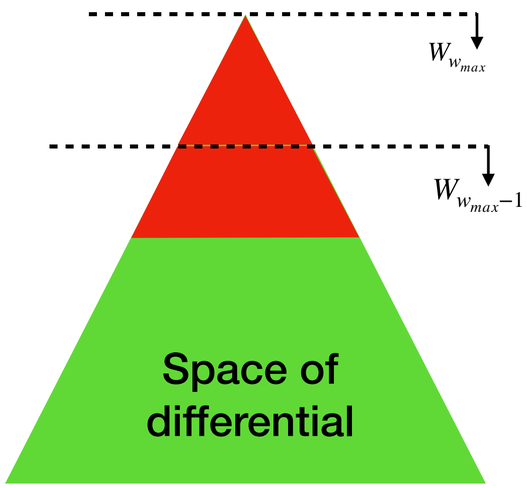
$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:

$$\chi = 1 \; dz_2 \longrightarrow \Phi = \frac{1}{z_2}\,\eta\,.$$



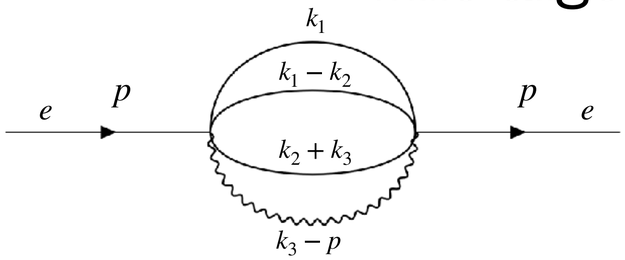
$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \; r = 2, \; o = 2.$$

$$\Psi_{0,1,0,0,0,0}[1], \quad |\mu| = 1, \; r = 2, \; o = 2.$$

Electron—Self energy

sector: 15

The algorithm



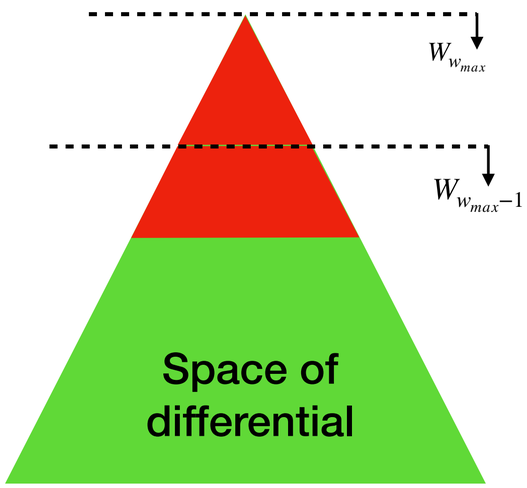
$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_1 = z_2 = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2 \to 0} = z_0^{3\varepsilon} z_1^{-\frac{1}{2}} (z_1 + 4xz_0)^{-\frac{1}{2}-\varepsilon} (z_1 + xz_0)^{-2\varepsilon}\,.$$

Running Laporta we get one dimension with its pre-image in the full system:



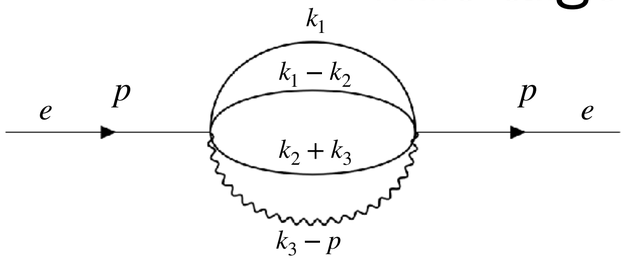
$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

sector: 15

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

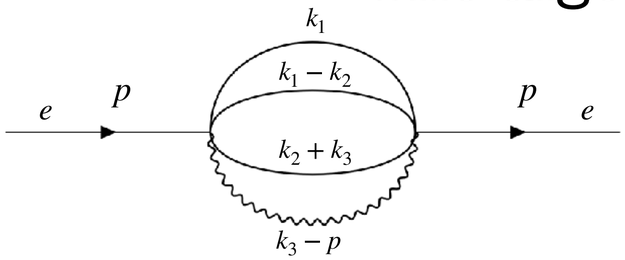
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$\frac{\Psi_{1,0,0,0,0,0}[1],}{|\mu| = 1,\ r = 2,\ o = 2.}$$

$$\frac{\Psi_{0,1,0,0,0,0}[1],}{|\mu| = 1,\ r = 2,\ o = 2.}$$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

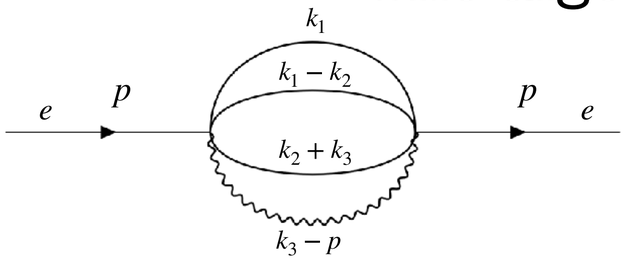
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_2 = (z_0 + z_2) = 0$$

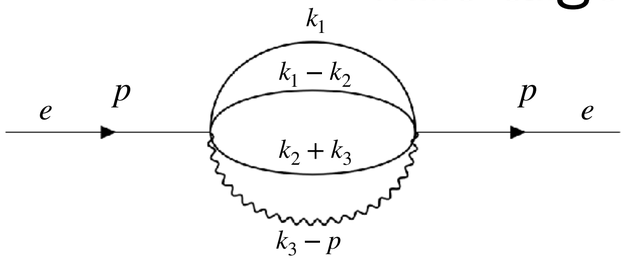
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_2 = (z_0 + z_2) = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

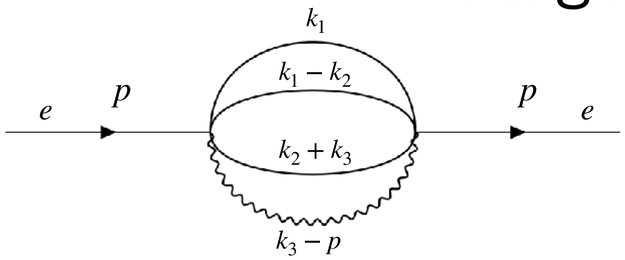
$$U_m(z_0, z, x)\Big|_{z_2+z_0\to 0} = z_0^{3\varepsilon} z_1^{\frac{1}{2}} (4xz_0 - z_1)^{-\frac{1}{2}-\varepsilon} \left[(z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2)\right]^{-\frac{1}{2}-\varepsilon}.$$

$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_2 = (z_0 + z_2) = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2+z_0\to 0} = z_0^{3\varepsilon} z_1^{\frac{1}{2}} (4xz_0 - z_1)^{-\frac{1}{2}-\varepsilon} \left[(z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2)\right]^{-\frac{1}{2}-\varepsilon}.$$

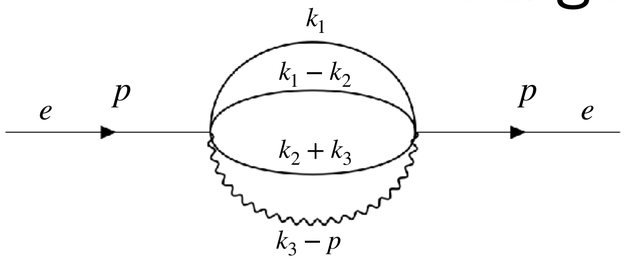
Running Laporta we get three dimensions with their pre-images in the full system:

$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

The algorithm



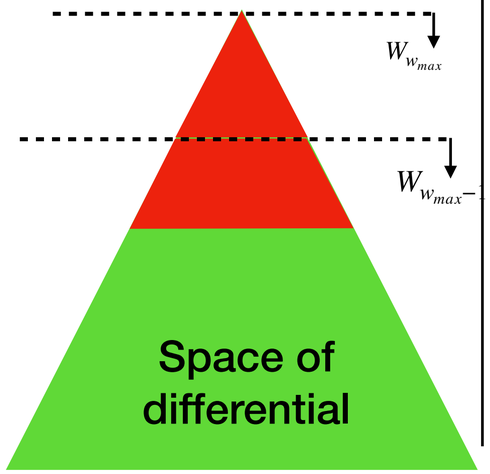
$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_2 = (z_0 + z_2) = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2+z_0\to 0} = z_0^{3\varepsilon} z_1^{\frac{1}{2}} (4xz_0 - z_1)^{-\frac{1}{2}-\varepsilon} \left[(z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2)\right]^{-\frac{1}{2}-\varepsilon}.$$

Running Laporta we get three dimensions with their pre-images in the full system:

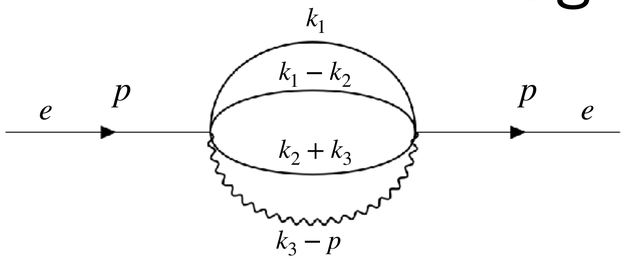


$$\Psi_{1,0,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

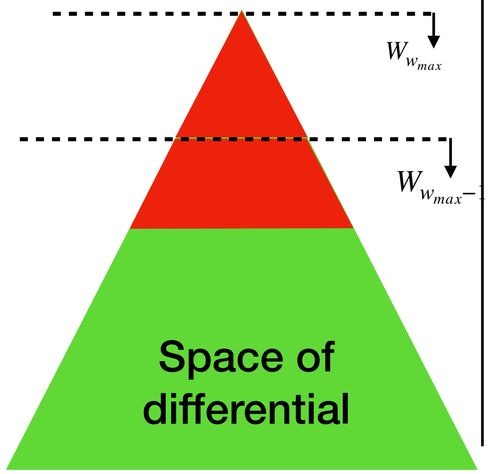
$$P_2 = (z_0 + z_2) = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2+z_0\to 0} = z_0^{3\varepsilon} z_1^{\frac{1}{2}} (4xz_0 - z_1)^{-\frac{1}{2}-\varepsilon} \left[(z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2)\right]^{-\frac{1}{2}-\varepsilon}.$$

