Series expansion approach and application to 2L mixed corrections to Drell-Yan

In collaboration with R. Bonciani, S. Devoto, N. Rana, A. Vicini

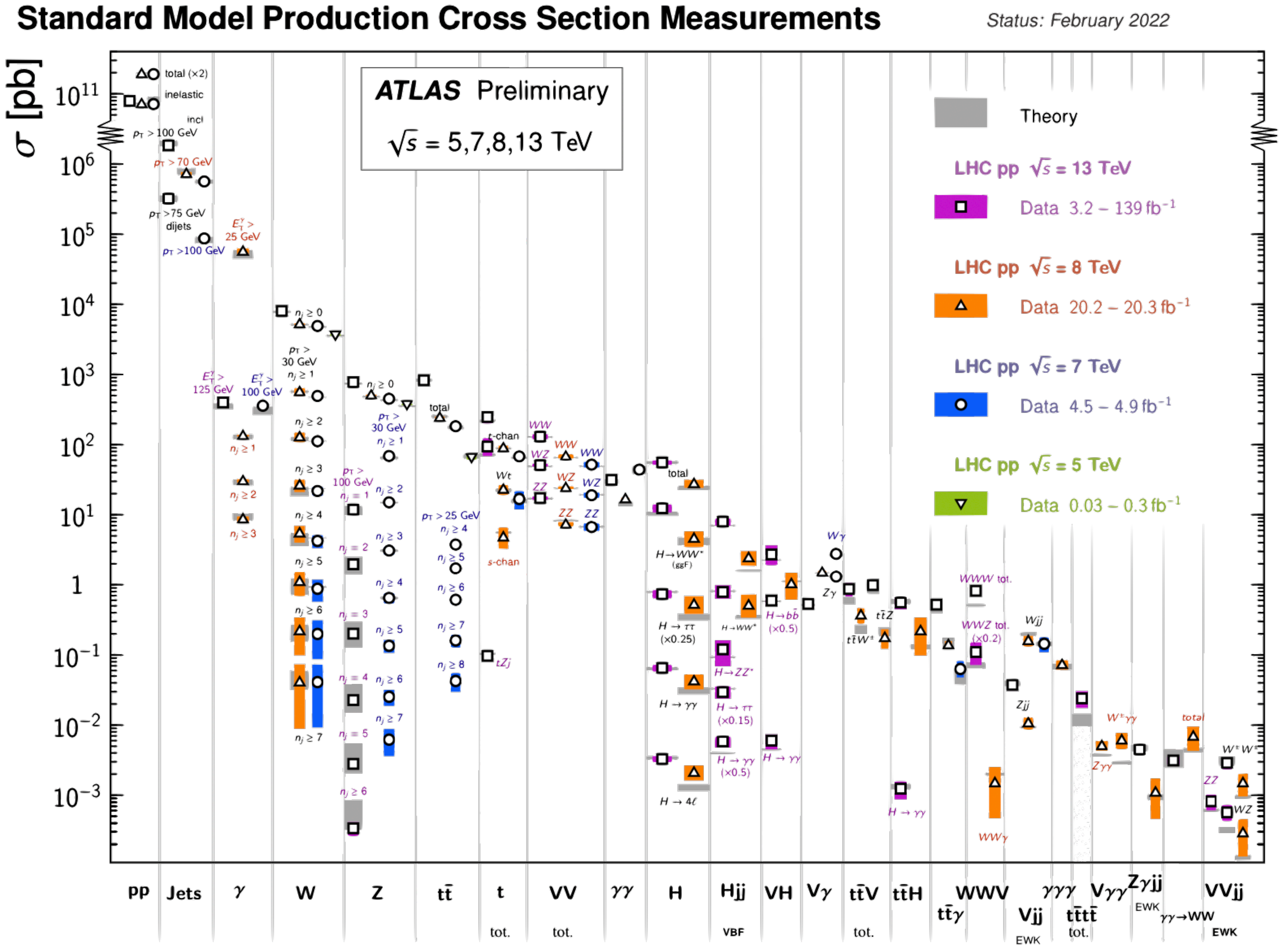
UCLouvain

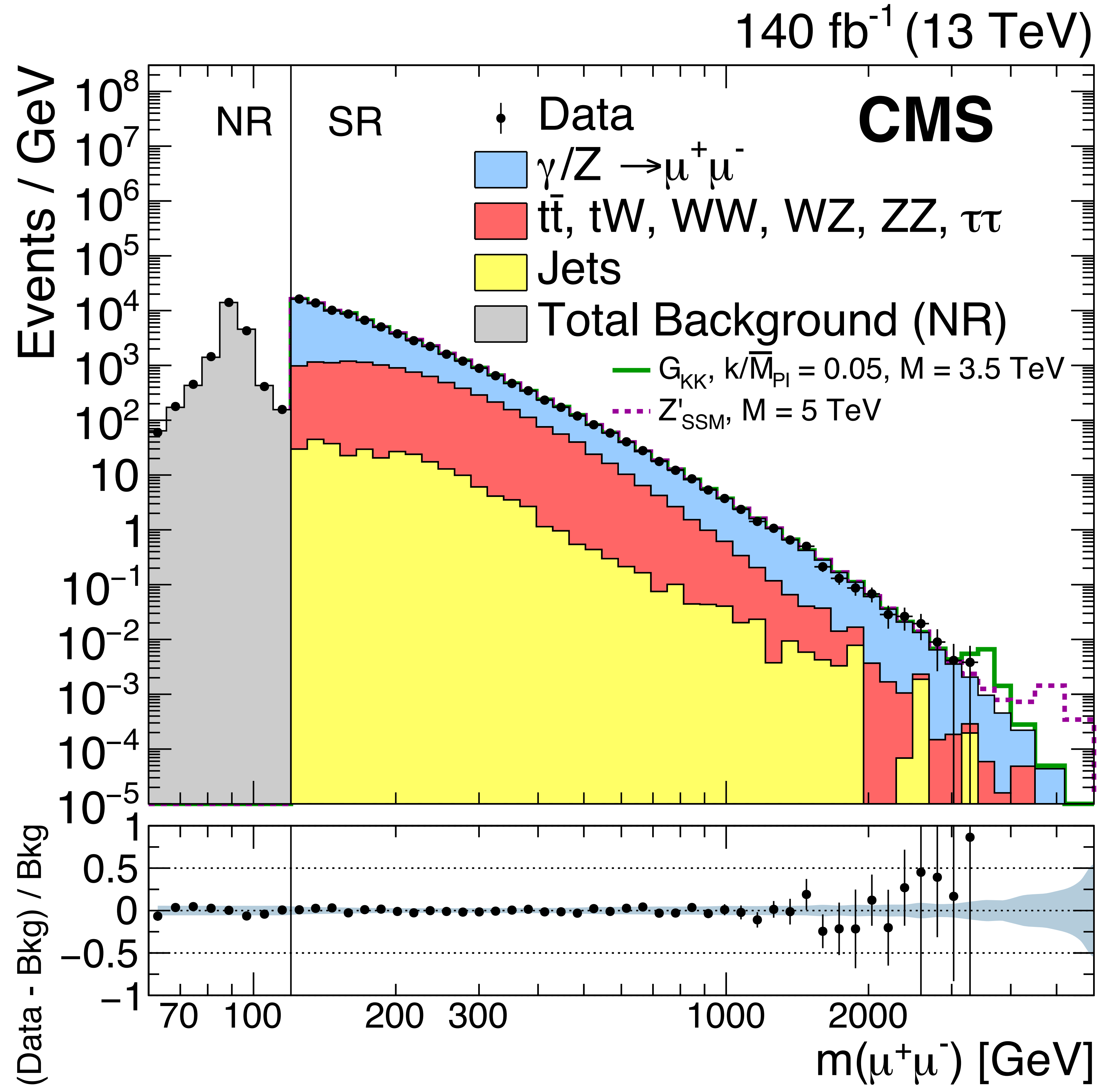
Tommaso Armadillo - UCLouvain & UNIMI

YOUNGST@RS - Loop-the-Loop - 12th November 2024

Motivations

- After the discovery of the Higgs, LHC is becoming more and more **precision machine**;



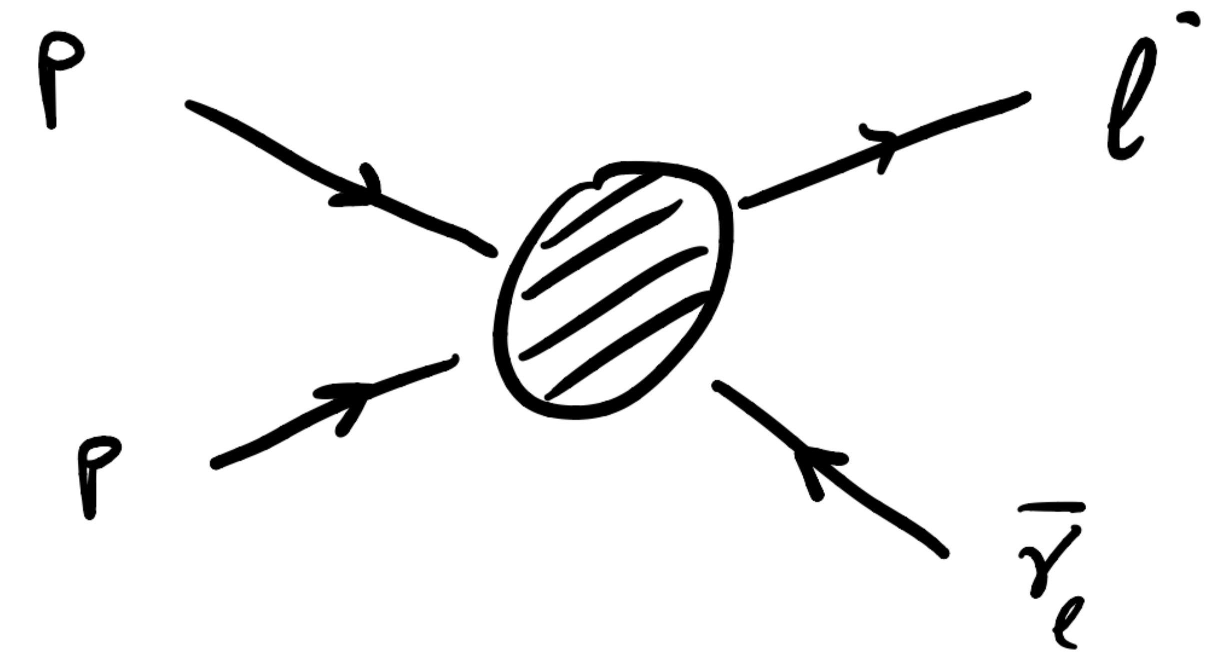


$\sigma(pp \rightarrow \mu^+\mu^- + X)$		
Bin range (GeV)	% error 140 fb^{-1}	% error 3 ab^{-1}
91-92	0.03	6 x 10^{-3}
120-400	0.1	0.02
400-600	0.6	0.13
600-900	1.4	0.30
900-1300	3.2	0.69

arXiv:2106.11953

Theoretical challenges

- We need **theoretical predictions at least as accurate as the experimental measurements**.



$$\sigma_{ij} = \sigma_{ij}^{(0,0)}$$
$$+ \alpha_s\, \sigma_{ij}^{(1,0)} + \alpha\, \sigma_{ij}^{(0,1)} +$$
$$+ \alpha_s^2\, \sigma_{ij}^{(2,0)} + \alpha_s\alpha\, \sigma_{ij}^{(1,1)} + \alpha^2\, \sigma_{ij}^{(0,2)} +$$
$$+ \alpha_s^3\, \sigma_{ij}^{(3,0)} + \dots$$

Higher order corrections

Feynman integrals

$$\sigma_{tot} = \sum_{i,j\in q,\bar{q},g,\gamma} \int_0^1 dx_1\, dx_2\, f_i(x_1,\mu_F)\, f_j(x_2,\mu_F)\, \sigma_{ij}(\mu_F,\mu_R)$$

Parton Distribution Functions

How to compute the Feynman Integrals?

- The first step is to reduce to a set of **Master Integrals**;

$$\sum_{i=1}^{N} c_i \, I_i = \sum_{j=1}^{n} \tilde{c}_j \, MI_j \qquad\qquad n \ll N$$

- One possible way to evaluate the MIs is through **differential equations**;

$$\frac{\partial}{\partial s_k} I(\alpha_i; s_j, d) = \sum scalar\ integrals = \sum master\ integrals$$

IBP

- By repeating the same process for every master integral we obtain a **system of first order linear and homogeneous differential equations**.