Running Laporta we get three dimensions with their pre-images in the full system:

$$\begin{cases} \chi_1 &= 1dz_1, \\ \chi_2 &= \frac{z_1}{z_0}dz_1, \\ \chi_3 &= \frac{z_0}{(4xz_0 + z_1)}dz_1\,. \end{cases} \longrightarrow \begin{cases} \Phi_1 &= \frac{1}{P_2}\eta, \\ \Phi_2 &= \frac{z_1}{z_0 P_2}\eta, \\ \Phi_3 &= \frac{z_0}{(4xz_0 + z_1)P_2}\eta\,. \end{cases}$$

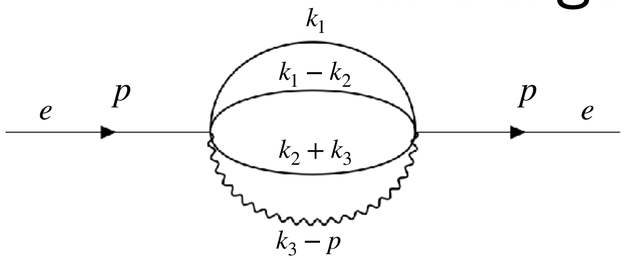


$\Psi_{1,0,0,0,0,0,0}[1]$, $|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0,0}[1]$, $|\mu| = 1,\ r = 2,\ o = 2.$

Electron—Self energy

The algorithm



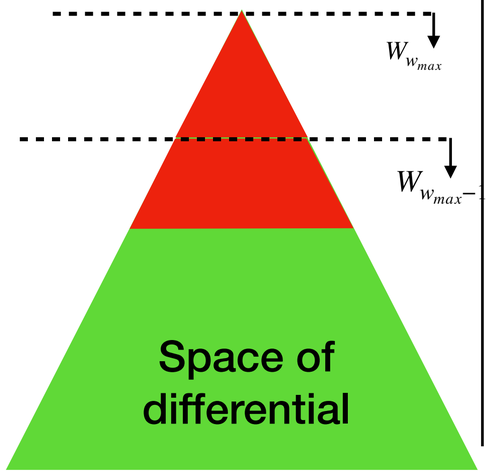
$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$P_2 = (z_0 + z_2) = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2+z_0\to 0} = z_0^{3\varepsilon} z_1^{\frac{1}{2}} (4xz_0 - z_1)^{-\frac{1}{2}-\varepsilon} \left[(z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2)\right]^{-\frac{1}{2}-\varepsilon}.$$

Running Laporta we get three dimensions with their pre-images in the full system:

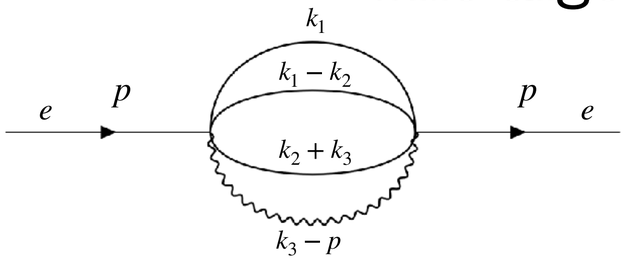


$$\Psi_{1,0,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

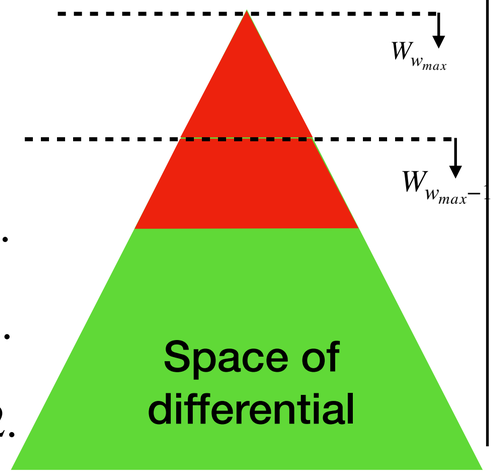
$$P_2 = (z_0 + z_2) = 0$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$U_m(z_0, z, x)\Big|_{z_2+z_0\to 0} = z_0^{3\varepsilon} z_1^{\frac{1}{2}} (4xz_0 - z_1)^{-\frac{1}{2}-\varepsilon} \left[(z_2^2 + 2(z_1 - xz_0)z_2 + (z_1 + xz_0)^2)\right]^{-\frac{1}{2}-\varepsilon}.$$

Running Laporta we get three dimensions with their pre-images in the full system:

$$\begin{cases} \Phi_1 = \frac{1}{P_2}\eta, & |\mu| = 1,\ r = 1,\ o = 1. \\ \Phi_2 = \frac{z_1}{z_0 P_2}\eta, & |\mu| = 2,\ r = 2,\ o = 2. \\ \Phi_3 = \frac{z_0}{(4xz_0 + z_1)P_2}\eta\,. & |\mu| = 2,\ r = 1,\ o = 2. \end{cases}$$

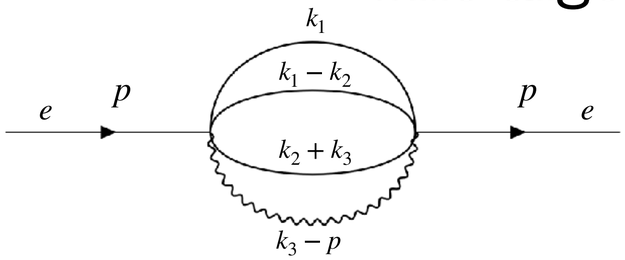


$\Psi_{1,0,0,0,0,0}[1]$, $|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1]$, $|\mu| = 1,\ r = 2,\ o = 2.$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

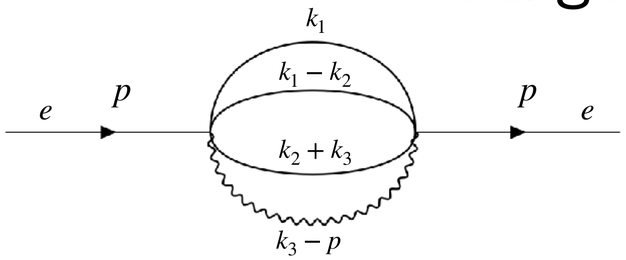
$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$$\Psi_{1,0,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

$$\Psi_{0,1,0,0,0,0}[1], \quad |\mu| = 1, \ r = 2, \ o = 2.$$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

$\Psi_{1,0,0,0,0,0}[1]$, $|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1]$, $|\mu| = 1,\ r = 2,\ o = 2.$

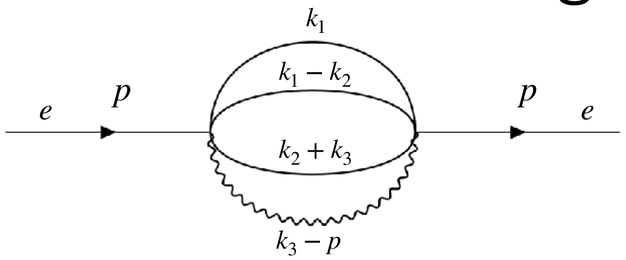
$\Psi_{0,0,1,0,0,0}[1]$, $|\mu| = 1,\ r = 1,\ o = 1.$

$\Psi_{1,0,1,0,0,0}[z_1]$, $|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,1,0}[z_0]\,.$ $|\mu| = 1,\ r = 1,\ o = 2.$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$$\begin{cases} P_0 = z_0, \\ P_1 = z_2, \\ P_2 = (z_0 + z_2)\,. \end{cases}$$

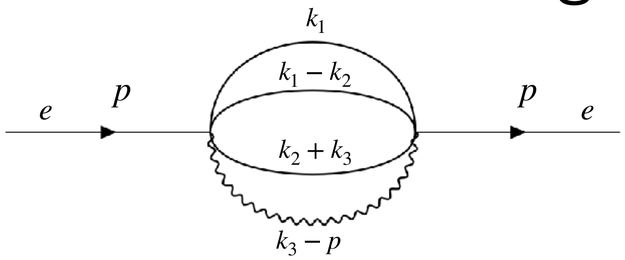
$\Psi_{1,0,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0]\,.$
$|\mu| = 1,\ r = 1,\ o = 2.$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$\Psi_{1,0,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

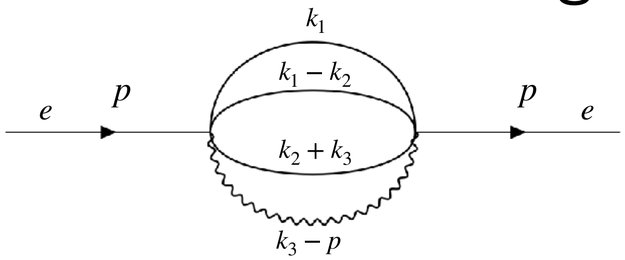
$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$

$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 1,\ o = 2.$

Electron—Self energy

The algorithm



$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

Full system

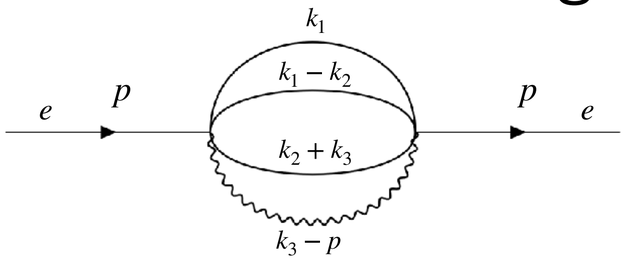
$\Psi_{1,0,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 1,\ o = 2.$

Electron—Self energy

The algorithm



Full system

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

$\Psi_{1,0,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

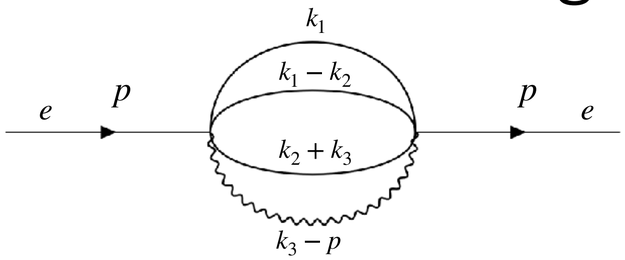
$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$

$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 1,\ o = 2.$

Electron—Self energy

The algorithm



Full system

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

Running Laporta on the full system we get one dimension with its pre-image in the full system:

$$(a, w, o, |\mu|, \dots)$$

$\Psi_{1,0,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

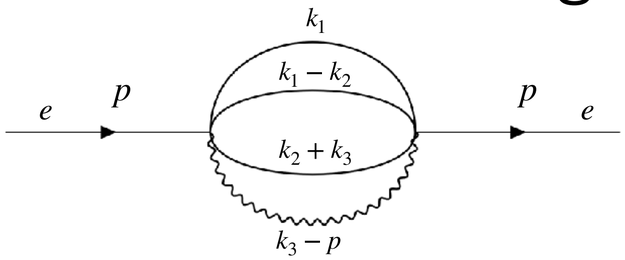
$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$