What are we looking for?

- So we just have to solve a system of first order differential equations… HOW?
- Ideally, we would like:
  - A method **easy to automatise**

```
ImportMasterIntegrals["my_master_integrals"]
ABISS: Succesfully imported 380 master integrals
```

  - A solution **compact and easy to handle** to allow for simplifications

$$\mathrm{Li}_2(x) + \mathrm{Li}_2\left(\frac{1}{x}\right) = -\frac{\pi^2}{6} - \frac{\log^2(-x)}{2}$$

$$\mathrm{Li}_2(x) + \mathrm{Li}_2(1-x) = \frac{\pi^2}{6} + \log(x)\log(1-x)$$

  - A solution **fast to evaluate** to be implemented in a Monte-Carlo
  - To have **high control on numerical precision**

$$\mathcal{O}(10 - 10^2) \leftarrow 2\mathrm{Re}\mathcal{M}^{(2)}\mathcal{M}^{(0)*} = \sum_i c_i\, MI_i \rightarrow \mathcal{O}(10^{-10} - 10^{10})$$

Analytical solution

- The first method is to solve it **analytically**;

$$B_0(p^2, m^2) = 2 - \gamma_E - \log m^2 + \frac{m^2}{p^2}\left(\frac{1}{r} - r\right)\log r \qquad \textit{with} \qquad r = \frac{-p^2 + 2m^2 + \sqrt{(p^2 - 2m^2)^2 - 4m^4}}{2m^2}$$

- The result is provided in closed form as a combination of elementary and special functions, such as **Generalised PolyLogarithms**;

$$G(a_1, \ldots, a_n; z) = \int_0^z \frac{dt}{t - a_1}\, G(a_2, \ldots, a_n; z) \qquad \textit{and} \qquad G(\vec{0}_n; z) = \frac{1}{n!}\log^n z$$

- When increasing the number of scales or legs, an analytical expression in terms of known classes of functions might not be available.

Semi-analytical solution

- A third possibility could be to use a **semi-analytical approach**. In this case the result is provided as a power series which can be easily evaluated in every point of the domain.

$$B_0(p^2, m^2 = 1) = -\gamma_E + \frac{1}{6}p^2 + \frac{1}{60}(p^2)^2 + \frac{1}{420}(p^2)^3 + \frac{1}{2520}(p^2)^4 + \frac{1}{13860}(p^2)^5 + \dots$$

- This method is quite **easy to automate**. Provided that we have infinite time and space, we could achieve **arbitrary precision**. Moreover, once we have the solution, it can be evaluated numerically in a negligible amount of time.
- However, **series have a limited radius of convergence**, hence, an algorithm for performing the analytic continuation of the solution must be provided.

SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

SEASYDE

A SIMPLE EXAMPLE

$$\begin{cases} f'(x) + \frac{1}{x^2-4x+5} f(x) = \frac{1}{x+2} \\ f(0) = 1 \end{cases}$$

$$f_{hom}(x) = x^r \sum_{k=0}^{\infty} c_k x^k$$

$$f_{hom}(x) = 5 - x - \frac{3}{10}x^2 + \frac{11}{150}x^3 + \dots$$

$$\begin{cases} r c_0 = 0 \\ \frac{1}{5}c_0 + c_1(r+1) = 0 \\ \frac{4}{25}c_0 + \frac{1}{5}c_1 + c_2(2+r) = 0 \\ \frac{11}{125}c_0 + \frac{4}{25}c_1 + \frac{1}{5}c_2 + c_3(3+r) = 0 \\ \dots \end{cases}$$

SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

SEASYDE

A SIMPLE EXAMPLE

$$\begin{cases} f'(x) + \dfrac{1}{x^2 - 4x + 5} f(x) = \dfrac{1}{x+2} \\ f(0) = 1 \end{cases}$$

$$f_{part}(x) = f_{hom}(x) \int_0^x dx' \, \frac{1}{(x'+2)} \, f_{hom}^{-1}(x')$$
$$= \frac{1}{2}x - \frac{7}{40}x^2 + \frac{2}{75}x^3 + \ldots$$

$$f(x) = c\, f_{hom}(x) + f_{part}(x)$$
$$= 1 + \frac{3}{10}x - \frac{47}{200}x^2 + \frac{3}{250}x^3 + \ldots$$

SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

SEASYDE

A SIMPLE EXAMPLE

$$\begin{cases} f'(x) + \frac{1}{x^2 - 4x + 5} f(x) = \frac{1}{x+2} \\ f(0) = 1 \end{cases}$$

$$\begin{aligned} f(x) &= c\, f_{hom}(x) + f_{part}(x) \\ &= 1 + \frac{3}{10}x - \frac{47}{200}x^2 + \frac{3}{250}x^3 + \dots \end{aligned}$$

- This procedure can be generalised to **systems of differential equations**;
- The method has been firstly implemented in the Mathematica package **DiffExp** for a **real kinematic variable** *[F.Moriello, arXiv:1907.13234], [M.Hidding, arXiv:2006.05510]*

SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

https: //github.com/TommasoArmadillo/SeaSyde

SEASYDE

- Our goal in the end is to fit the W mass to the data, hence, we need to employ a gauge invariant definition of the mass. For this reason, it is important to perform the calculations in the **complex-mass scheme**.

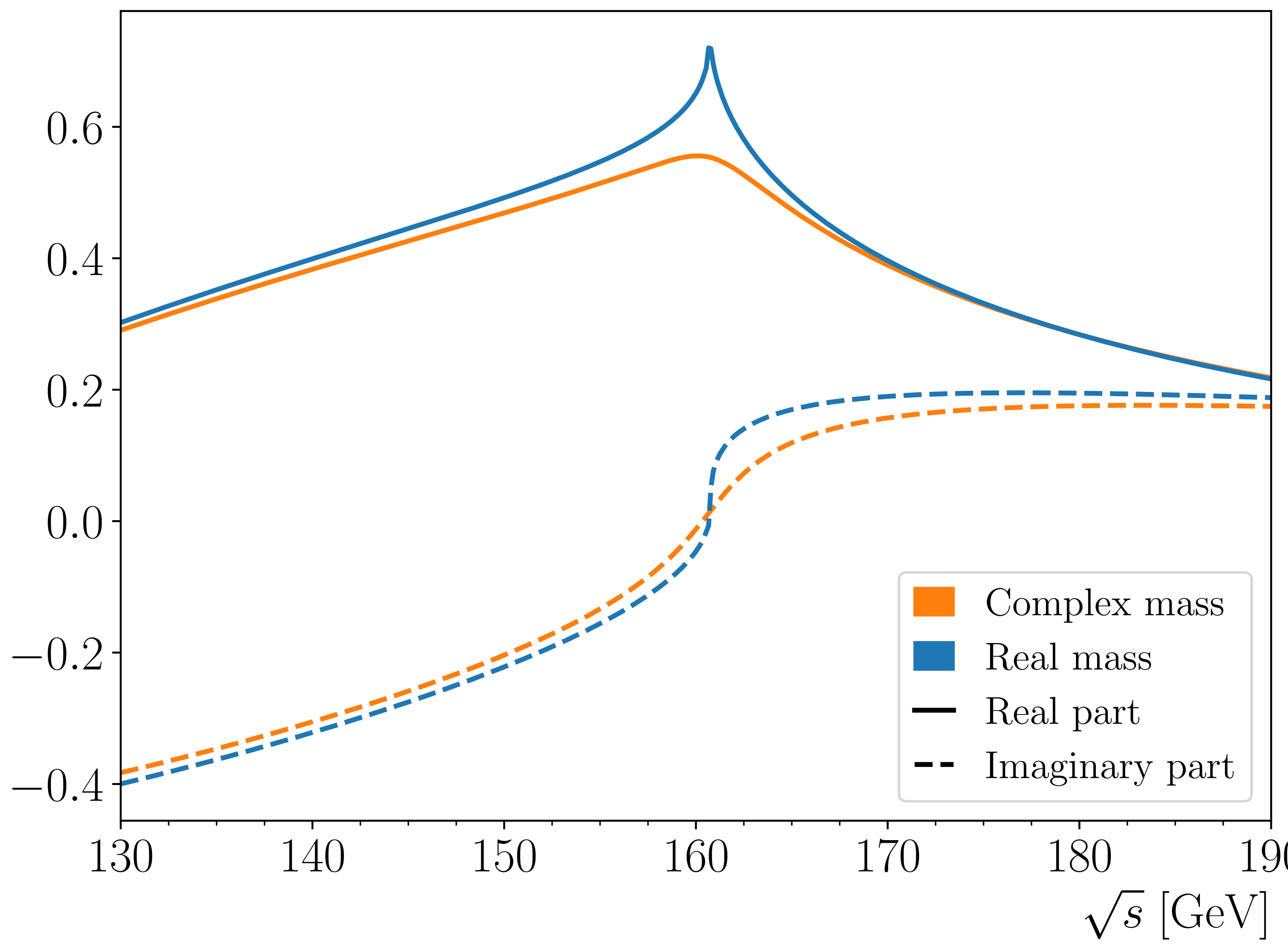
$$\mu_V^2 = m_V^2 - i\Gamma_V m_V$$

- The complex mass scheme **regularises** the behaviour at the resonance:

$$\frac{1}{s - \mu_V^2 + i\delta}$$

- If we utilise **adimensional variables**, they become complex-valued:

$$\tilde{s} = \frac{s}{m_V^2} \to \frac{s}{\mu_V^2}$$

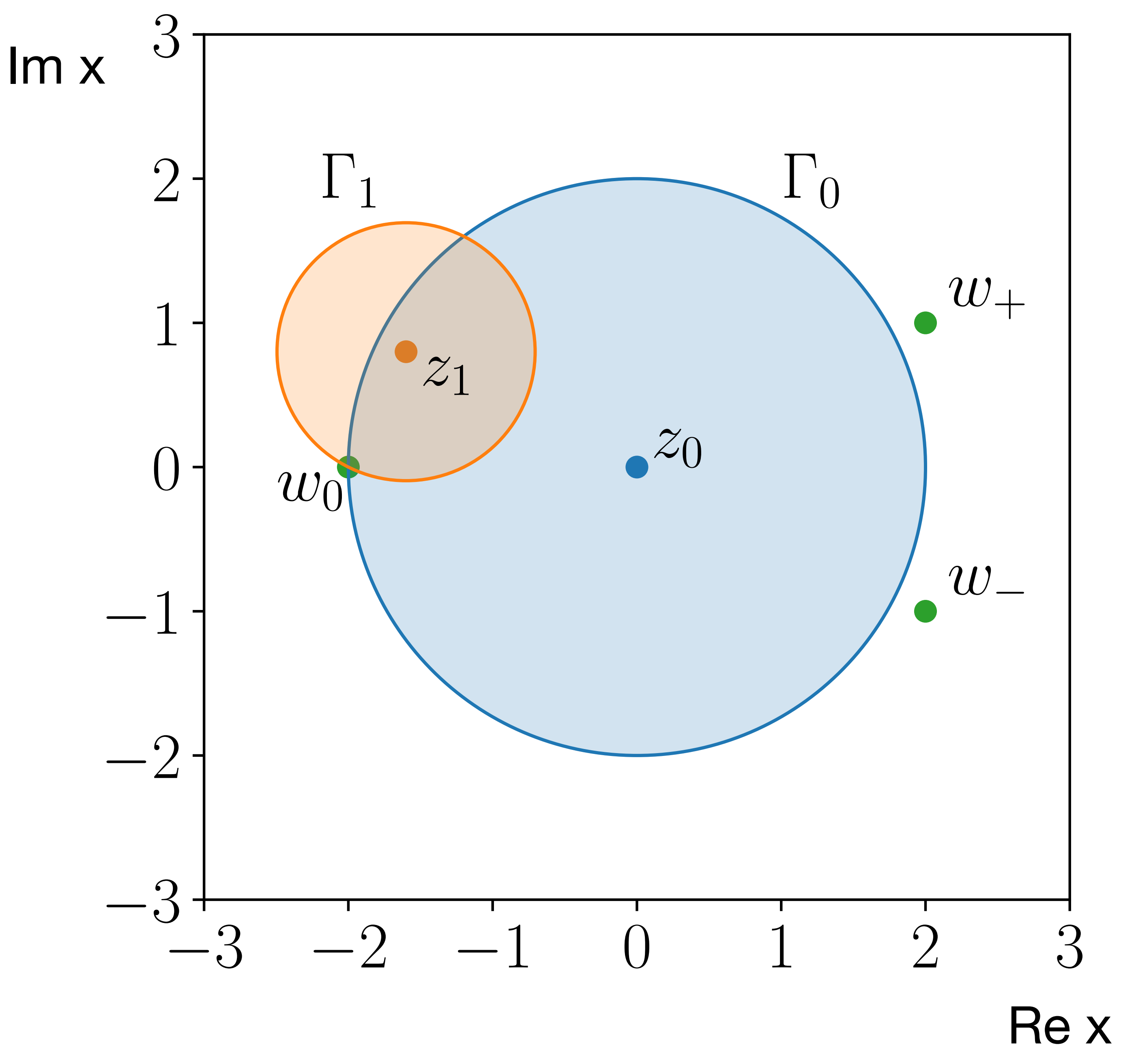


SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

https: //github.com/TommasoArmadillo/SeaSyde

- Power series have a limited **radius of convergence** which is determined by the position of the **nearest singularity**.
- We need to be able to extend the solution beyond the radius of convergence, to the entire **complex plane**.



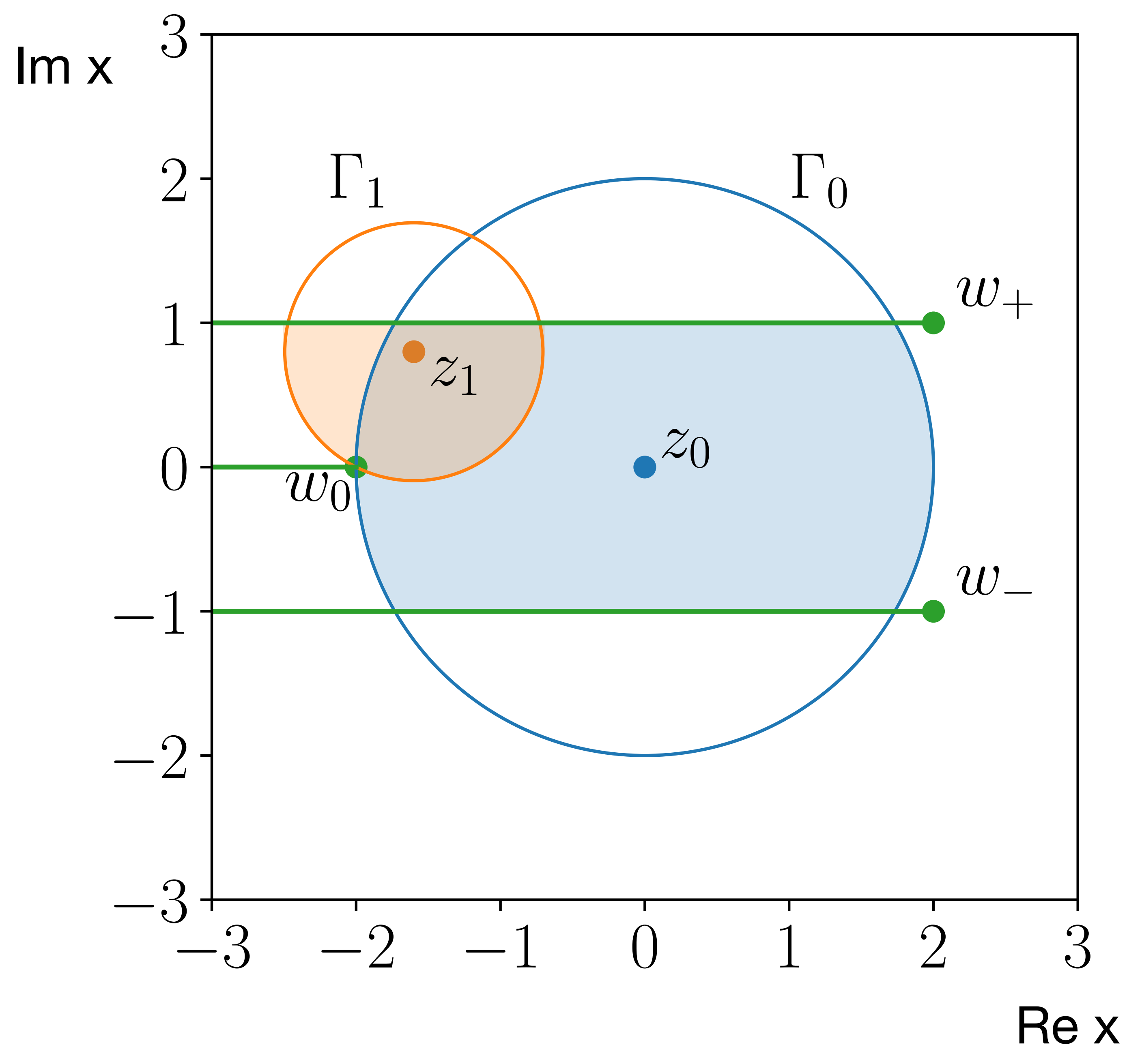
SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

https: //github.com/TommasoArmadillo/SeaSyde

SEASYDE

- Power series have a limited **radius of convergence** which is determined by the position of the **nearest singularity**.
- We need to be able to extend the solution beyond the radius of convergence, to the entire **complex plane**.

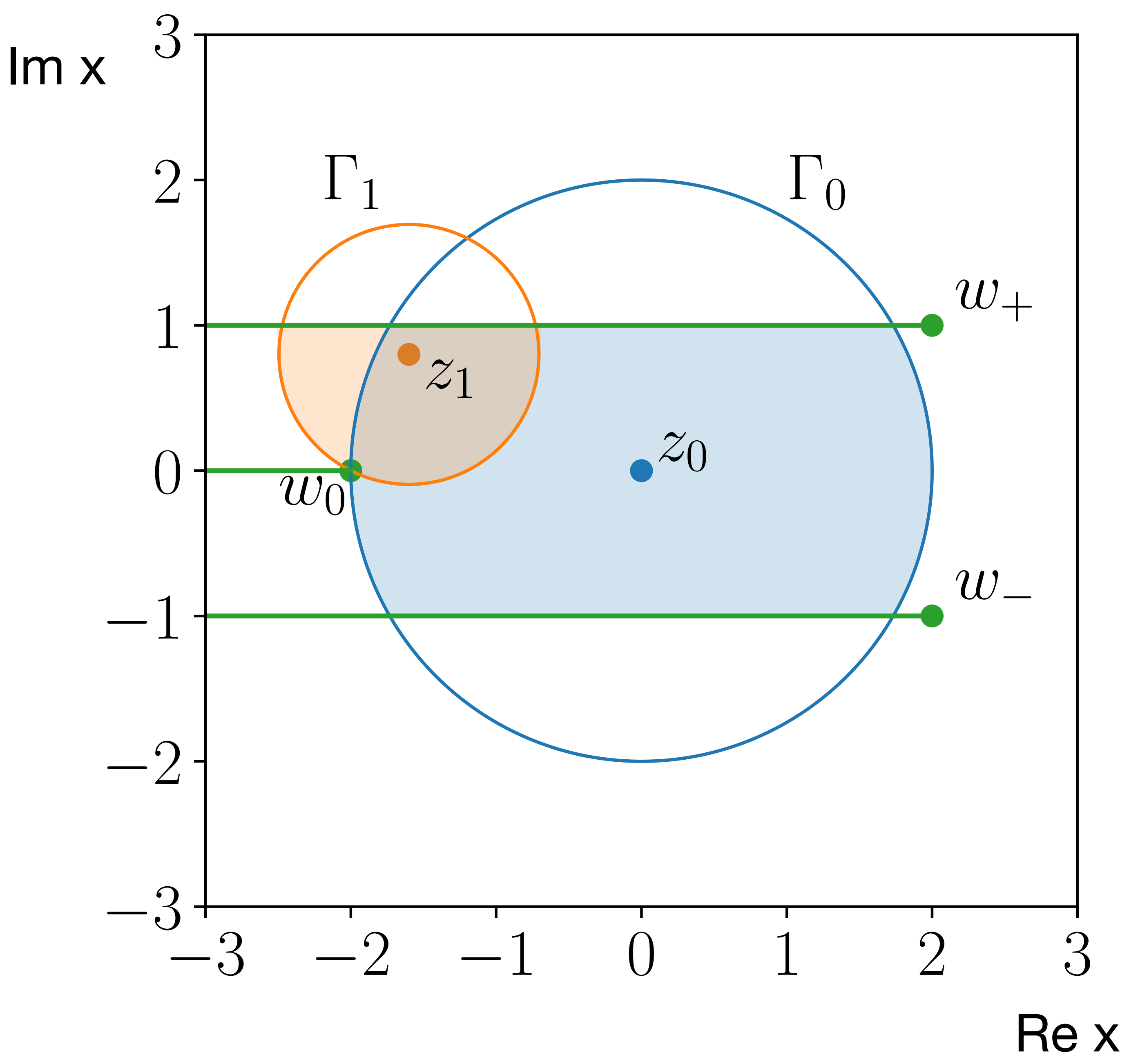


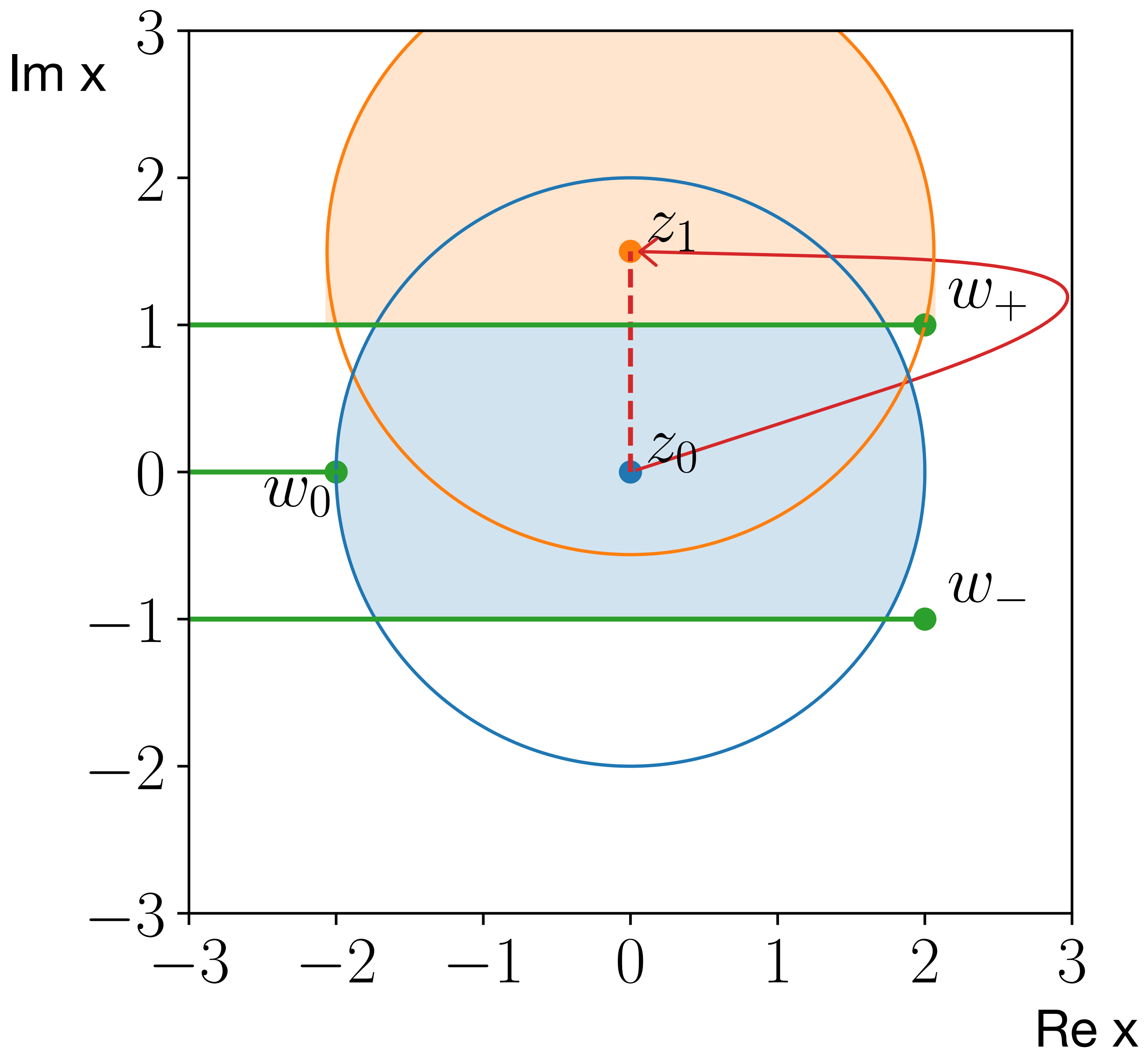
SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

https: //github.com/TommasoArmadillo/SeaSyde

- Power series have a limited **radius of convergence** which is determined by the position of the **nearest singularity**.
- We need to be able to extend the solution beyond the radius of convergence, to the entire **complex plane**.