$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 1,\ o = 2.$

Electron—Self energy

The algorithm



Full system

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

Running Laporta on the full system we get one dimension with its pre-image in the full system:

$$(a, w, o, |\mu|, \dots)$$

$\Psi_{1,0,0,0,0,0}[1], \quad a = -2$
$|\mu| = 1, \ r = 2, \ o = 2.$

$\Psi_{0,1,0,0,0,0}[1], \quad a = -2$
$|\mu| = 1, \ r = 2, \ o = 2.$

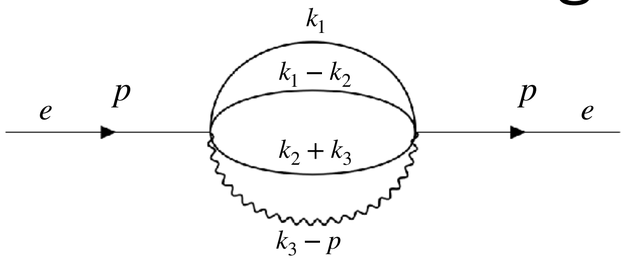
$\Psi_{0,0,1,0,0,0}[1], \quad a = -1$
$|\mu| = 1, \ r = 1, \ o = 1.$

$\Psi_{1,0,1,0,0,0}[z_1], \quad a = -2$
$|\mu| = 1, \ r = 2, \ o = 2.$

$\Psi_{0,0,1,0,1,0}[z_0] \cdot \quad a = -1$
$|\mu| = 1, \ r = 1, \ o = 2.$

Electron—Self energy

The algorithm



Full system

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

Running Laporta on the full system we get one dimension with its pre-image in the full system:

$$(a, w, o, |\mu|, \dots)$$

$$\Phi = \frac{1}{(4xz_0 + z1)}\,\eta \longrightarrow \Psi_{0,0,0,0,1,0}[1],$$

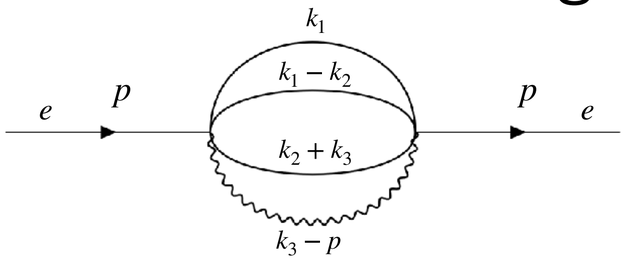
$\Psi_{1,0,0,0,0,0}[1], \quad a = -2$
$|\mu| = 1, \ r = 2, \ o = 2.$

$\Psi_{0,1,0,0,0,0}[1], \quad a = -2$
$|\mu| = 1, \ r = 2, \ o = 2.$

$\Psi_{0,0,1,0,0,0}[1], \quad a = -1$
$|\mu| = 1, \ r = 1, \ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1], \quad a = -2$
$|\mu| = 1, \ r = 2, \ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0] \cdot \quad a = -1$
$|\mu| = 1, \ r = 1, \ o = 2.$

Electron—Self energy

The algorithm



Full system

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

Running Laporta on the full system we get one dimension with its pre-image in the full system:

$$(a, w, o, |\mu|, \dots)$$

$$\Phi = \frac{1}{(4xz_0 + z1)}\,\eta \longrightarrow \Psi_{0,0,0,0,1,0}[1],$$
$$|\mu| = 1,\ r = 0,\ o = 1.$$

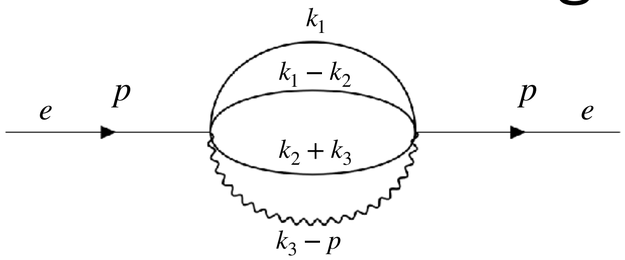
$\Psi_{1,0,0,0,0,0}[1],\ a = -2$
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,1,0,0,0,0}[1],\ a = -2$
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,0,0}[1],\ a = -1$
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1],\ a = -2$
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0] \cdot a = -1$
$|\mu| = 1,\ r = 1,\ o = 2.$

Electron—Self energy

The algorithm



Full system

$$U_m(z_0, z, x) = \prod_{i \in I_{odd}} P_i^{-\frac{1}{2}+\frac{1}{2}b_j\varepsilon}(z_0, z, x) \prod_{j \in I_{even}} P_j^{\frac{1}{2}b_j\varepsilon}(z_0, z, x),$$

Running Laporta on the full system we get one dimension with its pre-image in the full system:

$$(a, w, o, |\mu|, \dots)$$

$$\Phi = \frac{1}{(4xz_0 + z1)}\,\eta \longrightarrow \Psi_{0,0,0,0,1,0}[1],$$
$$|\mu| = 1,\ r = 0,\ o = 1.$$

$\Psi_{1,0,0,0,0,0}[1],\ a = -2$
$|\mu| = 1,\ r = 2,\ o = 2.$

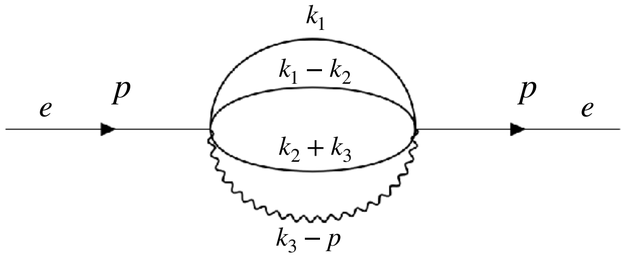
$\Psi_{0,1,0,0,0,0}[1],\ a = -2$
$|\mu| = 1,\ r = 2,\ o = 2.$

$\Psi_{0,0,1,0,0,0}[1],\ a = -1$
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1],\ a = -2$
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0] \cdot\ a = -1$
$|\mu| = 1,\ r = 1,\ o = 2.$

$\Psi_{0,0,0,0,1,0}[1]$
$|\mu| = 1,\ r = 0,\ o = 1.$

Electron—Self energy

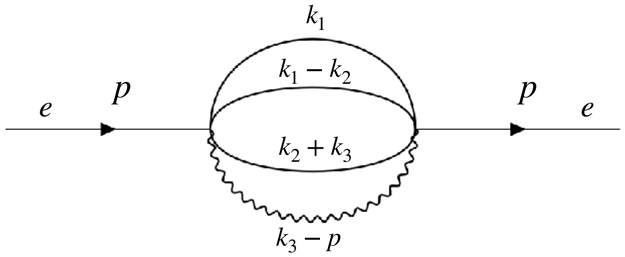
Feynman Integrals



$$I_{\nu_1,\nu_2,\nu_3,\nu_4,0,0,0,0,0}$$

Electron—Self energy

Feynman Integrals



$$I_{\nu_1,\nu_2,\nu_3,\nu_4,0,0,0,0,0,0}$$

Form space

$$\Psi_{1,0,0,0,0,0}[1],$$

$$\Psi_{0,1,0,0,0,0}[1],$$

$$\Psi_{0,0,1,0,0,0}[1],$$

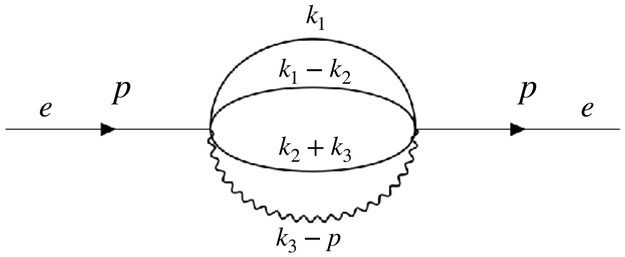
$$\Psi_{1,0,1,0,0,0}[z_1],$$

$$\Psi_{0,0,1,0,1,0}[z_0]\,.$$

$$\Psi_{0,0,0,0,1,0}[1],$$

Electron—Self energy

Feynman Integrals



$$I_{\nu_1,\nu_2,\nu_3,\nu_4,0,0,0,0,0}$$

Form space

Feynman integrals

$$\Psi_{1,0,0,0,0,0}[1],$$

$$\Psi_{0,1,0,0,0,0}[1],$$

$$\Psi_{0,0,1,0,0,0}[1],$$

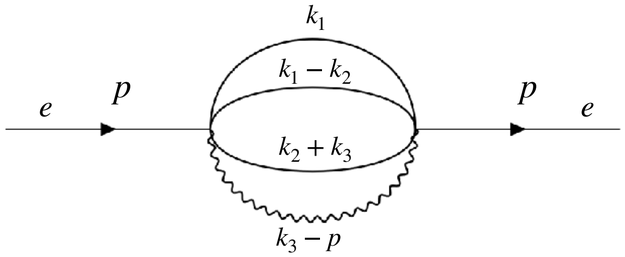
$$\Psi_{1,0,1,0,0,0}[z_1],$$

$$\Psi_{0,0,1,0,1,0}[z_0]\,.$$

$$\Psi_{0,0,0,0,1,0}[1],$$

Electron—Self energy

Feynman Integrals



$$I_{\nu_1,\nu_2,\nu_3,\nu_4,0,0,0,0,0,0}$$

Form space

Feynman integrals

$$\Psi_{1,0,0,0,0,0}[1],$$

$$\Psi_{0,1,0,0,0,0}[1],$$

Symmetries

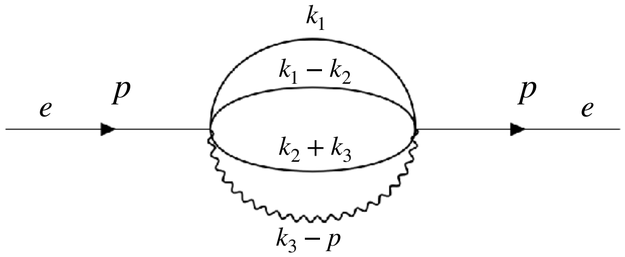
$$\Rightarrow$$

$$\Psi_{0,0,1,0,0,0}[1],$$
$$\Psi_{1,0,1,0,0,0}[z_1],$$
$$\Psi_{0,0,1,0,1,0}[z_0]\,.$$

$$\Psi_{0,0,0,0,1,0}[1],$$

Electron—Self energy

Feynman Integrals



$$I_{\nu_1,\nu_2,\nu_3,\nu_4,0,0,0,0,0}$$

Form space

$$\Psi_{1,0,0,0,0,0}[1],$$

$$\Psi_{0,1,0,0,0,0}[1],$$

$$\Psi_{0,0,0,0,1,0}[1],$$

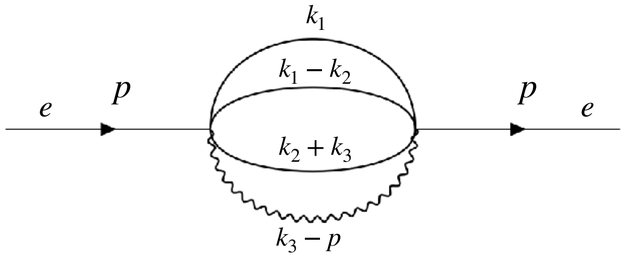
Symmetries ⇒

$$\Psi_{0,0,1,0,0,0}[1],$$
$$\Psi_{1,0,1,0,0,0}[z_1],$$
$$\Psi_{0,0,1,0,1,0}[z_0]\,.$$