*For simplicity, we are not showing all the intermediate circles.

SeaSyde

[**TA**, R. Bonciani, S. Devoto, N.Rana, A.Vicini, arXiv:2205.03345]

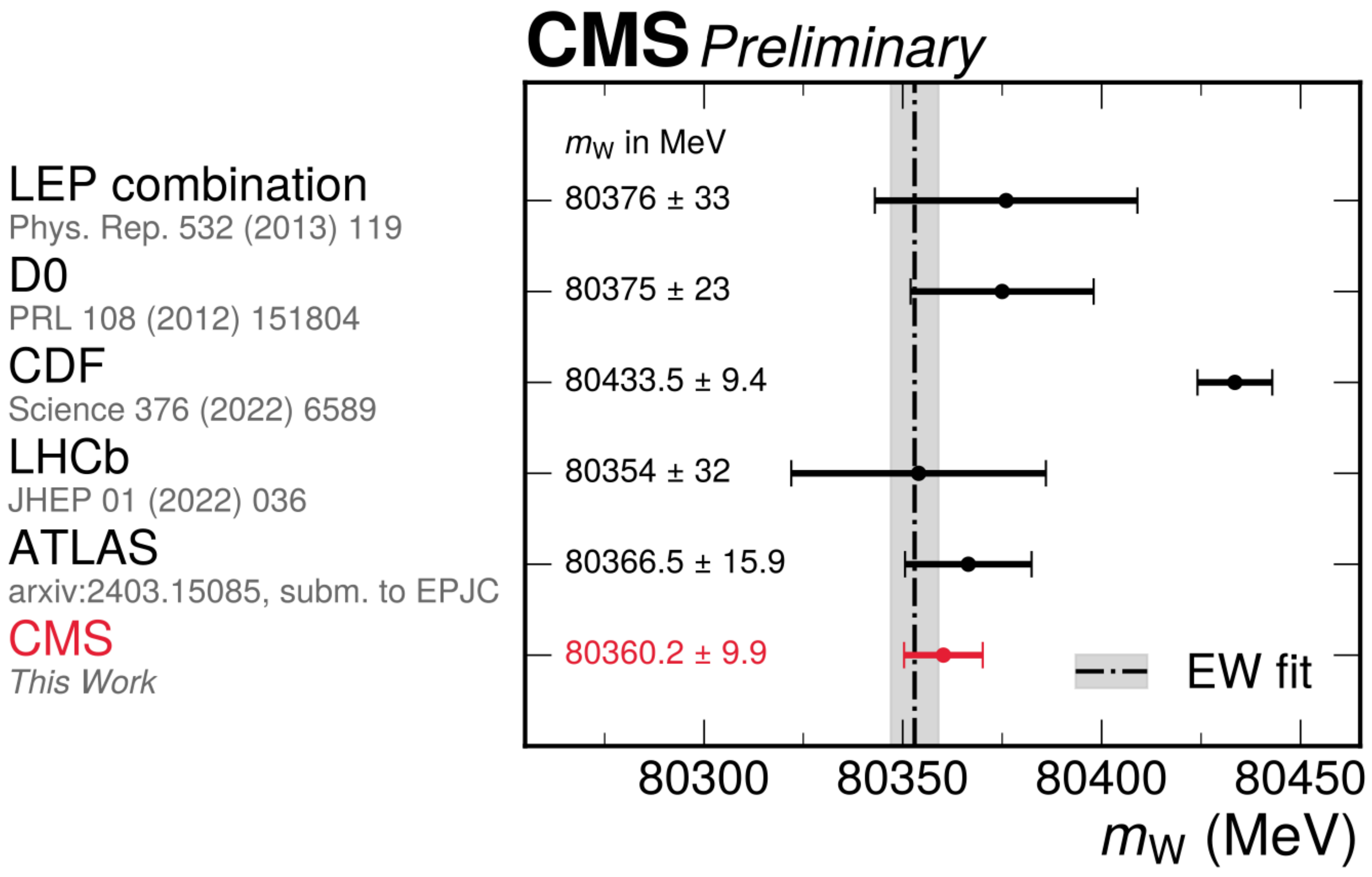
https: //github.com/TommasoArmadillo/SeaSyde

SEASYDE

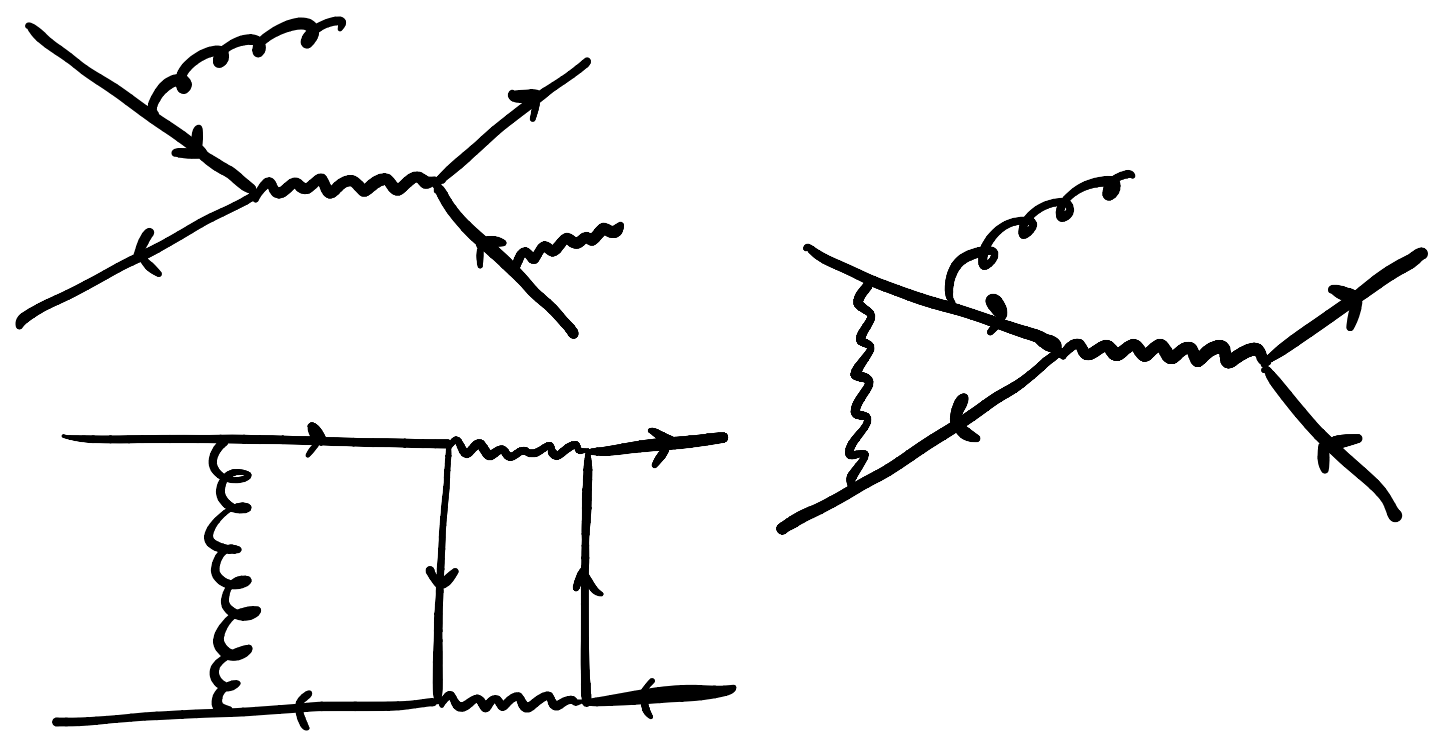
- **SeaSyde** (**S**eries **E**xpansion **A**pproach for **SY**stems of **D**ifferential **E**quations) is a general package for solving a system of differential equations using the series expansion approach;
- Seasyde can handle **complex kinematic variables** by introducing an original algorithm for the analytic continuation in the complex plane, thus being able to handle **complex internal masses**;
- **SeaSyde** can deal with arbitrary system of differential equations, covering also the case of **elliptic integrals**.

Drell-Yan

- High precision predictions for the Drell-Yan are important for the m_W **measurement**;



$$
\begin{aligned}
\sigma_{ij} = \sigma_{ij}^{(0,0)} \\
&+ \alpha_s\, \sigma_{ij}^{(1,0)} + \alpha\, \sigma_{ij}^{(0,1)} + \\
&+ \alpha_s^2\, \sigma_{ij}^{(2,0)} + \alpha_s \alpha\, \sigma_{ij}^{(1,1)} + \alpha^2\, \sigma_{ij}^{(0,2)} + \\
&+ \alpha_s^3\, \sigma_{ij}^{(3,0)} + \ldots
\end{aligned}
$$



Extremely important for high precision phenomenology (per-cent and sub per-cent level)

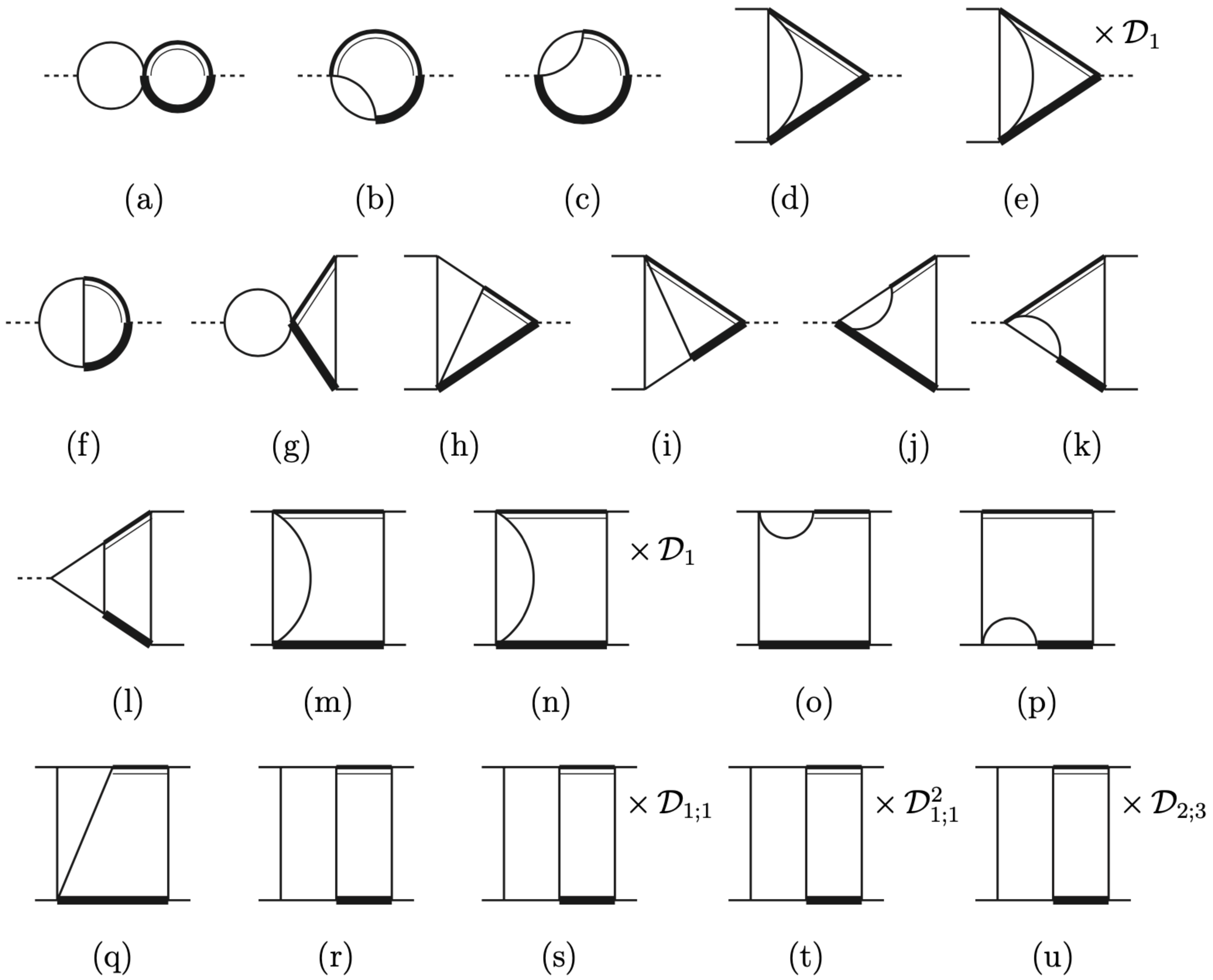
Reduction to Master Integrals

- We identified **11 integral families** with either 0, 1 or 2 masses. We reduced them to Master Integrals using **Kira** in combination with **Firefly**. The complete reduction took $\mathcal{O}(16h)$.