Feynman integrals ⇒

Electron—Self energy

Feynman Integrals



$$I_{\nu_1,\nu_2,\nu_3,\nu_4,0,0,0,0,0,0}$$

Form space

$$\Psi_{1,0,0,0,0,0}[1],$$

$$\Psi_{0,1,0,0,0,0}[1],$$

Symmetries

$$\Rightarrow$$

$$\Psi_{0,0,1,0,0,0}[1],$$
$$\Psi_{1,0,1,0,0,0}[z_1],$$
$$\Psi_{0,0,1,0,1,0}[z_0]\,.$$

Feynman integrals

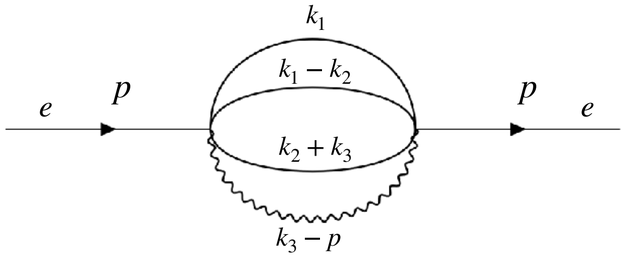
$$\Rightarrow$$

$$-2\varepsilon^3 I_{1,1,1,1,0,0,0,0,0,0},$$
$$\varepsilon^2 I_{1,1,2,1,0,0,0,0,0,0},$$
$$-8\varepsilon^3 I_{1,1,1,-1,0,0,0,0,0,0}\,.$$

$$\Psi_{0,0,0,0,1,0}[1],$$

Electron–Self energy

Feynman Integrals



$$I_{\nu_1,\nu_2,\nu_3,\nu_4,0,0,0,0,0}$$

Form space

$$\Psi_{1,0,0,0,0,0}[1],$$

$$\Psi_{0,1,0,0,0,0}[1],$$

Symmetries

$$\Rightarrow$$

$$\Psi_{0,0,1,0,0,0}[1],$$
$$\Psi_{1,0,1,0,0,0}[z_1],$$
$$\Psi_{0,0,1,0,1,0}[z_0]\,.$$

Feynman integrals

$$\Rightarrow$$

$$-2\varepsilon^3 I_{1,1,1,1,0,0,0,0,0},$$
$$\varepsilon^2 I_{1,1,2,1,0,0,0,0,0},$$
$$-8\varepsilon^3 I_{1,1,1,-1,0,0,0,0,0}\,.$$

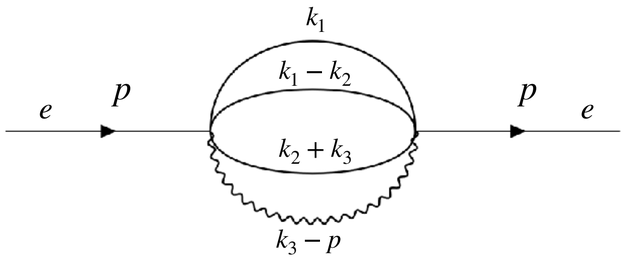
$$\Psi_{0,0,0,0,1,0}[1],$$

Non reducible supersector

$$I_{1,1,1,1,0,1,0,0,0}$$

Electron—Self energy

DEQs

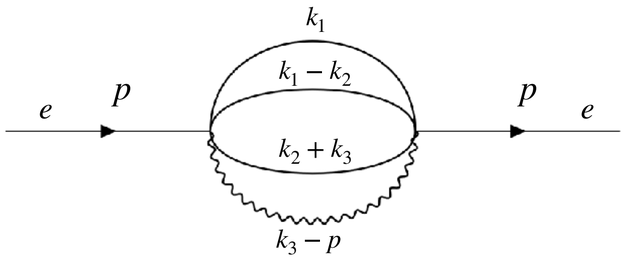


$$A_{i,j}(\varepsilon, x) = \sum_{k=-(|\mu_i|-|\mu_i|)}^{1} \varepsilon^k A_{i,j}^{(k)}(x)$$

$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 2,\ o = 2.$

Electron—Self energy

DEQs



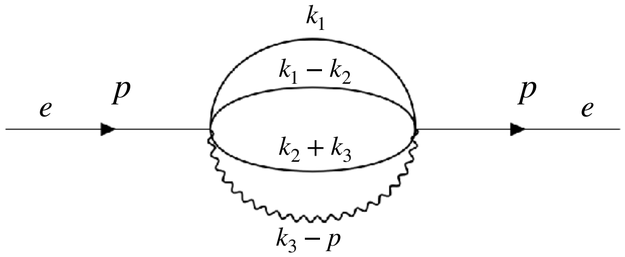
$$A_{i,j}(\varepsilon, x) = \sum_{k=-(|\mu_i|-|\mu_i|)}^{1} \varepsilon^k A_{i,j}^{(k)}(x)$$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2\varepsilon^3 I_{1,1,1,1,0,0,0,0,0}, \varepsilon^2 I_{1,1,2,1,0,0,0,0,0}, -8\varepsilon^3 I_{1,1,1,1,0,0,0,0,0}\}$$

$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 2,\ o = 2.$

Electron—Self energy

DEQs



$$A_{i,j}(\varepsilon, x) = \sum_{k=-(|\mu_i|-|\mu_i|)}^{1} \varepsilon^k A_{i,j}^{(k)}(x)$$

$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 2,\ o = 2.$

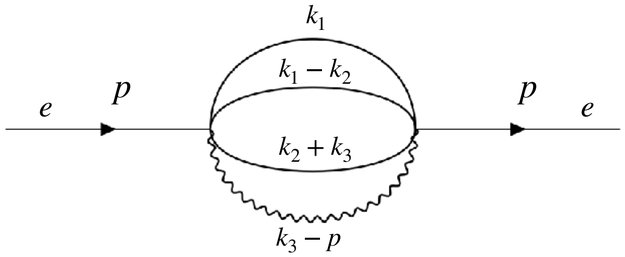
$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2\varepsilon^3 I_{1,1,1,1,0,0,0,0,0}, \varepsilon^2 I_{1,1,2,1,0,0,0,0,0}, -8\varepsilon^3 I_{1,1,1,1,0,0,0,0,0}\}$$

Laurent series in ε

$$dJ = \begin{pmatrix} 0 & 6\varepsilon & 0 \\ \frac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \frac{\varepsilon-1}{x} + \frac{36\varepsilon+9}{1-9x} + \frac{4\varepsilon+1}{1-x} & \frac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\frac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\frac{4\varepsilon}{x} \end{pmatrix} \cdot J$$

Electron—Self energy

DEQs



$$A_{i,j}(\varepsilon, x) = \sum_{k=-(|\mu_i|-|\mu_i|)}^{1} \varepsilon^k A_{i,j}^{(k)}(x)$$

$\Psi_{0,0,1,0,0,0}[1]$,
$|\mu| = 1,\ r = 1,\ o = 1.$
$\Psi_{1,0,1,0,0,0}[z_1]$,
$|\mu| = 1,\ r = 2,\ o = 2.$
$\Psi_{0,0,1,0,1,0}[z_0]$.
$|\mu| = 1,\ r = 2,\ o = 2.$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2\varepsilon^3 I_{1,1,1,1,0,0,0,0,0}, \varepsilon^2 I_{1,1,2,1,0,0,0,0,0}, -8\varepsilon^3 I_{1,1,1,1,0,0,0,0,0}\}$$

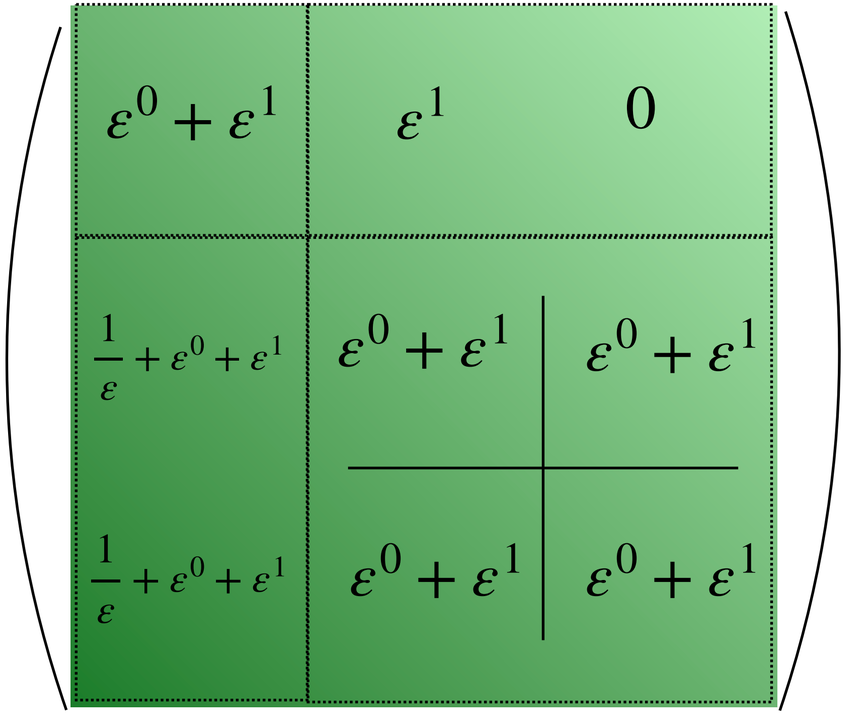
Laurent series in ε

$$dJ = \left(\begin{array}{c|cc} 0 & 6\varepsilon & 0 \\ \hline \dfrac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \dfrac{\varepsilon-1}{x} + \dfrac{36\varepsilon+9}{1-9x} + \dfrac{4\varepsilon+1}{1-x} & \dfrac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\dfrac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\dfrac{4\varepsilon}{x} \end{array}\right) \cdot J$$

Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad$ $\vec{\mu} = \{1,2,2\}$

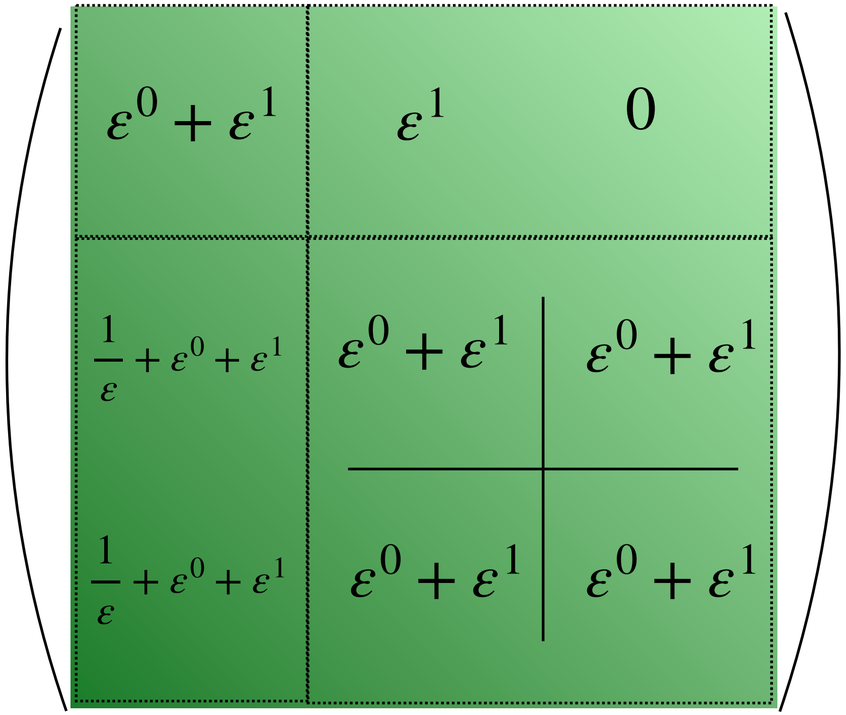
$n = 2$ $\qquad$ $A(x) = \sum_{k=-1}^{1} B^k(x)$



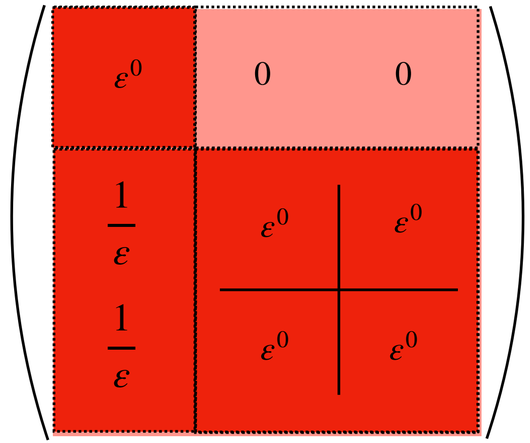
Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$



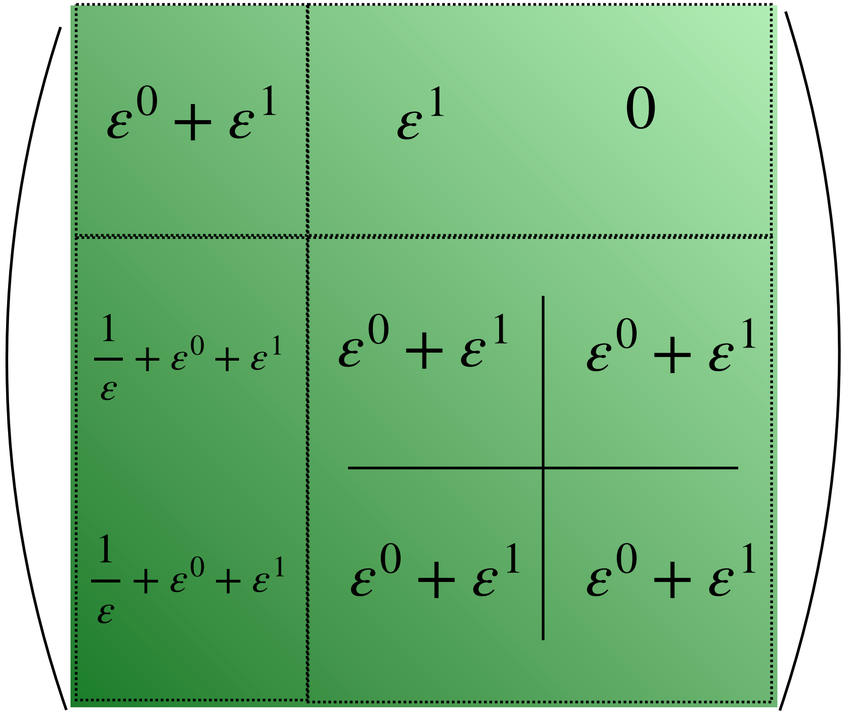
$B^{-1}(x) =$



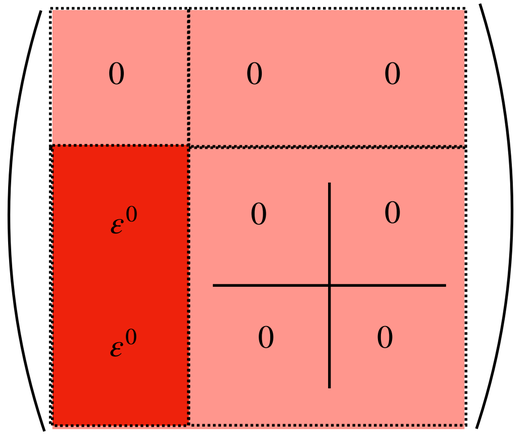
Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

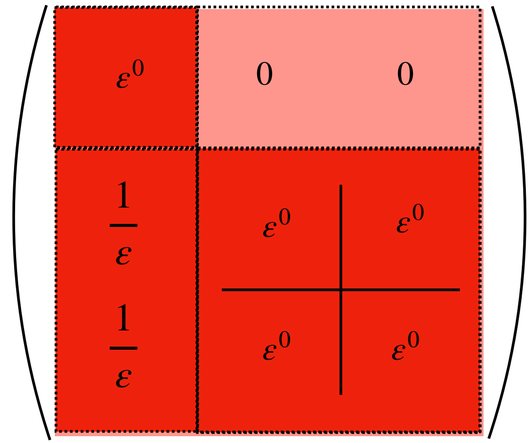
$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$



$B^0(x) =$



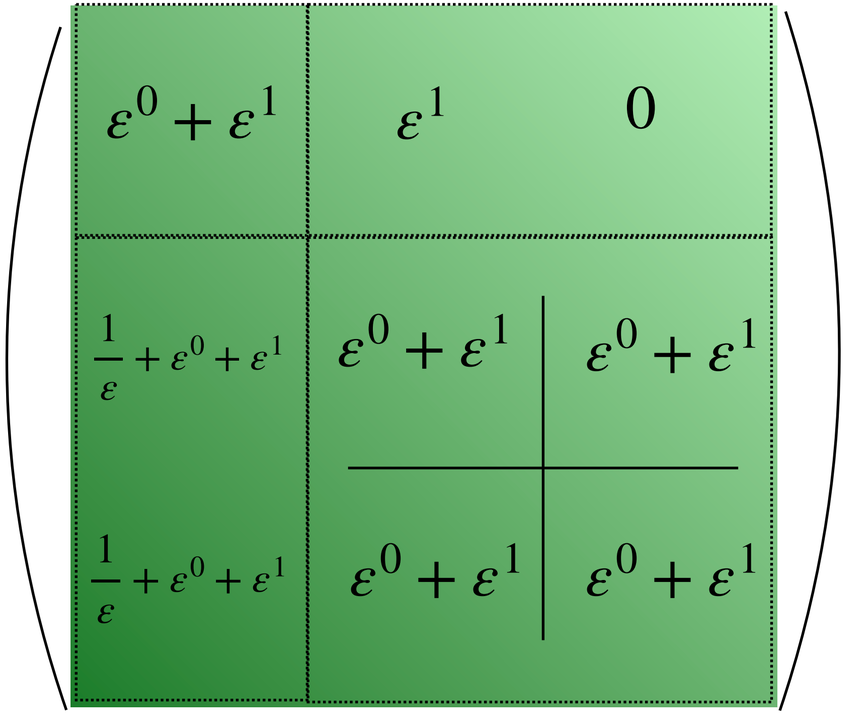
$B^{-1}(x) =$



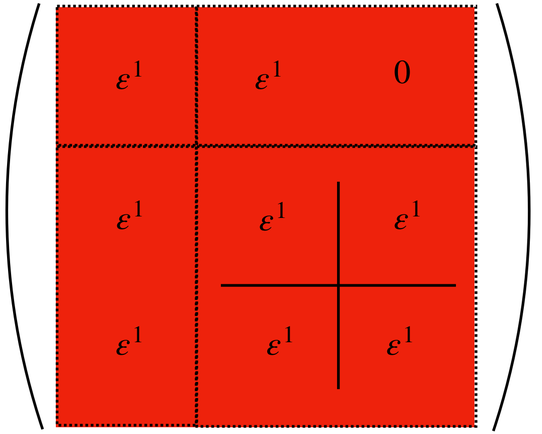
Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

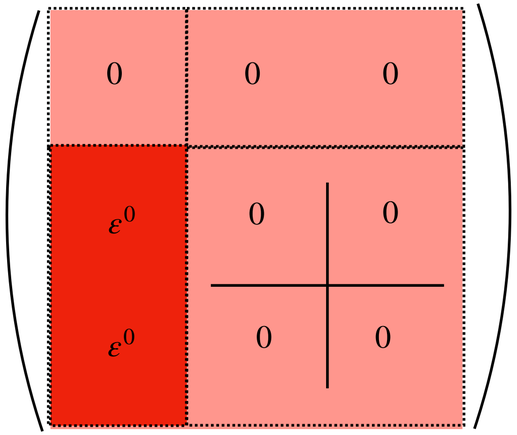
$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$