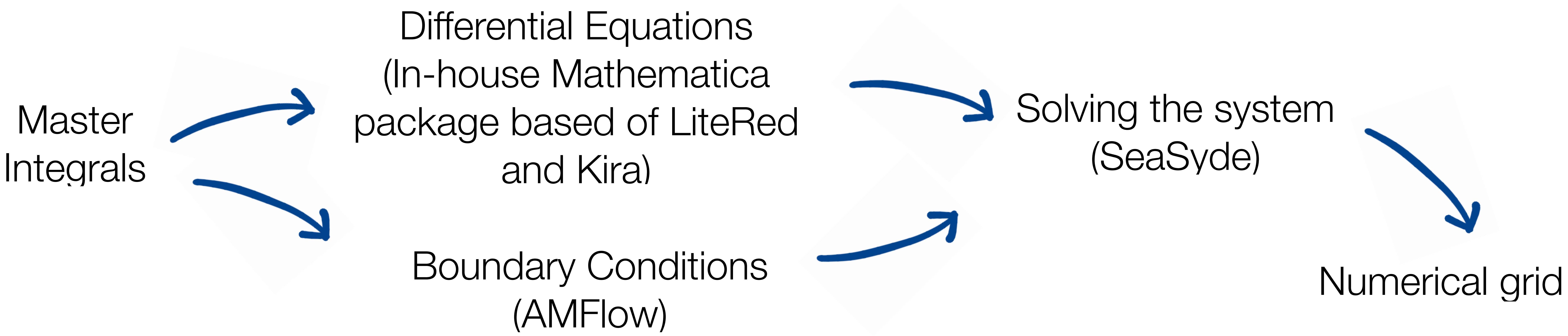
- We ended up with **274 masters integrals** to evaluate.

- The most complicated topology was a **two-loop box with two internal different masses**;

- We evaluated all the masters using the method of **differential equations**, using a **semi-analytical approach**.



Creating a grid



- The computation of a **grid with 3250 points** for the two-loop box with two internal and different masses (56 equations) required ∼3 weeks on 26 cores.
- This approach is **completely general and easy to automate**, and can be applied, in principle, to **any integral family**.

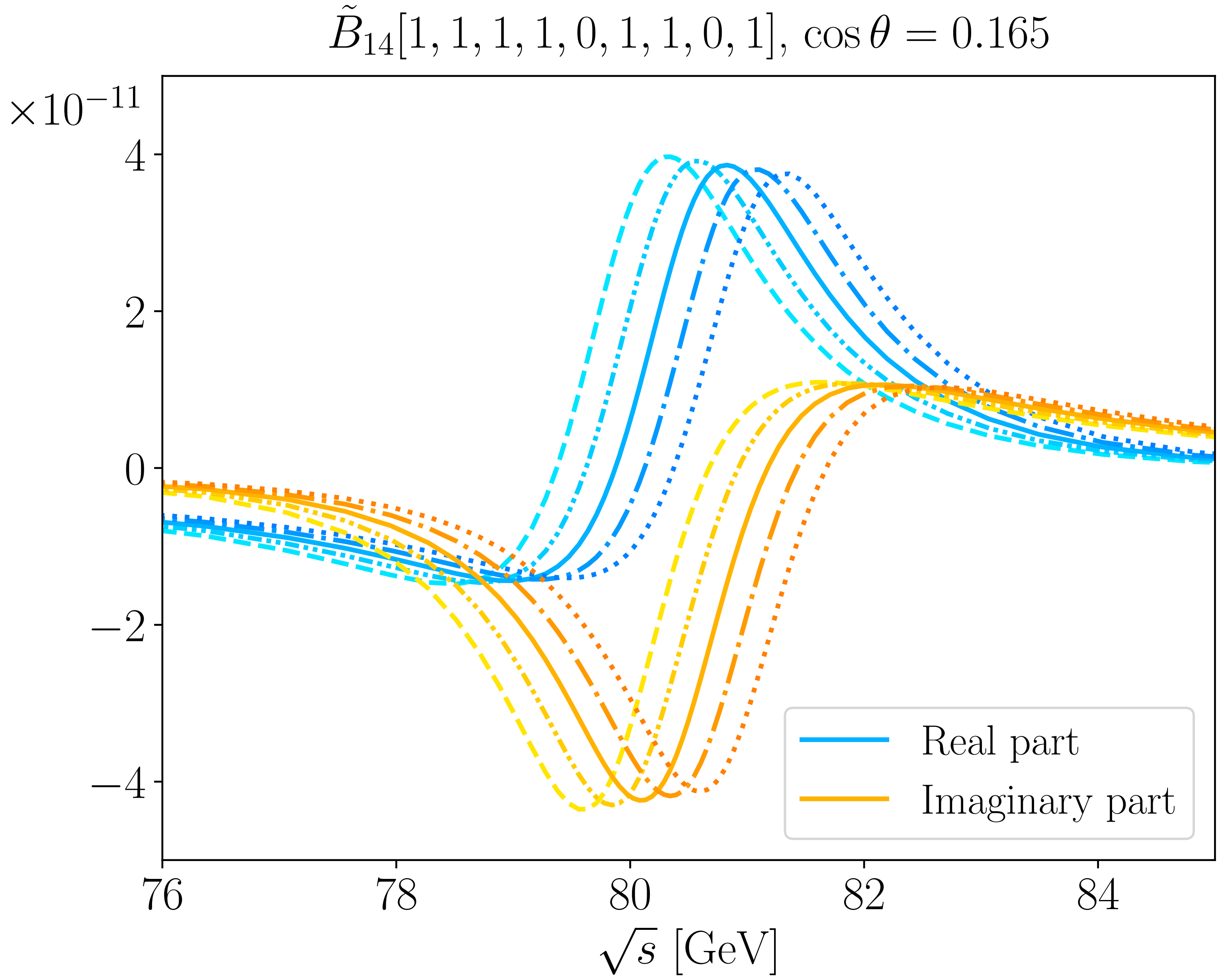
The expansion in $\delta\mu_W$

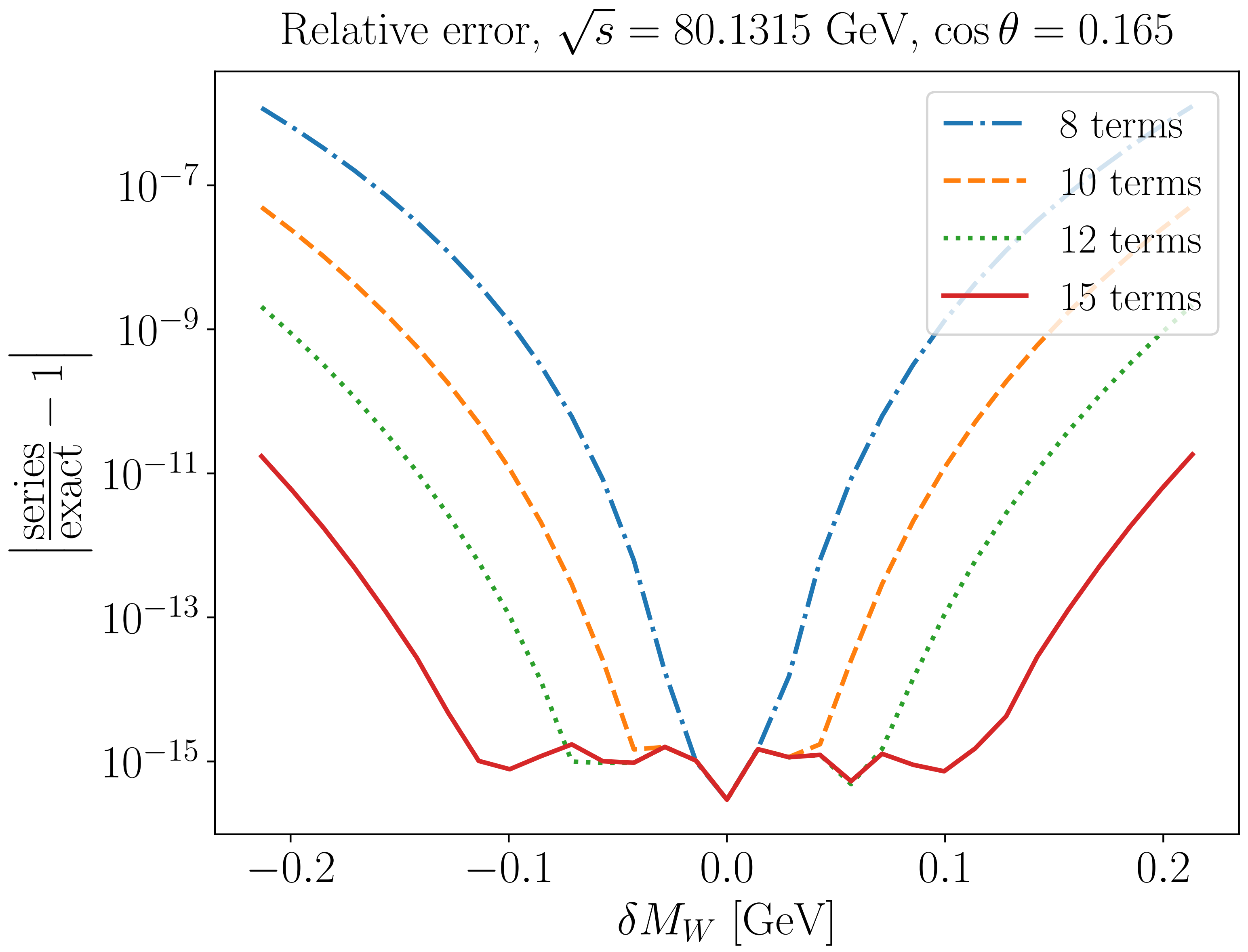
- In m_W **determination studies** we need $\mathcal{O}(10^2)$ templates with different values of μ_W. If we need 3 weeks for a single grid, this is **not feasible**.