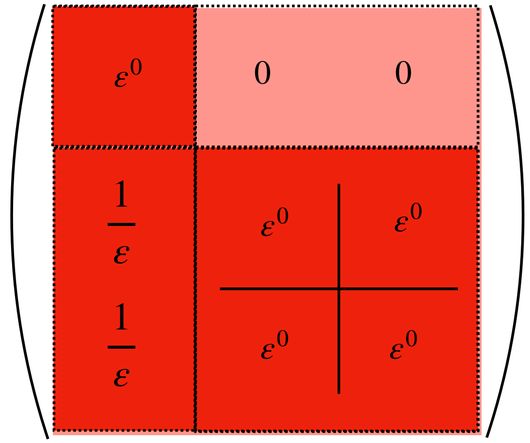
$B^1(x) =$



$B^0(x) =$



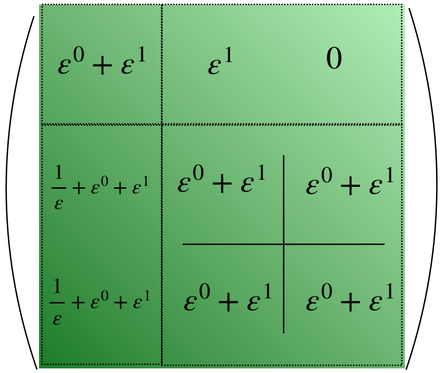
$B^{-1}(x) =$



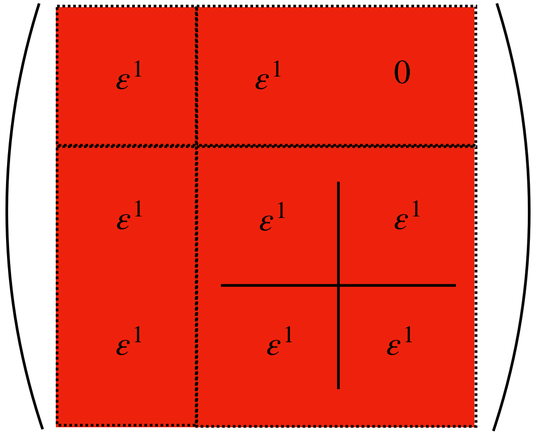
Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

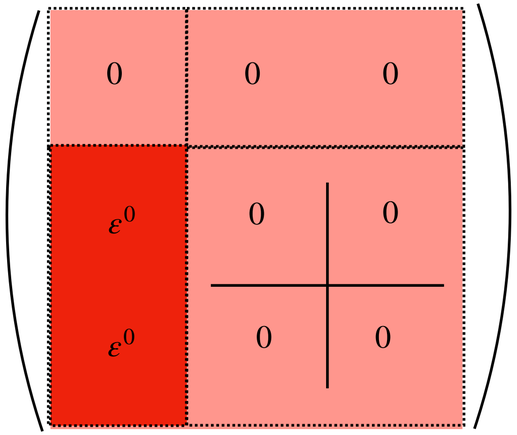
$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$



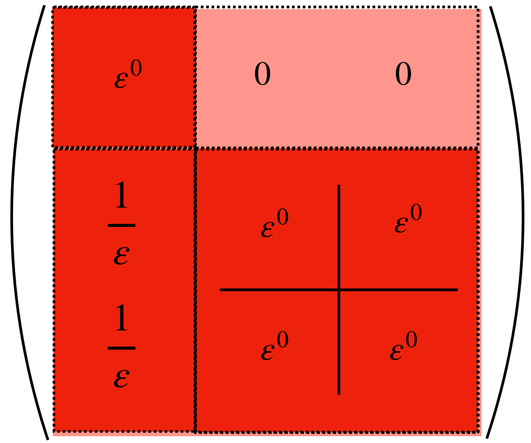
$B^1(x) =$



$B^0(x) =$



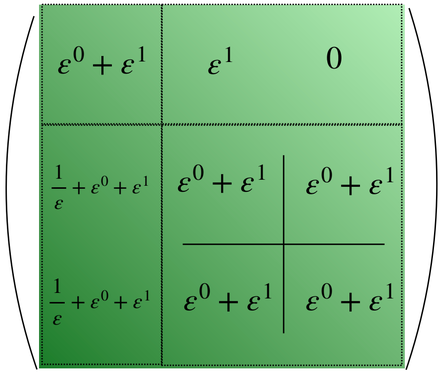
$B^{-1}(x) =$



Are we done ? NO!!

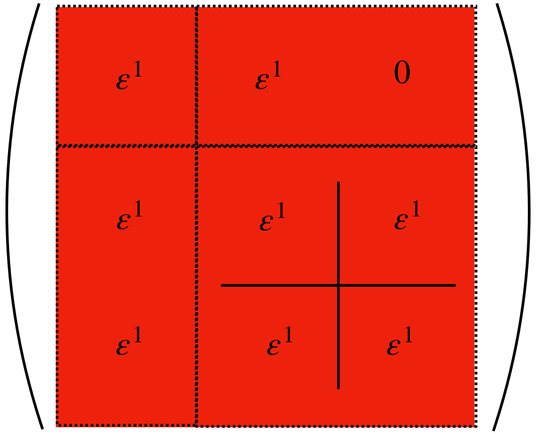
$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$

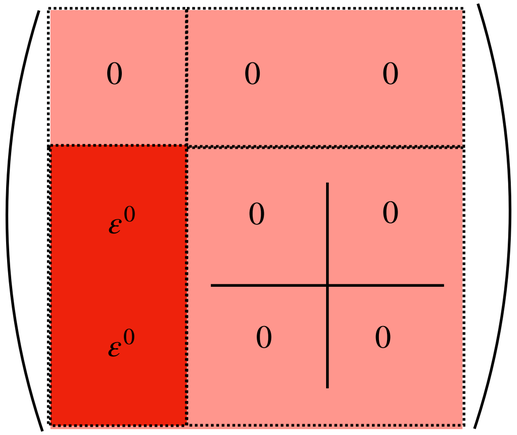


$$R^{-1}(x) = \begin{pmatrix} \varepsilon^0 R_{11}^{-1}(x) & 0 & 0 \\ \frac{1}{\varepsilon} R_{31}^{-1}(x) & \varepsilon^0 R_{22}^{-1}(x) & \varepsilon^0 R_{23}^{-1}(x) \\ \frac{1}{\varepsilon} R_{41}^{-1}(x) & \varepsilon^0 R_{32}^{-1}(x) & \varepsilon^0 R_{33}^{-1}(x) \end{pmatrix}$$

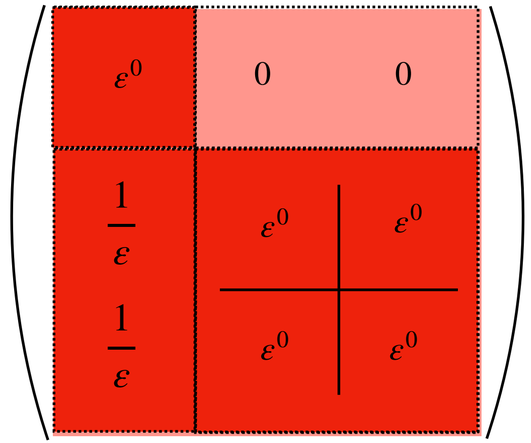
$B^1(x) =$



$B^0(x) =$



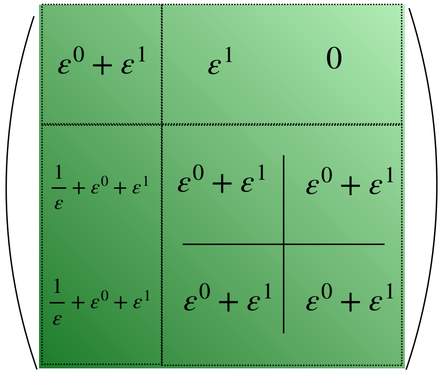
$B^{-1}(x) =$



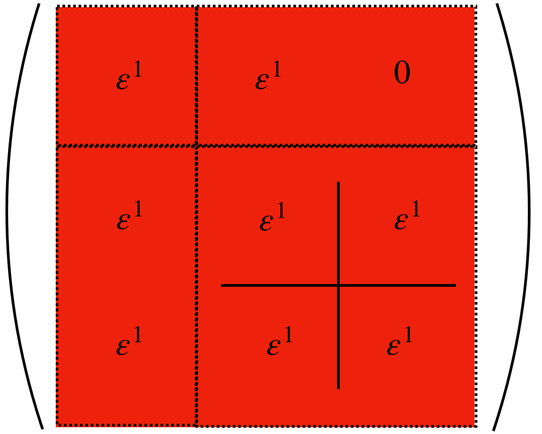
Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

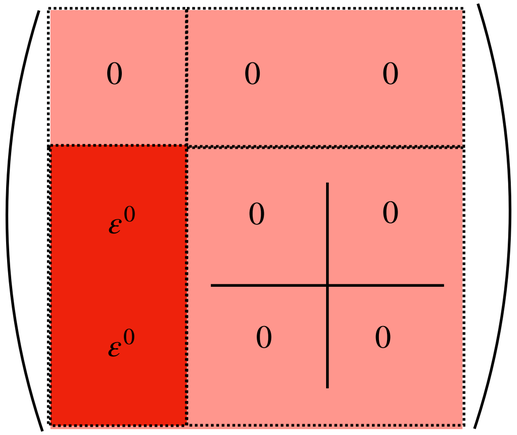
$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$



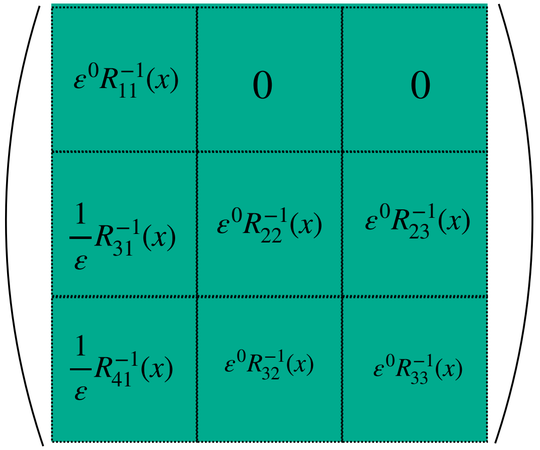
$B^1(x) =$



$B^0(x) =$



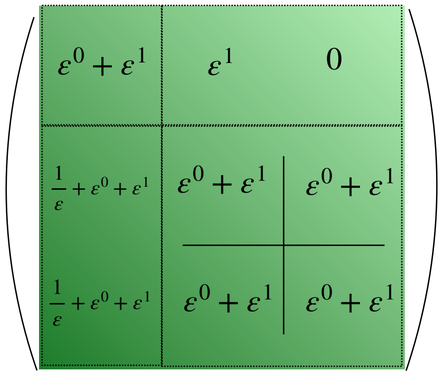
$R^{-1}(x) =$



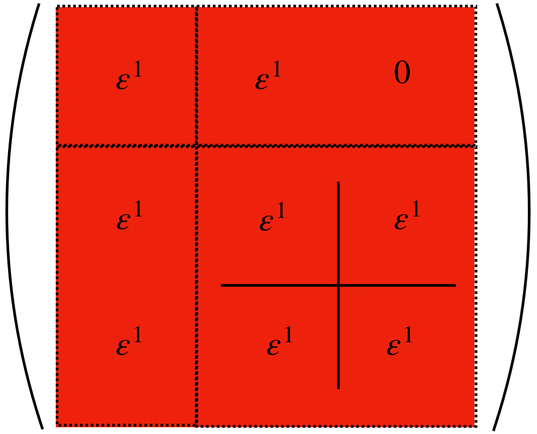
Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$