Grid point
$s = s_0, t = t_0$

Diff. eqs.
w.r.t μ_W

For each point we reconstruct **a posteriori** the μ_w dependence





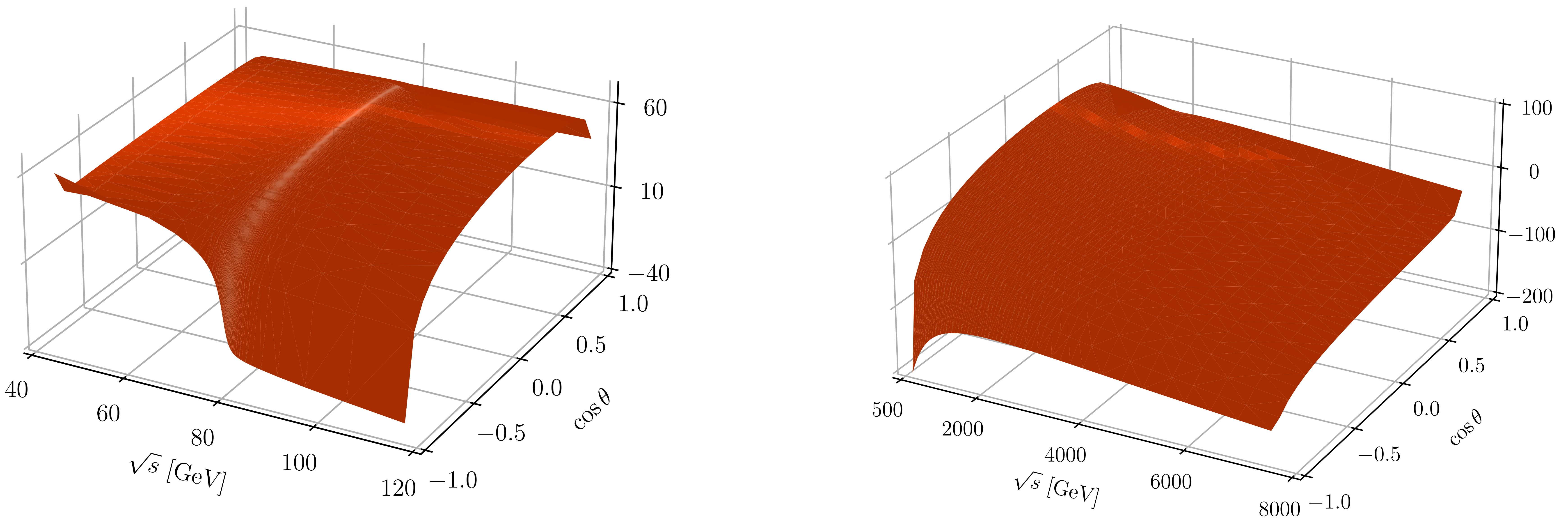
- Every point of our grid becomes a **series expansion** in $\delta\mu_W = \mu_W - \bar{\mu}_W$, which can be evaluated in a negligible amount of time for arbitrary (but reasonable) values of the W mass;
- The calculation of the $\delta\mu_W$ expansion for the entire grid took $\sim$ 1.5 days.

The hard function

- We present our final result in the form of the **hard function** $H^{(1,1)}$, which can be passed to a Monte-Carlo generator, e.g. **MATRIX**

$$H^{(1,1)} = \frac{1}{16}\left[2\mathrm{Re}\left(\frac{\langle \mathcal{M}^{(0,0)} | \mathcal{M}^{(1,1)}_{fin}\rangle}{\langle \mathcal{M}^{(0,0)} | \mathcal{M}^{(0,0)}\rangle}\right)\right]$$

- We can interpolate the value of $H^{(1,1)}$ in the entire phase-space. Thanks to its smoothness the error is, at worst, at the 10^{-3} level.



Summary & Outlook

- We presented how to use the series expansion approach to compute **2-loop Feynman integrals with internal (complex) masses**;

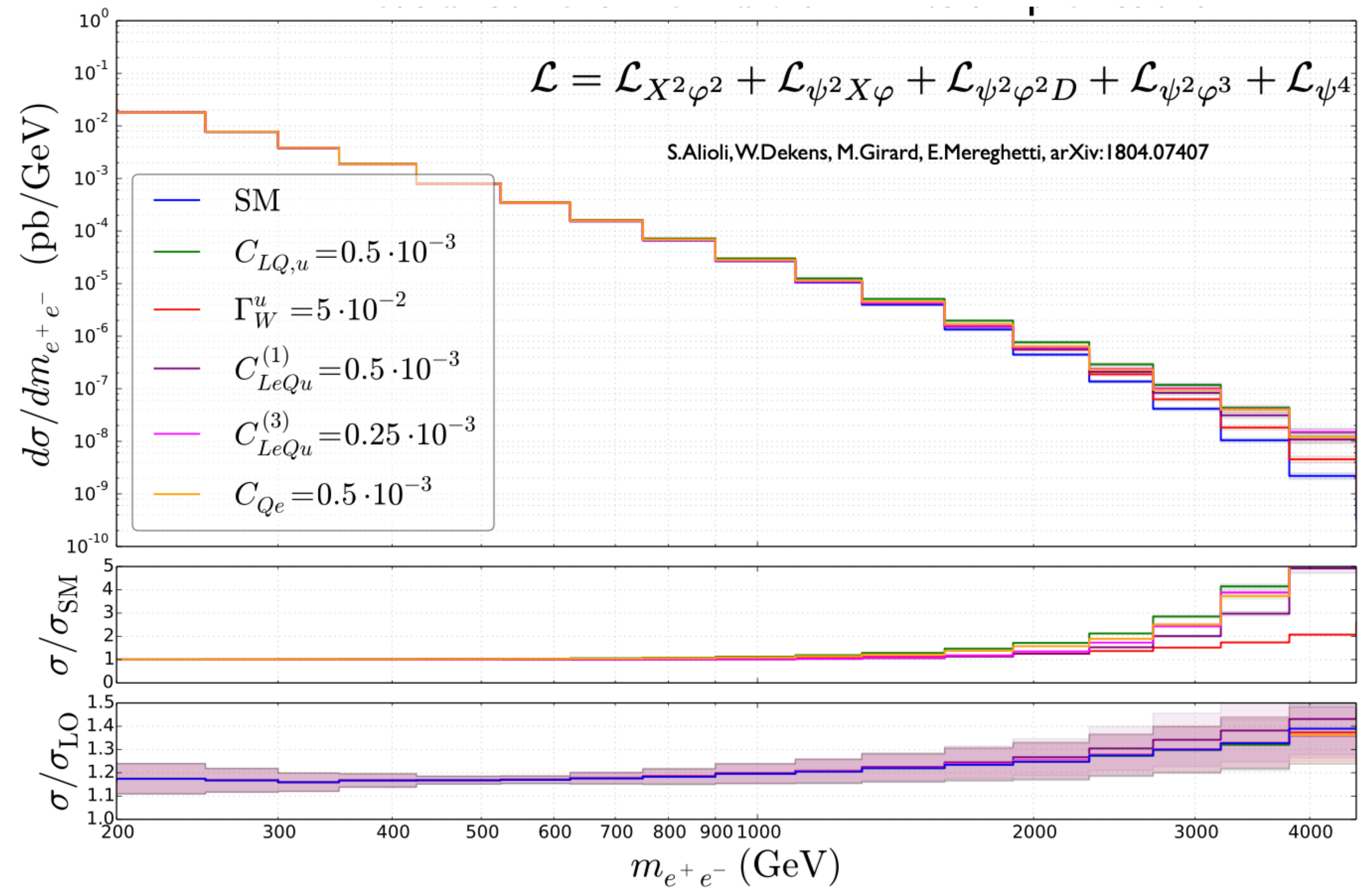
- We showed how we applied those techniques to the computation of the **virtual contribution to mixed QCD-EW corrections to Charged-current Drell-Yan**;

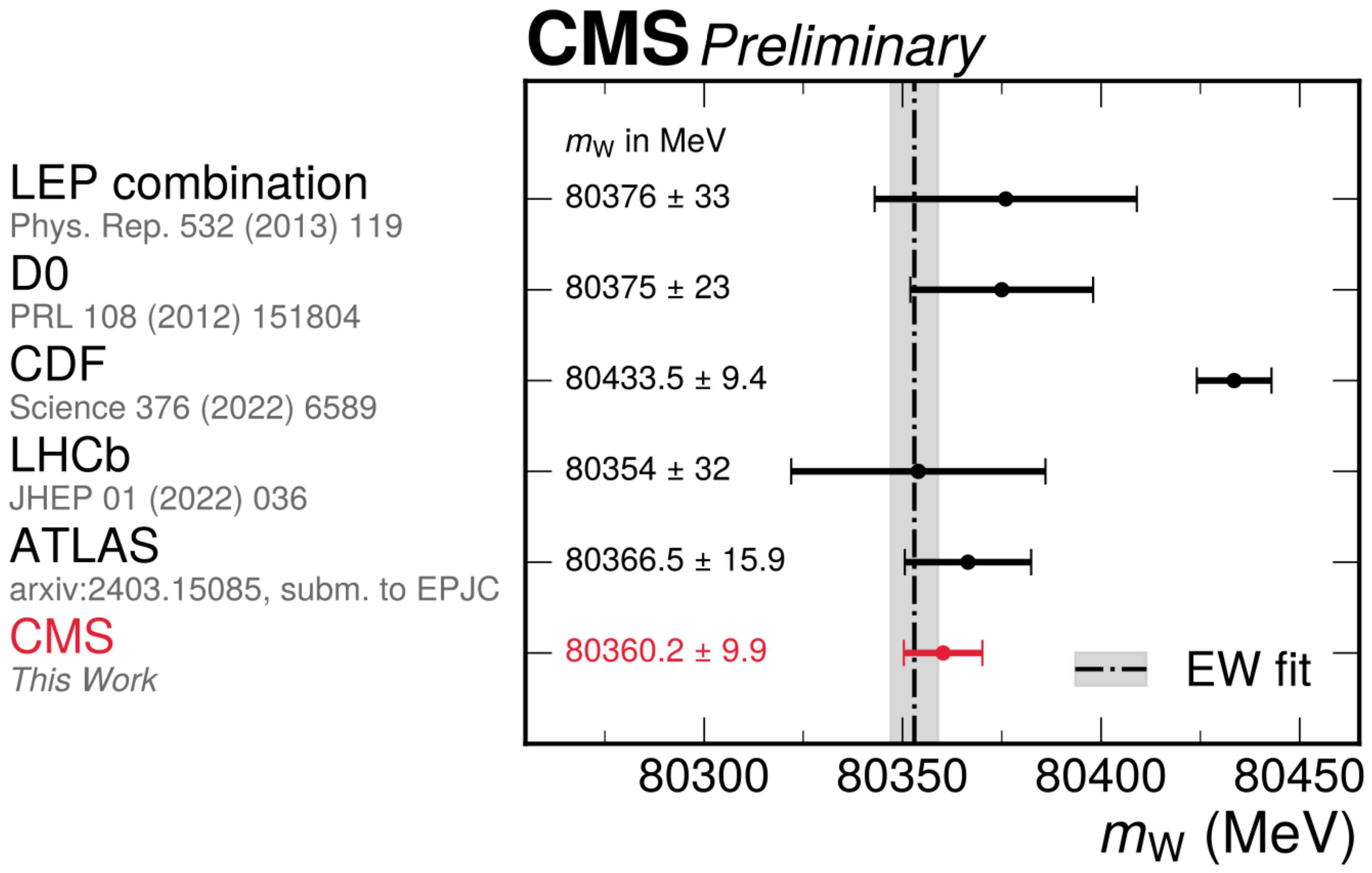
- We showed how to reconstruct a posteriori the **exact dependence on the W mass**;

- Finally, the techniques employed in this calculation are completely general, and can be applied to **other relevant 2->2 process** at **NNLO QCD-EW** level or even **NNLO EW**.