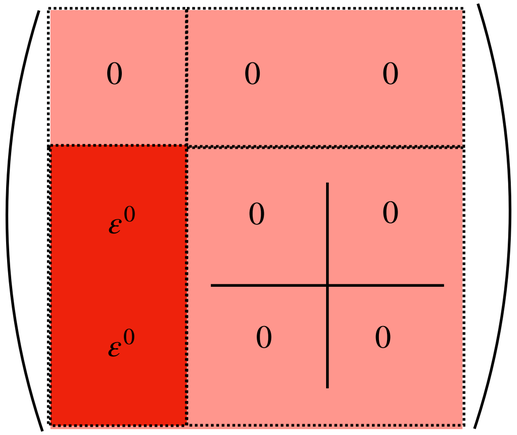


$B^1(x) =$

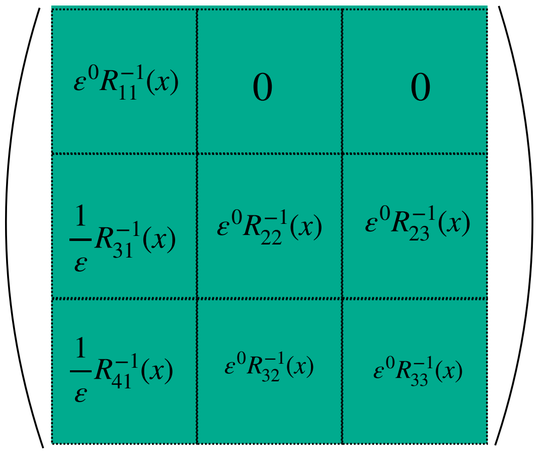


$$R^0(x) = \begin{pmatrix} 1 & 0 & 0 \\ \varepsilon^0 R_{21}^0(x) & 1 & 0 \\ \varepsilon^0 R_{31}^0(x) & 0 & 1 \end{pmatrix}$$

$B^0(x) =$



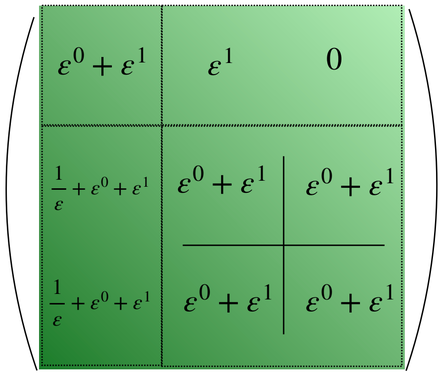
$R^{-1}(x) =$



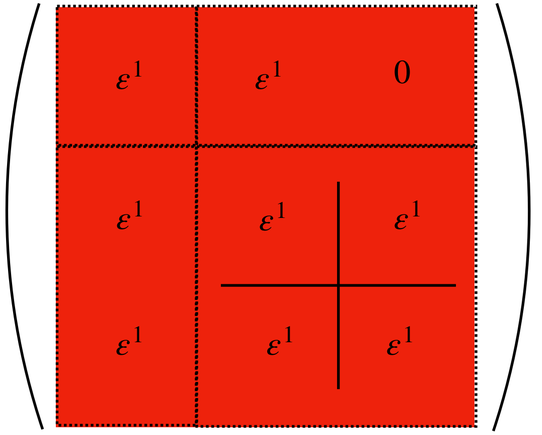
Are we done ? NO!!

$J = (J_1, J_2, J_3)$ $\qquad \vec{\mu} = \{1,2,2\}$

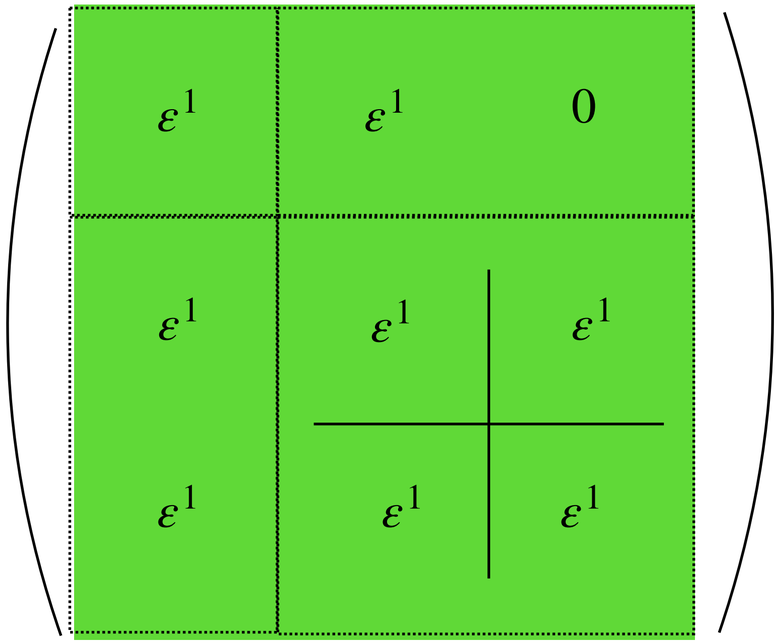
$n = 2$ $\qquad A(x) = \sum_{k=-1}^{1} B^k(x)$



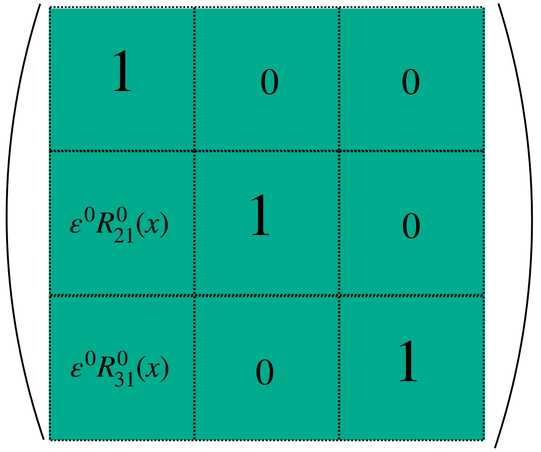
$B^1(x) =$



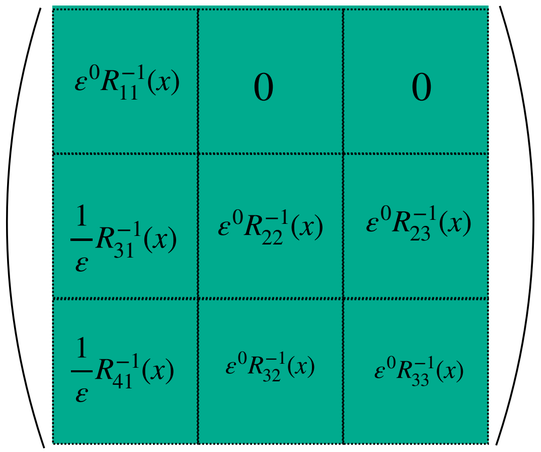
$\tilde{A}(x) =$



$R^0(x) =$

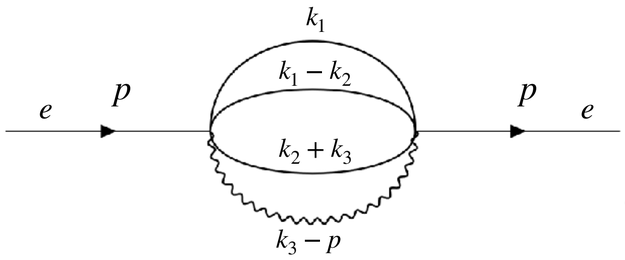


$R^{-1}(x) =$



Electron—Self energy

Rotation

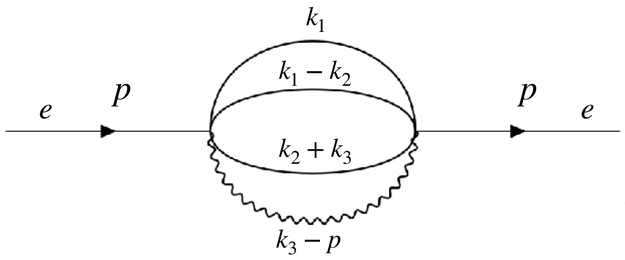


$$dJ = \begin{pmatrix} 0 & 6\varepsilon & 0 \\ \frac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \frac{\varepsilon-1}{x} + \frac{36\varepsilon+9}{1-9x} + \frac{4\varepsilon+1}{1-x} & \frac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\frac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\frac{4\varepsilon}{x} \end{pmatrix} \cdot J\,.$$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2I_{1,1,1,1,0,0,0,0,0}, \frac{1}{\varepsilon} I_{1,1,2,1,0,0,0,0,0}, -8I_{1,1,1,1,0,0,0,0,0}\}\,.$$

Electron—Self energy

Rotation



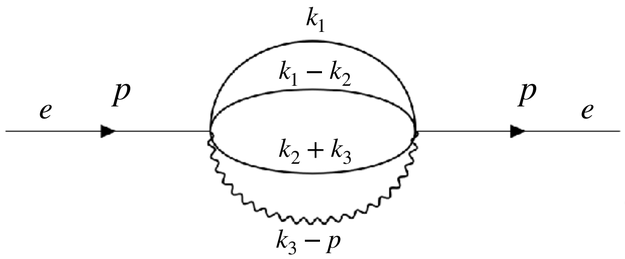
$$dJ = \begin{pmatrix} 0 & 6\varepsilon & 0 \\ \frac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \frac{\varepsilon-1}{x} + \frac{36\varepsilon+9}{1-9x} + \frac{4\varepsilon+1}{1-x} & \frac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\frac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\frac{4\varepsilon}{x} \end{pmatrix} \cdot J .$$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2I_{1,1,1,1,0,0,0,0,0}, \frac{1}{\varepsilon} I_{1,1,2,1,0,0,0,0,0}, -8I_{1,1,1,1,0,0,0,0,0}\} .$$

$$R^{-1}(x) = \begin{pmatrix} \psi(x) & 0 & 0 \\ \frac{\psi'(x)}{6\varepsilon} & \frac{1}{(9x^3 - 10x^2 + x)\psi(x)} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Electron—Self energy

Rotation



$$dJ = \begin{pmatrix} 0 & 6\varepsilon & 0 \\ \frac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \frac{\varepsilon-1}{x} + \frac{36\varepsilon+9}{1-9x} + \frac{4\varepsilon+1}{1-x} & \frac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\frac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\frac{4\varepsilon}{x} \end{pmatrix} \cdot J .$$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2I_{1,1,1,1,0,0,0,0,0}, \frac{1}{\varepsilon} I_{1,1,2,1,0,0,0,0,0}, -8I_{1,1,1,1,0,0,0,0,0}\} .$$

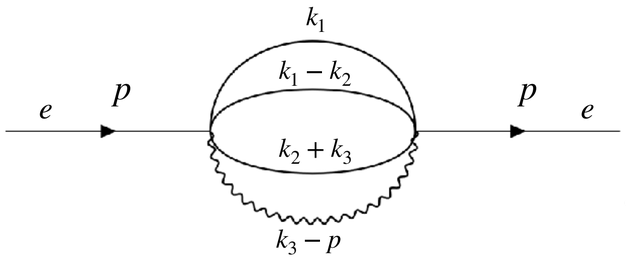
ψ : is the holomorphic period

$L^{2,0}\psi = 0$

$$R^{-1}(x) = \begin{pmatrix} \psi(x) & 0 & 0 \\ \frac{\psi'(x)}{6\varepsilon} & \frac{1}{(9x^3 - 10x^2 + x)\psi(x)} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Electron—Self energy

Rotation



$$dJ = \begin{pmatrix} 0 & 6\varepsilon & 0 \\ \frac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \frac{\varepsilon-1}{x} + \frac{36\varepsilon+9}{1-9x} + \frac{4\varepsilon+1}{1-x} & \frac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\frac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\frac{4\varepsilon}{x} \end{pmatrix} \cdot J\,.$$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2I_{1,1,1,1,0,0,0,0,0}, \frac{1}{\varepsilon} I_{1,1,2,1,0,0,0,0,0}, -8I_{1,1,1,1,0,0,0,0,0}\}\,.$$

ψ : is the holomorphic period

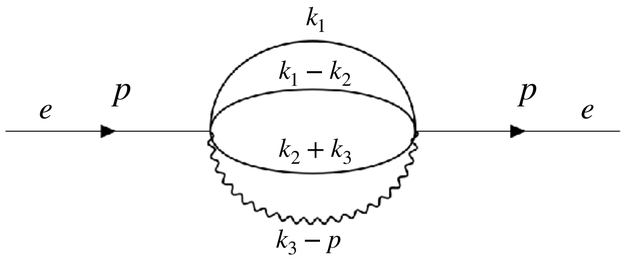
$L^{2,0}\psi = 0$

$$R^{-1}(x) = \begin{pmatrix} \psi(x) & 0 & 0 \\ \frac{\psi'(x)}{6\varepsilon} & \frac{1}{(9x^3 - 10x^2 + x)\psi(x)} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R^{0}(x) = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{12}\left(63x^2 - 30x - 1\right)\psi(x)^2 & 1 & 0 \\ -\frac{4}{3}(3x+1)\psi(x) & 0 & 1 \end{pmatrix},$$

Electron—Self energy

Rotation



$$dJ = \begin{pmatrix} 0 & 6\varepsilon & 0 \\ \frac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \frac{\varepsilon-1}{x} + \frac{36\varepsilon+9}{1-9x} + \frac{4\varepsilon+1}{1-x} & \frac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\frac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\frac{4\varepsilon}{x} \end{pmatrix} \cdot J\,.$$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2I_{1,1,1,1,0,0,0,0,0}, \frac{1}{\varepsilon} I_{1,1,2,1,0,0,0,0,0}, -8I_{1,1,1,1,0,0,0,0,0}\}\,.$$

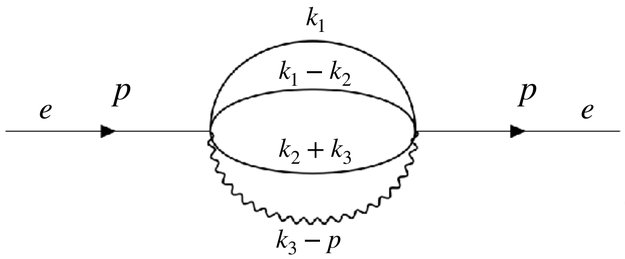
ψ : is the holomorphic period

$L^{2,0}\psi = 0$

$$R = R^{-1}(x)R^{0}(x),$$

Electron—Self energy

Rotation



$$dJ = \begin{pmatrix} 0 & 6\varepsilon & 0 \\ \frac{2\varepsilon^2 - 3(3\varepsilon+1)(4\varepsilon+1)x^2 + (18\varepsilon+5)\varepsilon x + x}{2\varepsilon(x-1)x^2(9x-1)} & \frac{\varepsilon-1}{x} + \frac{36\varepsilon+9}{1-9x} + \frac{4\varepsilon+1}{1-x} & \frac{3\varepsilon}{2x^2(9x^2-10x+1)} \\ -\frac{8\varepsilon(6x+1)}{3x} - 4 & -8\varepsilon(3x+1) & -\frac{4\varepsilon}{x} \end{pmatrix} \cdot J\,.$$

$$J = \{\Psi_{0,0,1,0,0,0}[1], \Psi_{0,0,1,0,1,0}[z_0], \Psi_{1,0,1,0,0,0}[z_1]\}$$
$$= \{-2I_{1,1,1,1,0,0,0,0,0}, \frac{1}{\varepsilon} I_{1,1,2,1,0,0,0,0,0}, -8I_{1,1,1,1,0,0,0,0,0}\}\,.$$

ψ : is the holomorphic period

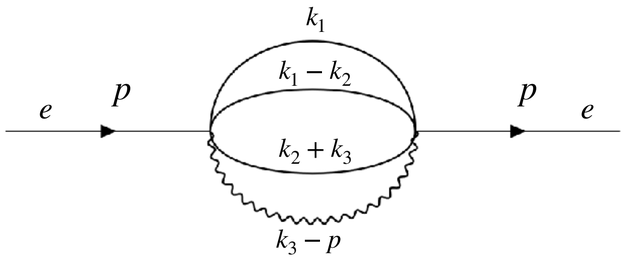
$L^{2,0}\psi = 0$

$$R = R^{-1}(x)R^0(x),$$

$$R(x) = \begin{pmatrix} \psi(x) & 0 & 0 \\ \frac{1}{12}\left(\frac{2\psi'(x)}{\varepsilon} + \frac{(-63x^2+30x+1)\psi(x)}{x(9x^2-10x+1)}\right) & \frac{1}{(9x^3-10x^2+x)\psi(x)} & 0 \\ -\frac{4}{3}(3x+1)\psi(x) & 0 & 1 \end{pmatrix},$$

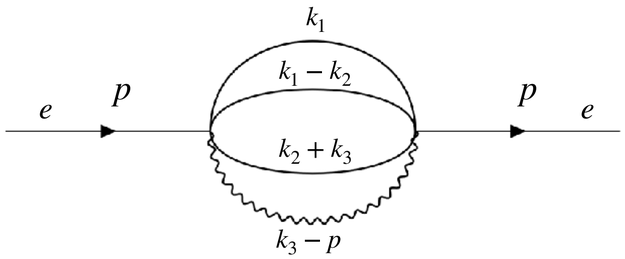
Electron—Self energy

DEQs



Electron—Self energy

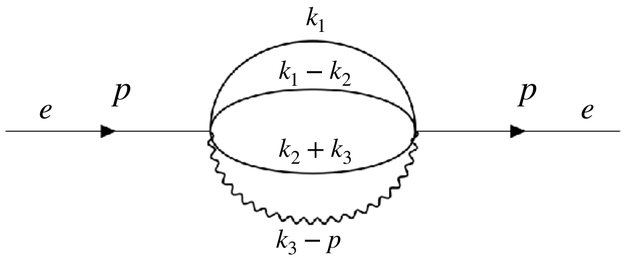
DEQs



$$d_x K = \varepsilon \begin{pmatrix} \frac{1}{2}\left(-\frac{4}{x-1}+\frac{1}{x}+\frac{36}{1-9x}\right) & \frac{6}{x(x-1)x(9x-1)\psi(x)^2} & 0 \\ \frac{\left(81x^4+1188x^3-594x^2+372x-23\right)\psi(x)^2}{24(x-1)x(9x-1)} & \frac{1}{2}\left(-\frac{4}{x-1}+\frac{1}{x}+\frac{36}{1-9x}\right) & \frac{3\psi(x)}{2x} \\ \frac{8\psi(x)}{3x} & 0 & -\frac{4}{x} \end{pmatrix} \cdot K\,.$$

Electron—Self energy

DEQs



ε-factorised

simple poles

$$d_x K = \varepsilon \begin{pmatrix} \frac{1}{2}\left(-\frac{4}{x-1}+\frac{1}{x}+\frac{36}{1-9x}\right) & \frac{6}{x(x-1)x(9x-1)\psi(x)^2} & 0 \\ \frac{\left(81x^4+1188x^3-594x^2+372x-23\right)\psi(x)^2}{24(x-1)x(9x-1)} & \frac{1}{2}\left(-\frac{4}{x-1}+\frac{1}{x}+\frac{36}{1-9x}\right) & \frac{3\psi(x)}{2x} \\ \frac{8\psi(x)}{3x} & 0 & -\frac{4}{x} \end{pmatrix} \cdot K\,.$$

Conclusion

- The algorithm to construct ε-factorised DEQs.
- Our method involves reorganising IBPs and using new coefficients.
- Generic algorithm independent of the geometric data.

Thank you for your attention

ε-collaboration

Iris Bree

Federico Gasparotto

Antonela Matijašić

Pouria Mazloumi

Dmytro Melnichenko

Sebastian Pögel

Toni Teschke

Xing Wang
(王星)

Stefan Weinzierl

Konglong Wu
(吴孔龙)

Xiaofeng Xu
(徐小峰)

Supplementary Material