THANK YOU

Motivations





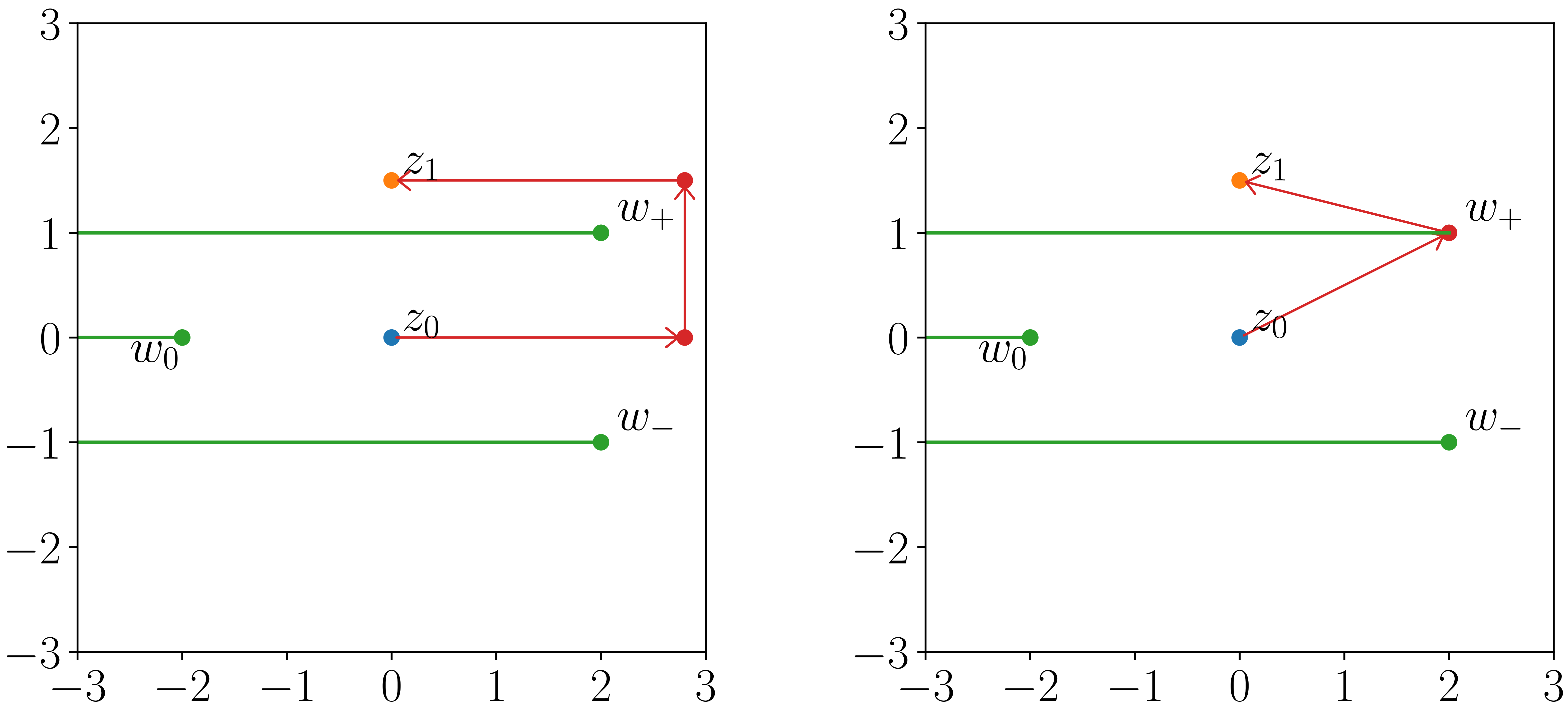
- The **computational challenges** are similar to the ones for FCC-ee.

$\sigma(e^+e^- \to \mu^+\mu^- + X)$			
sqrt(s) (GeV)	Luminosity (ab^{-1})	σ (fb)	% error
91	150	2.17595×10^6	0.0002
240	5	1870.84 ± 0.612	0.03
365	1.5	787.74 ± 0.725	0.09

arXiv:2206.08326

Taylor vs Logarithmic

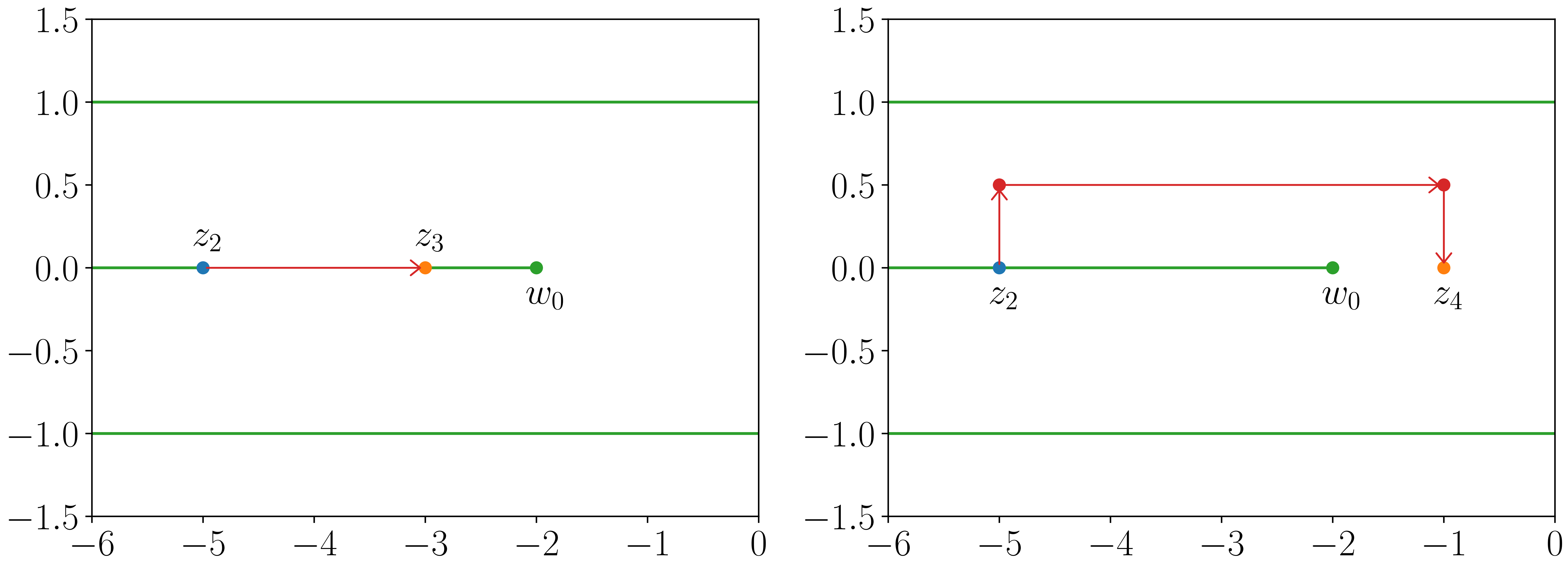
- When moving along an horizontal line, the Feynman prescription plays an important role



Analytic continuation

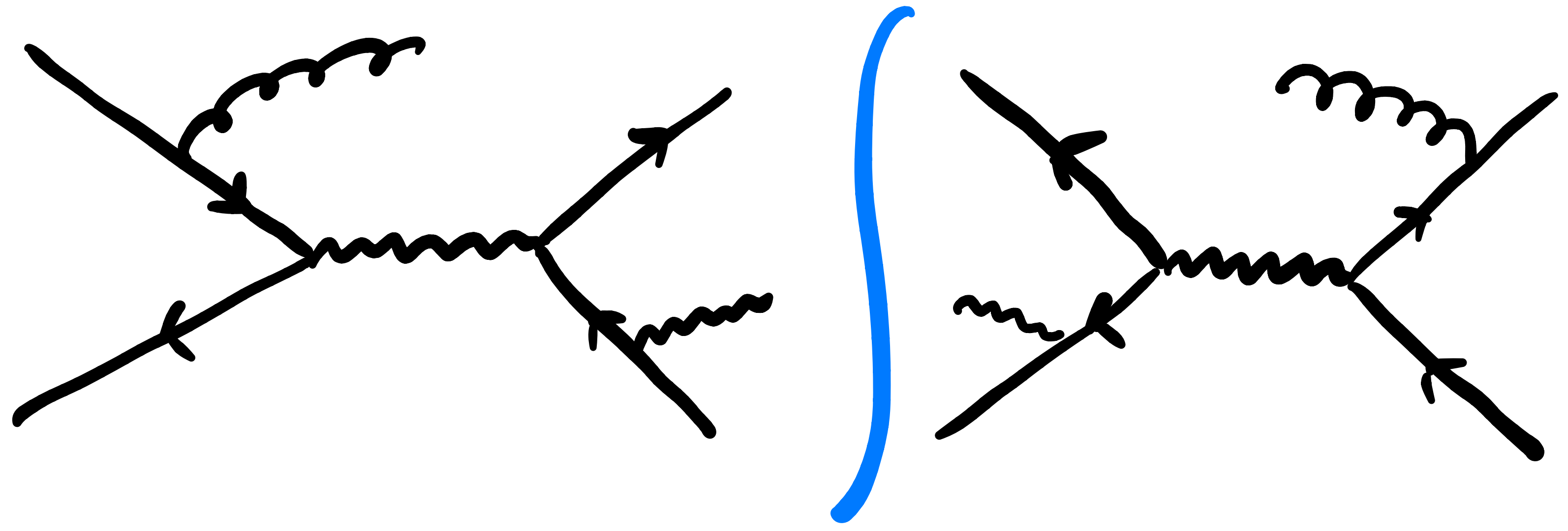
- When moving along an horizontal line, the **Feynman prescription** plays an important role

$$\frac{1}{s - m_V^2 + i\delta}$$

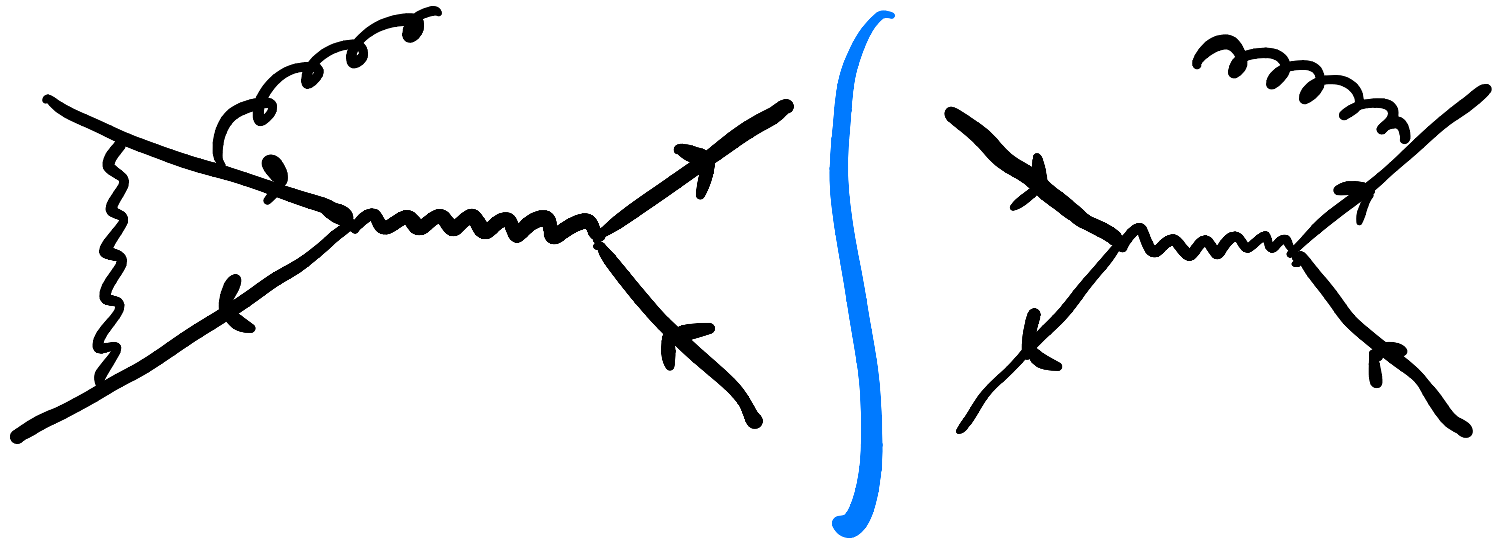


NNLO corrections

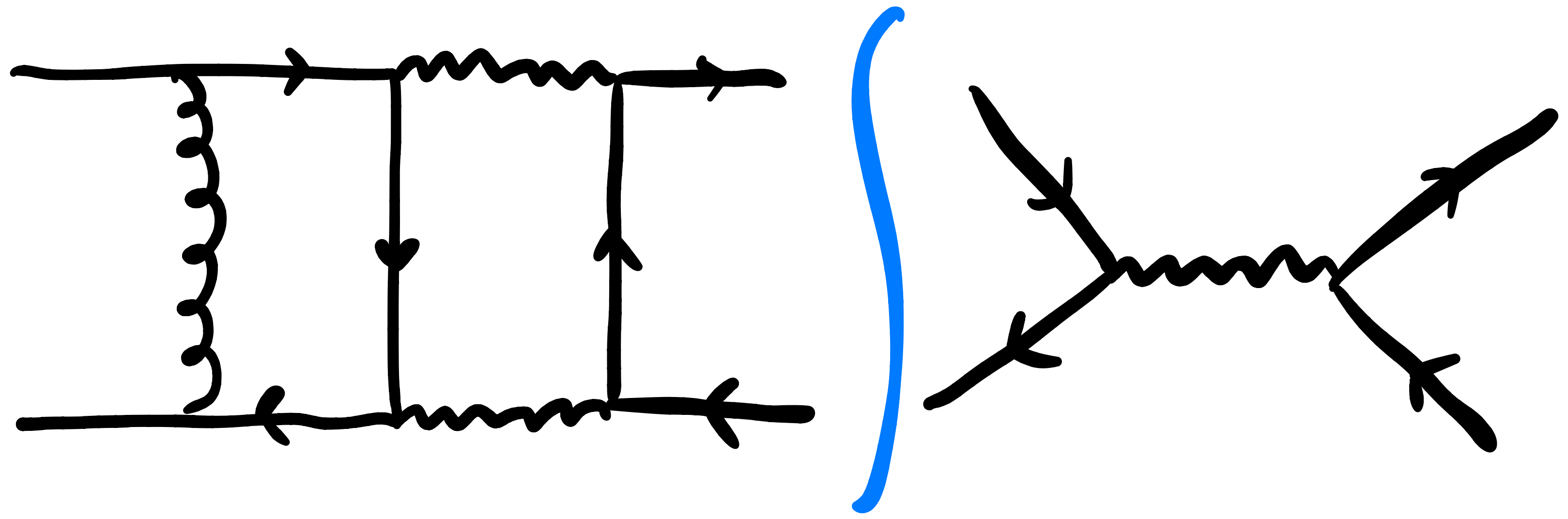
Double Real



Real-Virtual



Pure Virtual



Feynman Integrals

- Each of the three pieces carries its own challenges;
- The **pure virtual contributions are usually the main bottleneck**;
- Each individual contribution is divergent in the dimensional regulator ϵ.

Evaluating Feynman integrals

- What we would like to compute are objects like this:

$$d = 4 - 2\epsilon$$

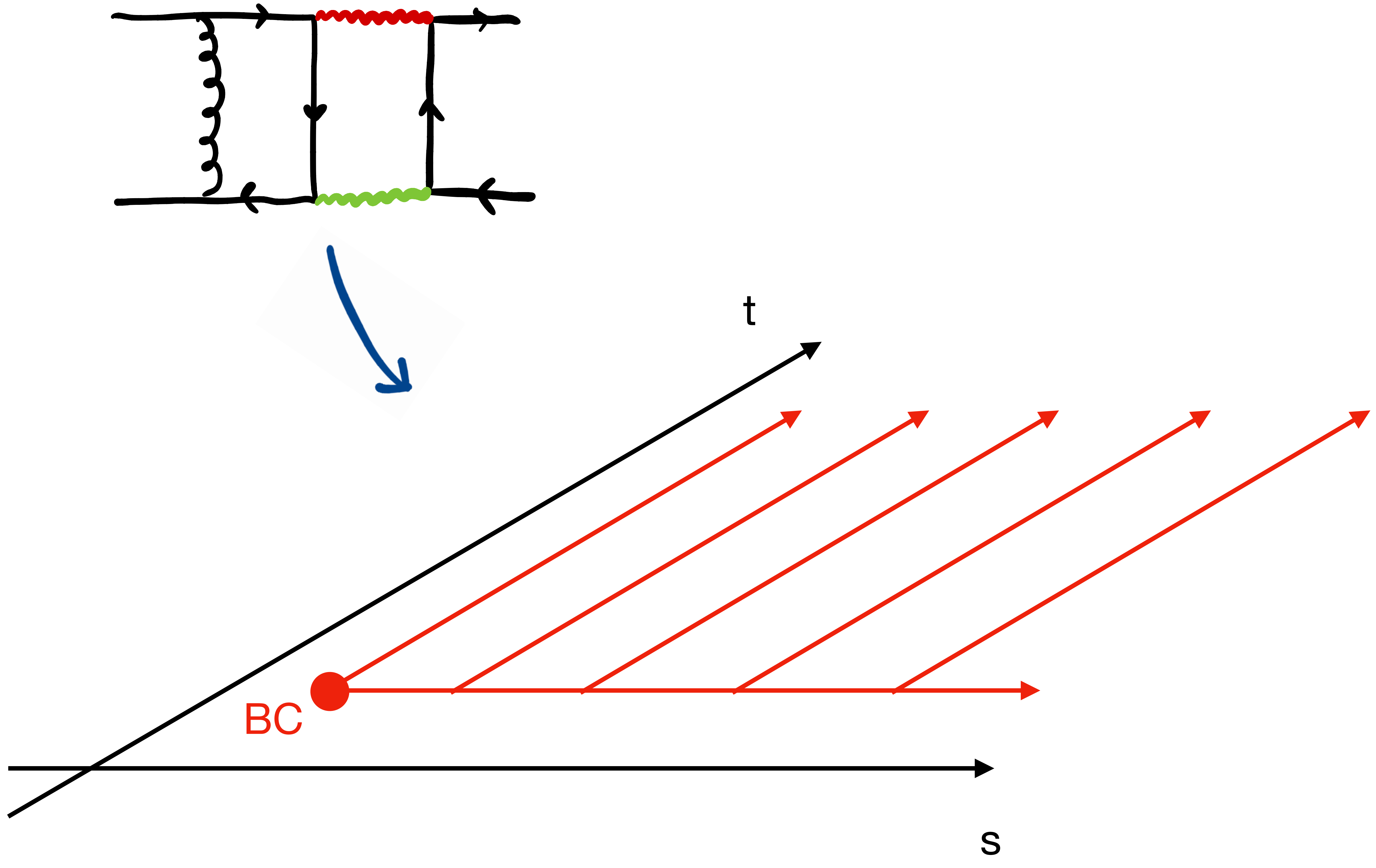
$$I(\alpha_i; s_j, d) = \int \prod_{k=1}^{l} \frac{d^d q_k}{i\pi^{d/2}} \frac{1}{\mathscr{D}_1^{\alpha_1} \dots \mathscr{D}_n^{\alpha_n}}$$

kinematic variables

$e.g.\ \ (p_1 - q_1)^2 - m^2 + i\delta$

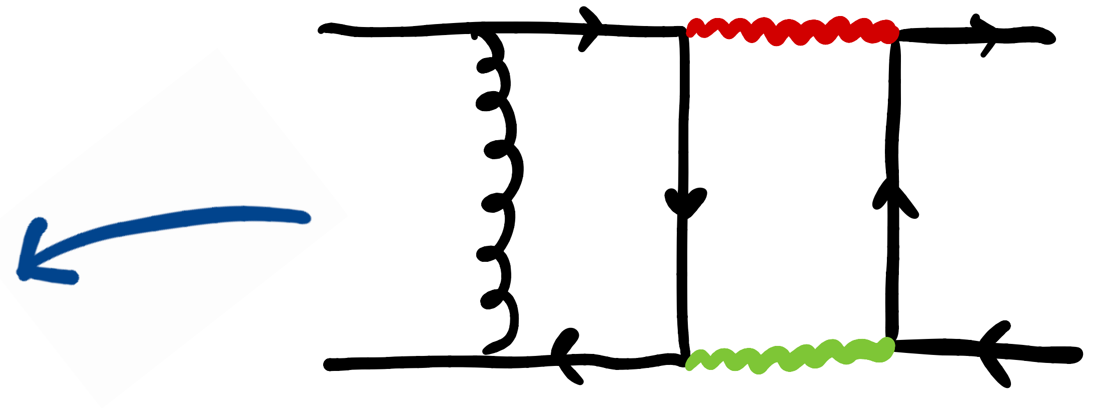
- A given set of denominators $\mathscr{D}_i$ constitutes an **integral family**. Inside an integral family an integral is uniquely identified by the set of the different powers α_i to which the denominators are raised.
- Using **Integration by Parts** (IBP) identities, we can express all the integrals of the given integral family in terms a smaller subset, the so-called **Master Integrals**.

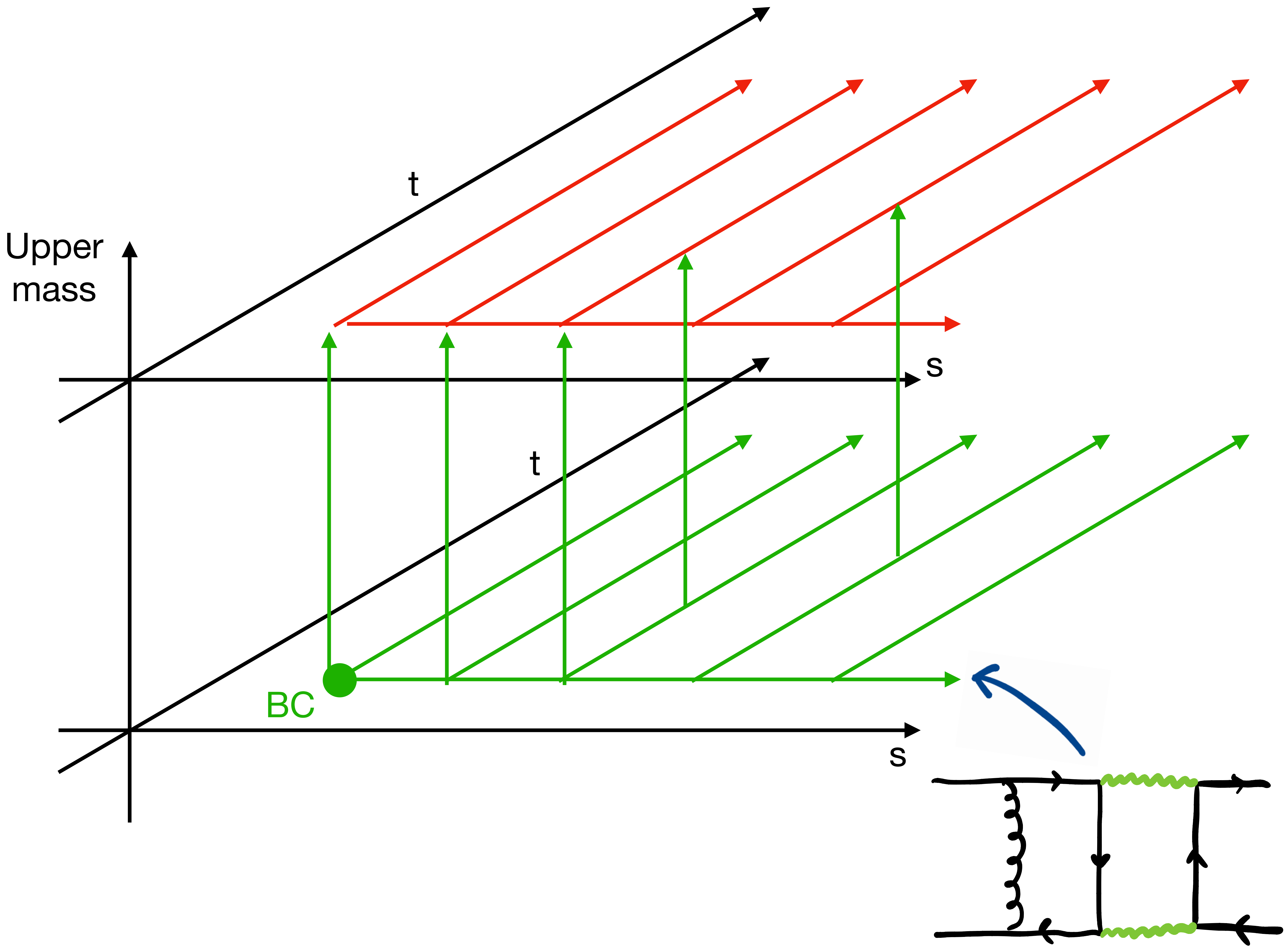
Creating a grid



- This approach is completely general and **easy to automate**;
- We have to solve a 56x56 system of differential equations w.r.t. to the Mandelstam variables s and t;
- Since we are not putting the system in canonical form, these are usually quite complicated and the solution might require some time;
- The computation of a grid with 3250 points required ∼3 weeks on 26 cores.

Mass evolution





- We can re-use the grid from the **Neutral-current Drell-Yan**;
- We have to solve a **36x36** system of differential equations w.r.t. to the Mandelstam variables s and t;
- Then, for every point, we have to solve a **56x56, but easier, system** w.r.t. one mass;
- We used this as a **cross-check**.